

SUSE Manager API

Version 27

May 29, 2024



Table of Contents

Frequently Asked Questions	1
1. What programming languages are supported by the SUSE Manager API?	2
2. When trying to call a specific function, the error "Fault returned from XML RPC Server, fault code -1: Could not find method METHOD in class..." is given. What is wrong?	3
3. Should I call an API method using the naming scheme "methodName" or "method_name"?	4
actionchain	5
4. Available methods	6
5. Description	7
6. Method: addConfigurationDeployment	8
7. Method: addErrataUpdate	9
8. Method: addErrataUpdate	10
9. Method: addErrataUpdate	11
10. Method: addPackageInstall	12
11. Method: addPackageRemoval	13
12. Method: addPackageUpgrade	14
13. Method: addPackageVerify	15
14. Method: addScriptRun	16
15. Method: addScriptRun	17
16. Method: addSystemReboot	18
17. Method: createChain	19
18. Method: deleteChain	20
19. Method: listChainActions	21
20. Method: listChains	22
21. Method: removeAction	23
22. Method: renameChain	24
23. Method: scheduleChain	25
activationkey	26
24. Available methods	27
25. Description	28
26. Method: addChildChannels	29
27. Method: addConfigChannels	30
28. Method: addEntitlements	31
29. Method: addPackages	32
30. Method: addServerGroups	33
31. Method: checkConfigDeployment	34
32. Method: clone	35
33. Method: create	36
34. Method: create	37
35. Method: delete	38
36. Method: disableConfigDeployment	39

37. Method: enableConfigDeployment	40
38. Method: getDetails	41
39. Method: listActivatedSystems	43
40. Method: listActivationKeys	44
41. Method: listChannels	46
42. Method: listConfigChannels	47
43. Method: removeChildChannels	48
44. Method: removeConfigChannels	49
45. Method: removeEntitlements	50
46. Method: removePackages	51
47. Method: removeServerGroups	52
48. Method: setConfigChannels	53
49. Method: setDetails	54
admin	55
50. Available methods	56
51. Description	57
52. Method: create	58
53. Method: create	59
54. Method: delete	60
55. Method: getDetails	61
56. Method: list	62
57. Method: setDetails	63
admin.configuration	64
58. Available methods	65
59. Description	66
60. Method: configure	67
admin.monitoring	68
61. Available methods	69
62. Description	70
63. Method: disable	71
64. Method: enable	72
65. Method: getStatus	73
ansible	74
66. Available methods	75
67. Description	76
68. Method: createAnsiblePath	77
69. Method: discoverPlaybooks	78
70. Method: fetchPlaybookContents	79
71. Method: introspectInventory	80
72. Method: listAnsiblePaths	81
73. Method: lookupAnsiblePathById	82
74. Method: removeAnsiblePath	83
75. Method: schedulePlaybook	84
76. Method: schedulePlaybook	85

77. Method: schedulePlaybook	86
78. Method: schedulePlaybook	87
79. Method: updateAnsiblePath	88
api	89
80. Available methods	90
81. Description	91
82. Method: getApiCallList	92
83. Method: getApiNamespaceCallList	93
84. Method: getApiNamespaces	94
85. Method: getVersion	95
86. Method: systemVersion	96
SUSE Manager API Documentation	97
87. Legal Notice	98
audit	101
88. Available methods	102
89. Description	103
90. Method: listImagesByPatchStatus	104
91. Method: listImagesByPatchStatus	105
92. Method: listSystemsByPatchStatus	106
93. Method: listSystemsByPatchStatus	107
auth	108
94. Available methods	109
95. Description	110
96. Method: login	111
97. Method: login	112
98. Method: logout	113
channel.access	114
99. Available methods	115
100. Description	116
101. Method: disableUserRestrictions	117
102. Method: enableUserRestrictions	118
103. Method: getOrgSharing	119
104. Method: setOrgSharing	120
channel	121
105. Available methods	122
106. Description	123
107. Method: listAllChannels	124
108. Method: listManageableChannels	125
109. Method: listMyChannels	126
110. Method: listPopularChannels	127
111. Method: listRetiredChannels	128
112. Method: listSharedChannels	129
113. Method: listSoftwareChannels	130
114. Method: listVendorChannels	131

channel.appstreams	132
115. Available methods	133
116. Description	134
117. Method: isModular	135
118. Method: listModular	136
119. Method: listModuleStreams	138
channel.org	139
120. Available methods	140
121. Description	141
122. Method: disableAccess	142
123. Method: enableAccess	143
124. Method: list	144
channel.software	145
125. Available methods	146
126. Description	149
127. Method: addPackages	150
128. Method: addRepoFilter	151
129. Method: alignMetadata	152
130. Method: applyChannelState	153
131. Method: associateRepo	154
132. Method: clearRepoFilters	156
133. Method: clone	157
134. Method: create	158
135. Method: create	159
136. Method: create	160
137. Method: create	161
138. Method: createRepo	162
139. Method: createRepo	163
140. Method: createRepo	164
141. Method: delete	165
142. Method: disassociateRepo	166
143. Method: getChannelLastBuildById	168
144. Method: getDetails	169
145. Method: getDetails	171
146. Method: getRepoDetails	173
147. Method: getRepoDetails	174
148. Method: getRepoSyncCronExpression	175
149. Method: isExisting	176
150. Method: isGloballySubscribable	177
151. Method: isUserManageable	178
152. Method: isUserSubscribable	179
153. Method: listAllPackages	180
154. Method: listAllPackages	181
155. Method: listAllPackages	182

156. Method: listArches	183
157. Method: listChannelRepos	184
158. Method: listChildren	185
159. Method: listErrata	187
160. Method: listErrata	188
161. Method: listErrata	189
162. Method: listErrata	190
163. Method: listErrataByType	191
164. Method: listErrataNeedingSync	192
165. Method: listLatestPackages	193
166. Method: listPackagesWithoutChannel	194
167. Method: listRepoFilters	195
168. Method: listSubscribedSystems	196
169. Method: listSystemChannels	197
170. Method: listUserRepos	198
171. Method: mergeErrata	199
172. Method: mergeErrata	200
173. Method: mergeErrata	201
174. Method: mergePackages	202
175. Method: mergePackages	203
176. Method: regenerateNeededCache	204
177. Method: regenerateNeededCache	205
178. Method: regenerateYumCache	206
179. Method: removeErrata	207
180. Method: removePackages	208
181. Method: removeRepo	209
182. Method: removeRepo	210
183. Method: removeRepoFilter	211
184. Method: setContactDetails	212
185. Method: setDetails	213
186. Method: setDetails	214
187. Method: setGloballySubscribable	215
188. Method: setRepoFilters	216
189. Method: setUserManageable	217
190. Method: setUserSubscribable	218
191. Method: syncErrata	219
192. Method: syncRepo	220
193. Method: syncRepo	221
194. Method: syncRepo	222
195. Method: syncRepo	223
196. Method: syncRepo	224
197. Method: updateRepo	225
198. Method: updateRepoLabel	226
199. Method: updateRepoLabel	227
200. Method: updateRepoSsl	228

201. Method: updateRepoSsl	229
202. Method: updateRepoUrl	230
203. Method: updateRepoUrl	231
configchannel	232
204. Available methods	233
205. Description	235
206. Method: channelExists	236
207. Method: create	237
208. Method: create	238
209. Method: create	239
210. Method: createOrUpdatePath	241
211. Method: createOrUpdateSymlink	243
212. Method: deleteChannels	245
213. Method: deleteFileRevisions	246
214. Method: deleteFiles	247
215. Method: deployAllSystems	248
216. Method: deployAllSystems	249
217. Method: deployAllSystems	250
218. Method: deployAllSystems	251
219. Method: getDetails	252
220. Method: getDetails	253
221. Method: getEncodedFileRevision	254
222. Method: getFileRevision	256
223. Method: getFileRevisions	258
224. Method: listAssignedSystemGroups	260
225. Method: listFiles	261
226. Method: listGlobals	262
227. Method: listSubscribedSystems	263
228. Method: lookupChannelInfo	264
229. Method: lookupFileInfo	265
230. Method: lookupFileInfo	267
231. Method: scheduleFileComparisons	269
232. Method: syncSaltFilesOnDisk	270
233. Method: update	271
234. Method: updateInitSls	272
contentmanagement	274
235. Available methods	275
236. Description	277
237. Method: attachFilter	278
238. Method: attachSource	279
239. Method: attachSource	280
240. Method: buildProject	281
241. Method: buildProject	282
242. Method: createAppStreamFilters	283

243. Method: createEnvironment	284
244. Method: createFilter	285
245. Method: createProject	287
246. Method: detachFilter	288
247. Method: detachSource	289
248. Method: listFilterCriteria	290
249. Method: listFilters	291
250. Method: listProjectEnvironments	292
251. Method: listProjectFilters	293
252. Method: listProjectSources	294
253. Method: listProjects	295
254. Method: lookupEnvironment	296
255. Method: lookupFilter	297
256. Method: lookupProject	298
257. Method: lookupSource	299
258. Method: promoteProject	300
259. Method: removeEnvironment	301
260. Method: removeFilter	302
261. Method: removeProject	303
262. Method: updateEnvironment	304
263. Method: updateFilter	305
264. Method: updateProject	306
distchannel	307
265. Available methods	308
266. Description	309
267. Method: listDefaultMaps	310
268. Method: listMapsForOrg	311
269. Method: listMapsForOrg	312
270. Method: setMapForOrg	313
errata	314
271. Available methods	315
272. Description	316
273. Method: addPackages	317
274. Method: applicableToChannels	318
275. Method: bugzillaFixes	319
276. Method: clone	320
277. Method: cloneAsOriginal	321
278. Method: cloneAsOriginalAsync	322
279. Method: cloneAsync	323
280. Method: create	324
281. Method: delete	326
282. Method: findByCve	327
283. Method: getDetails	328
284. Method: listAffectedSystems	330

285. Method: listCves	331
286. Method: listKeywords	332
287. Method: listPackages	333
288. Method: publish	335
289. Method: publishAsOriginal	336
290. Method: removePackages	337
291. Method: setDetails	338
formula	340
292. Available methods	341
293. Description	342
294. Method: getCombinedFormulaDataByServerIds	343
295. Method: getCombinedFormulasByServerId	344
296. Method: getFormulasByGroupId	345
297. Method: getFormulasByServerId	346
298. Method: getGroupFormulaData	347
299. Method: getSystemFormulaData	348
300. Method: listFormulas	349
301. Method: setFormulasOfGroup	350
302. Method: setFormulasOfServer	351
303. Method: setGroupFormulaData	352
304. Method: setSystemFormulaData	353
image	354
305. Available methods	355
306. Description	356
307. Method: addImageFile	357
308. Method: delete	358
309. Method: deleteImageFile	359
310. Method: getCustomValues	360
311. Method: getDetails	361
312. Method: getPillar	363
313. Method: getRelevantErrata	364
314. Method: importContainerImage	365
315. Method: importImage (Deprecated)	366
316. Method: importOSImage	367
317. Method: listImages	368
318. Method: listPackages	369
319. Method: scheduleImageBuild	370
320. Method: setPillar	371
image.delta	372
321. Available methods	373
322. Description	374
323. Method: createDeltaImage	375
324. Method: getDetails	376
325. Method: listDeltas	377

image.profile	378
326. Available methods	379
327. Description	380
328. Method: create	381
329. Method: create	382
330. Method: delete	383
331. Method: deleteCustomValues	384
332. Method: getCustomValues	385
333. Method: getDetails	386
334. Method: listImageProfileTypes	387
335. Method: listImageProfiles	388
336. Method: setCustomValues	389
337. Method: setDetails	390
image.store	391
338. Available methods	392
339. Description	393
340. Method: create	394
341. Method: delete	395
342. Method: getDetails	396
343. Method: listImageStoreTypes	397
344. Method: listImageStores	398
345. Method: setDetails	399
kickstart	400
346. Available methods	401
347. Description	402
348. Method: cloneProfile	403
349. Method: createProfile	404
350. Method: createProfile	405
351. Method: createProfileWithCustomUrl	406
352. Method: createProfileWithCustomUrl	407
353. Method: deleteProfile	408
354. Method: disableProfile	409
355. Method: findKickstartForIp	410
356. Method: importFile	411
357. Method: importFile	412
358. Method: importFile	413
359. Method: importRawFile	414
360. Method: importRawFile	415
361. Method: isProfileDisabled	416
362. Method: listAllIpRanges	417
363. Method: listAutoinstallableChannels	418
364. Method: listKickstartableChannels	420
365. Method: listKickstarts	422
366. Method: renameProfile	423

kickstart.filepreservation	424
367. Available methods	425
368. Description	426
369. Method: create	427
370. Method: delete	428
371. Method: getDetails	429
372. Method: listAllFilePreservations	430
kickstart.keys	431
373. Available methods	432
374. Description	433
375. Method: create	434
376. Method: delete	435
377. Method: getDetails	436
378. Method: listAllKeys	437
379. Method: update	438
kickstart.profile	439
380. Available methods	440
381. Description	442
382. Method: addIpRange	443
383. Method: addScript	444
384. Method: addScript	445
385. Method: addScript	446
386. Method: compareActivationKeys	447
387. Method: compareAdvancedOptions	449
388. Method: comparePackages	450
389. Method: downloadKickstart	451
390. Method: downloadRenderedKickstart	452
391. Method: getAdvancedOptions	453
392. Method: getAvailableRepositories	454
393. Method: getCfgPreservation	455
394. Method: getChildChannels	456
395. Method: getCustomOptions	457
396. Method: getKickstartTree	458
397. Method: getRepositories	459
398. Method: getUpdateType	460
399. Method: getVariables	461
400. Method: getVirtualizationType	462
401. Method: listIpRanges	463
402. Method: listScripts	464
403. Method: orderScripts	465
404. Method: removeIpRange	466
405. Method: removeScript	467
406. Method: setAdvancedOptions	468
407. Method: setCfgPreservation	469

408. Method: setChildChannels	470
409. Method: setCustomOptions	471
410. Method: setKickstartTree	472
411. Method: setLogging	473
412. Method: setRepositories	474
413. Method: setUpdateType	475
414. Method: setVariables	476
415. Method: setVirtualizationType	477
kickstart.profile.keys	478
416. Available methods	479
417. Description	480
418. Method: addActivationKey	481
419. Method: getActivationKeys	482
420. Method: removeActivationKey	484
kickstart.profile.software	485
421. Available methods	486
422. Description	487
423. Method: appendToSoftwareList	488
424. Method: getSoftwareDetails	489
425. Method: getSoftwareList	490
426. Method: setSoftwareDetails	491
427. Method: setSoftwareList	492
428. Method: setSoftwareList	493
kickstart.profile.system	494
429. Available methods	495
430. Description	496
431. Method: addFilePreservations	497
432. Method: addKeys	498
433. Method: checkConfigManagement	499
434. Method: checkRemoteCommands	500
435. Method: disableConfigManagement	501
436. Method: disableRemoteCommands	502
437. Method: enableConfigManagement	503
438. Method: enableRemoteCommands	504
439. Method: getLocale	505
440. Method: getPartitioningScheme	506
441. Method: getRegistrationType	507
442. Method: getSELinux	508
443. Method: listFilePreservations	509
444. Method: listKeys	510
445. Method: removeFilePreservations	511
446. Method: removeKeys	512
447. Method: setLocale	513
448. Method: setPartitioningScheme	514

449. Method: setRegistrationType	515
450. Method: setSELinux	516
kickstart.snippet	517
451. Available methods	518
452. Description	519
453. Method: createOrUpdate	520
454. Method: delete	521
455. Method: listAll	522
456. Method: listCustom	523
457. Method: listDefault	524
kickstart.tree	525
458. Available methods	526
459. Description	527
460. Method: create	528
461. Method: create	529
462. Method: delete	530
463. Method: deleteTreeAndProfiles	531
464. Method: getDetails	532
465. Method: list	533
466. Method: listInstallTypes	534
467. Method: rename	535
468. Method: update	536
469. Method: update	537
maintenance	538
470. Available methods	539
471. Description	540
472. Method: assignScheduleToSystems	541
473. Method: createCalendar	542
474. Method: createCalendarWithUrl	543
475. Method: createSchedule	544
476. Method: createSchedule	545
477. Method: deleteCalendar	546
478. Method: deleteSchedule	547
479. Method: getCalendarDetails	548
480. Method: getScheduleDetails	549
481. Method: listCalendarLabels	550
482. Method: listScheduleNames	551
483. Method: listSystemsWithSchedule	552
484. Method: refreshCalendar	553
485. Method: retractScheduleFromSystems	555
486. Method: updateCalendar	556
487. Method: updateSchedule	558
org	560
488. Available methods	561

489. Description	562
490. Method: create	563
491. Method: createFirst	565
492. Method: delete	566
493. Method: getCImSyncPatchesConfig	567
494. Method: getDetails	568
495. Method: getDetails	569
496. Method: getPolicyForScapFileUpload	570
497. Method: getPolicyForScapResultDeletion	571
498. Method: isContentStagingEnabled	572
499. Method: isErrataEmailNotifsForOrg	573
500. Method: isOrgConfigManagedByOrgAdmin	574
501. Method: listOrgs	575
502. Method: listUsers	576
503. Method: migrateSystems (Deprecated)	577
504. Method: setCImSyncPatchesConfig	578
505. Method: setContentStaging	579
506. Method: setErrataEmailNotifsForOrg	580
507. Method: setOrgConfigManagedByOrgAdmin	581
508. Method: setPolicyForScapFileUpload	582
509. Method: setPolicyForScapResultDeletion	583
510. Method: transferSystems	584
511. Method: updateName	585
org.trusts	586
512. Available methods	587
513. Description	588
514. Method: addTrust	589
515. Method: getDetails	590
516. Method: listChannelsConsumed	591
517. Method: listChannelsProvided	592
518. Method: listOrgs	593
519. Method: listSystemsAffected	594
520. Method: listTrusts	595
521. Method: removeTrust	596
packages	597
522. Available methods	598
523. Description	599
524. Method: findByNvrea	600
525. Method: getDetails	602
526. Method: getPackage	604
527. Method: getPackageUrl	605
528. Method: listChangelog	606
529. Method: listDependencies	607
530. Method: listFiles	608

531. Method: listProvidingChannels	609
532. Method: listProvidingErrata	610
533. Method: listSourcePackages	611
534. Method: removePackage	612
535. Method: removeSourcePackage	613
packages.provider	614
536. Available methods	615
537. Description	616
538. Method: associateKey	617
539. Method: list	618
540. Method: listKeys	619
packages.search	620
541. Available methods	621
542. Description	622
543. Method: advanced	623
544. Method: advancedWithActKey	624
545. Method: advancedWithChannel	625
546. Method: name	626
547. Method: nameAndDescription	627
548. Method: nameAndSummary	628
preferences.locale	629
549. Available methods	630
550. Description	631
551. Method: listLocales	632
552. Method: listTimeZones	633
553. Method: setLocale	634
554. Method: setTimeZone	635
proxy	636
555. Available methods	637
556. Description	638
557. Method: activateProxy	639
558. Method: containerConfig	640
559. Method: containerConfig	641
560. Method: createMonitoringScout	642
561. Method: deactivateProxy	643
562. Method: isProxy	644
563. Method: listAvailableProxyChannels	645
564. Method: listProxies	646
565. Method: listProxyClients	647
recurring	648
566. Available methods	649
567. Description	650
568. Method: delete	651
569. Method: listByEntity	652

570. Method: lookupById	653
recurring.custom	654
571. Available methods	655
572. Description	656
573. Method: create	657
574. Method: listAvailable	658
575. Method: update	659
recurring.highstate	660
576. Available methods	661
577. Description	662
578. Method: create	663
579. Method: update	664
recurringaction	665
580. Available methods	666
581. Description	667
582. Method: create (Deprecated)	668
583. Method: delete (Deprecated)	669
584. Method: listByEntity (Deprecated)	670
585. Method: lookupById (Deprecated)	672
586. Method: update (Deprecated)	673
saltkey	674
587. Available methods	675
588. Description	676
589. Method: accept	677
590. Method: acceptedList	678
591. Method: delete	679
592. Method: deniedList	680
593. Method: pendingList	681
594. Method: reject	682
595. Method: rejectedList	683
schedule	684
596. Available methods	685
597. Description	686
598. Method: archiveActions	687
599. Method: cancelActions	688
600. Method: deleteActions	689
601. Method: failSystemAction	690
602. Method: failSystemAction	691
603. Method: listAllActions	692
604. Method: listAllArchivedActions	693
605. Method: listAllCompletedActions	694
606. Method: listArchivedActions	695
607. Method: listCompletedActions	696
608. Method: listCompletedSystems	697

609. Method: listFailedActions	698
610. Method: listFailedSystems	699
611. Method: listInProgressActions	700
612. Method: listInProgressSystems	701
613. Method: rescheduleActions	702
Sample Scripts	703
614. JSON over HTTP API scripts	704
614.1. HTTP login with Curl.....	704
614.2. HTTP GET example	704
614.3. HTTP POST example	705
614.3.1. Using the query string.....	705
614.3.2. Using the request body.....	706
615. XMLRPC Scripts	708
615.1. Perl Example.....	708
615.2. Python 2 Example	708
615.3. Python 3 with SSL Example	709
615.4. Python 3 Example.....	710
615.5. Ruby Example	710
subscriptionmatching.pinnedsubscription	712
616. Available methods	713
617. Description	714
618. Method: create	715
619. Method: delete	716
620. Method: list	717
sync.content	718
621. Available methods	719
622. Description	720
623. Method: addChannel	721
624. Method: addChannels	722
625. Method: addCredentials	723
626. Method: deleteCredentials	724
627. Method: listChannels	725
628. Method: listCredentials	726
629. Method: listProducts	727
630. Method: synchronizeChannelFamilies	729
631. Method: synchronizeChannels	730
632. Method: synchronizeProducts	731
633. Method: synchronizeRepositories	732
634. Method: synchronizeSubscriptions	733
sync.master	734
635. Available methods	735
636. Description	736
637. Method: addToMaster	737
638. Method: create	738
639. Method: delete	739

640. Method: getDefaultMaster	740
641. Method: getMaster	741
642. Method: getMasterByLabel	742
643. Method: getMasterOrgs	743
644. Method: getMasters	744
645. Method: hasMaster	745
646. Method: makeDefault	746
647. Method: mapToLocal	747
648. Method: setCaCert	748
649. Method: setMasterOrgs	749
650. Method: unsetDefaultMaster	750
651. Method: update	751
sync.slave	752
652. Available methods	753
653. Description	754
654. Method: create	755
655. Method: delete	756
656. Method: getAllowedOrgs	757
657. Method: getSlave	758
658. Method: getSlaveByName	759
659. Method: getSlaves	760
660. Method: setAllowedOrgs	761
661. Method: update	762
system	763
662. Available methods	764
663. Description	773
664. Method: addEntitlements	774
665. Method: addNote	775
666. Method: bootstrap	776
667. Method: bootstrap	777
668. Method: bootstrap	778
669. Method: bootstrap	779
670. Method: bootstrapWithPrivateSshKey	780
671. Method: bootstrapWithPrivateSshKey	781
672. Method: bootstrapWithPrivateSshKey	782
673. Method: bootstrapWithPrivateSshKey	783
674. Method: changeProxy	784
675. Method: comparePackageProfile	785
676. Method: comparePackages	787
677. Method: createPackageProfile	788
678. Method: createSystemProfile	789
679. Method: createSystemRecord	790
680. Method: createSystemRecord	791
681. Method: deleteCustomValues	792

682. Method: deleteGuestProfiles	793
683. Method: deleteNote	794
684. Method: deleteNotes	795
685. Method: deletePackageProfile	796
686. Method: deleteSystem	797
687. Method: deleteSystem	798
688. Method: deleteSystem	799
689. Method: deleteSystems	800
690. Method: deleteSystems	801
691. Method: deleteTagFromSnapshot	802
692. Method: downloadSystemId	803
693. Method: getCoCoAttestationConfig	804
694. Method: getCoCoAttestationResultDetails	805
695. Method: getConnectionPath	806
696. Method: getCpu	807
697. Method: getCustomValues	808
698. Method: getDetails	809
699. Method: getDevices	811
700. Method: getDmi	812
701. Method: getEntitlements	813
702. Method: getEventDetails	814
703. Method: getEventHistory (Deprecated)	816
704. Method: getEventHistory	817
705. Method: getEventHistory	818
706. Method: getEventHistory	819
707. Method: getId	820
708. Method: getInstalledProducts	821
709. Method: getKernelLivePatch	822
710. Method: getLatestCoCoAttestationReport	823
711. Method: getMemory	824
712. Method: getMinionIdMap	825
713. Method: getName	826
714. Method: getNetwork	827
715. Method: getNetworkDevices	828
716. Method: getNetworkForSystems	829
717. Method: getOsaPing	830
718. Method: getPillar	831
719. Method: getPillar	832
720. Method: getRegistrationDate	833
721. Method: getRelevantErrata	834
722. Method: getRelevantErrata	835
723. Method: getRelevantErrataByType	836
724. Method: getRunningKernel	837
725. Method: getScriptActionDetails	838
726. Method: getScriptResults	839

727. Method: getSubscribedBaseChannel	840
728. Method: getSystemCurrencyMultipliers	842
729. Method: getSystemCurrencyScores	843
730. Method: getUnscheduledErrata	844
731. Method: getUuid	845
732. Method: getVariables	846
733. Method: isNvreInstalled	847
734. Method: isNvreInstalled	848
735. Method: listActivationKeys	849
736. Method: listActiveSystems	850
737. Method: listActiveSystemsDetails	851
738. Method: listAdministrators	854
739. Method: listAllInstallablePackages	855
740. Method: listCoCoAttestationReports	856
741. Method: listCoCoAttestationReports	857
742. Method: listCoCoAttestationReports	858
743. Method: listDuplicatesByHostname	859
744. Method: listDuplicatesByIp	860
745. Method: listDuplicatesByMac	861
746. Method: listEmptySystemProfiles	862
747. Method: listExtraPackages	863
748. Method: listFqdns	864
749. Method: listGroups	865
750. Method: listInactiveSystems	866
751. Method: listInactiveSystems	867
752. Method: listInstalledPackages	868
753. Method: listLatestAvailablePackage	869
754. Method: listLatestInstallablePackages	870
755. Method: listLatestUpgradablePackages	871
756. Method: listMigrationTargets	872
757. Method: listMigrationTargets	873
758. Method: listNewerInstalledPackages	874
759. Method: listNotes	875
760. Method: listOlderInstalledPackages	876
761. Method: listOutOfDateSystems	877
762. Method: listPackageProfiles	878
763. Method: listPackageState	879
764. Method: listPackages (Deprecated)	880
765. Method: listPackagesFromChannel	881
766. Method: listPackagesLockStatus	882
767. Method: listPhysicalSystems	883
768. Method: listSubscribableBaseChannels	884
769. Method: listSubscribableChildChannels	885
770. Method: listSubscribedChildChannels	886
771. Method: listSuggestedReboot	888

772. Method: listSystemEvents	889
773. Method: listSystemEvents	891
774. Method: listSystemEvents	893
775. Method: listSystemEvents	895
776. Method: listSystemGroupsForSystemsWithEntitlement	897
777. Method: listSystems	898
778. Method: listSystemsWithEntitlement	899
779. Method: listSystemsWithExtraPackages	900
780. Method: listSystemsWithPackage	901
781. Method: listSystemsWithPackage	902
782. Method: listUngroupedSystems	903
783. Method: listUserSystems	904
784. Method: listUserSystems	905
785. Method: listVirtualGuests	906
786. Method: listVirtualHosts	907
787. Method: obtainReactivationKey	908
788. Method: obtainReactivationKey	909
789. Method: provisionSystem	910
790. Method: provisionSystem	911
791. Method: provisionSystem	912
792. Method: provisionSystem	913
793. Method: provisionVirtualGuest	914
794. Method: provisionVirtualGuest	915
795. Method: provisionVirtualGuest	916
796. Method: refreshPillar	917
797. Method: refreshPillar	918
798. Method: registerPeripheralServer	919
799. Method: removeEntitlements	920
800. Method: scheduleApplyErrata	921
801. Method: scheduleApplyErrata	922
802. Method: scheduleApplyErrata	923
803. Method: scheduleApplyErrata	924
804. Method: scheduleApplyErrata	925
805. Method: scheduleApplyErrata	926
806. Method: scheduleApplyErrata	927
807. Method: scheduleApplyErrata	928
808. Method: scheduleApplyErrata	929
809. Method: scheduleApplyErrata	930
810. Method: scheduleApplyHighstate	931
811. Method: scheduleApplyHighstate	932
812. Method: scheduleApplyStates	933
813. Method: scheduleApplyStates	934
814. Method: scheduleCertificateUpdate	935
815. Method: scheduleCertificateUpdate	936
816. Method: scheduleChangeChannels	937

817. Method: scheduleChangeChannels	938
818. Method: scheduleCoCoAttestation	939
819. Method: scheduleDistUpgrade	940
820. Method: scheduleDistUpgrade	941
821. Method: scheduleGuestAction	942
822. Method: scheduleGuestAction	943
823. Method: scheduleHardwareRefresh	944
824. Method: schedulePackageInstall	945
825. Method: schedulePackageInstall	946
826. Method: schedulePackageInstall	947
827. Method: schedulePackageInstall	948
828. Method: schedulePackageInstallByNevra	949
829. Method: schedulePackageInstallByNevra	950
830. Method: schedulePackageInstallByNevra	951
831. Method: schedulePackageInstallByNevra	952
832. Method: schedulePackageLockChange	953
833. Method: schedulePackageRefresh	954
834. Method: schedulePackageRemove	955
835. Method: schedulePackageRemove	956
836. Method: schedulePackageRemove	957
837. Method: schedulePackageRemove	958
838. Method: schedulePackageRemoveByNevra	959
839. Method: schedulePackageRemoveByNevra	960
840. Method: schedulePackageRemoveByNevra	961
841. Method: schedulePackageRemoveByNevra	962
842. Method: schedulePackageUpdate	963
843. Method: scheduleProductMigration	964
844. Method: scheduleProductMigration	965
845. Method: scheduleProductMigration	966
846. Method: scheduleProductMigration	967
847. Method: scheduleProductMigration	968
848. Method: scheduleReboot	969
849. Method: scheduleSPMigration (Deprecated)	970
850. Method: scheduleSPMigration (Deprecated)	971
851. Method: scheduleSPMigration (Deprecated)	972
852. Method: scheduleSPMigration (Deprecated)	973
853. Method: scheduleScriptRun	974
854. Method: scheduleScriptRun	975
855. Method: scheduleScriptRun	976
856. Method: scheduleScriptRun	977
857. Method: scheduleSyncPackagesWithSystem	978
858. Method: searchByName	979
859. Method: sendOsaPing	980
860. Method: setBaseChannel (Deprecated)	981
861. Method: setBaseChannel (Deprecated)	982

862. Method: setChildChannels (Deprecated)	983
863. Method: setCoCoAttestationConfig	984
864. Method: setCustomValues	985
865. Method: setDetails	986
866. Method: setGroupMembership	988
867. Method: setGuestCpus	989
868. Method: setGuestMemory	990
869. Method: setLockStatus	991
870. Method: setPillar	992
871. Method: setPillar	993
872. Method: setPrimaryFqdn	994
873. Method: setPrimaryInterface	995
874. Method: setProfileName	996
875. Method: setVariables	997
876. Method: tagLatestSnapshot	998
877. Method: unentitle	999
878. Method: updatePackageState	1000
879. Method: updatePeripheralServerInfo	1001
880. Method: upgradeEntitlement	1002
881. Method: whoRegistered	1003
system.appstreams	1004
882. Available methods	1005
883. Description	1006
884. Method: disable	1007
885. Method: enable	1008
886. Method: listModuleStreams	1009
system.config	1010
887. Available methods	1011
888. Description	1012
889. Method: addChannels	1013
890. Method: createOrUpdatePath	1014
891. Method: createOrUpdateSymlink	1017
892. Method: deleteFiles	1019
893. Method: deployAll	1020
894. Method: listChannels	1021
895. Method: listFiles	1022
896. Method: lookupFileInfo	1024
897. Method: removeChannels	1026
898. Method: scheduleApplyConfigChannel	1027
899. Method: setChannels	1028
system.custominfo	1029
900. Available methods	1030
901. Description	1031
902. Method: createKey	1032

903. Method: deleteKey	1033
904. Method: listAllKeys	1034
905. Method: updateKey	1035
system.monitoring	1036
906. Available methods	1037
907. Description	1038
908. Method: listEndpoints	1039
system.provisioning.powermanagement	1040
909. Available methods	1041
910. Description	1042
911. Method: getDetails	1043
912. Method: getDetails	1044
913. Method: getStatus	1045
914. Method: getStatus	1046
915. Method: listTypes	1047
916. Method: powerOff	1048
917. Method: powerOff	1049
918. Method: powerOn	1050
919. Method: powerOn	1051
920. Method: reboot	1052
921. Method: reboot	1053
922. Method: setDetails	1054
923. Method: setDetails	1055
system.provisioning.snapshot	1056
924. Available methods	1057
925. Description	1058
926. Method: addTagToSnapshot	1059
927. Method: deleteSnapshot	1060
928. Method: deleteSnapshots	1061
929. Method: deleteSnapshots	1062
930. Method: deleteSnapshots	1063
931. Method: deleteSnapshots	1064
932. Method: listSnapshotConfigFiles	1065
933. Method: listSnapshotPackages	1067
934. Method: listSnapshots	1068
935. Method: listSnapshots	1070
936. Method: rollbackToSnapshot	1072
937. Method: rollbackToTag	1073
938. Method: rollbackToTag	1074
system.scap	1075
939. Available methods	1076
940. Description	1077
941. Method: deleteXccdfScan	1078
942. Method: getXccdfScanDetails	1079

943. Method: getXccdfScanRuleResults	1080
944. Method: listXccdfScans	1081
945. Method: scheduleXccdfScan	1082
946. Method: scheduleXccdfScan	1083
947. Method: scheduleXccdfScan	1084
948. Method: scheduleXccdfScan	1085
949. Method: scheduleXccdfScan	1086
system.search	1087
950. Available methods	1088
951. Description	1089
952. Method: deviceDescription	1090
953. Method: deviceDriver	1091
954. Method: deviceId	1092
955. Method: deviceVendorId	1093
956. Method: hostname	1094
957. Method: ip	1095
958. Method: nameAndDescription	1096
959. Method: uuid	1097
systemgroup	1098
960. Available methods	1099
961. Description	1100
962. Method: addOrRemoveAdmins	1101
963. Method: addOrRemoveSystems	1102
964. Method: create	1103
965. Method: delete	1104
966. Method: getDetails	1105
967. Method: getDetails	1106
968. Method: listActiveSystemsInGroup	1107
969. Method: listAdministrators	1108
970. Method: listAllGroups	1109
971. Method: listAssignedConfigChannels	1110
972. Method: listAssignedFormuals	1111
973. Method: listGroupsWithNoAssociatedAdmins	1112
974. Method: listInactiveSystemsInGroup	1113
975. Method: listInactiveSystemsInGroup	1114
976. Method: listSystems	1115
977. Method: listSystemsMinimal	1117
978. Method: scheduleApplyErrataToActive	1118
979. Method: scheduleApplyErrataToActive	1119
980. Method: scheduleApplyErrataToActive	1120
981. Method: subscribeConfigChannel	1121
982. Method: unsubscribeConfigChannel	1122
983. Method: update	1123
user	1124

984. Available methods	1125
985. Description	1127
986. Method: addAssignedSystemGroup	1128
987. Method: addAssignedSystemGroups	1129
988. Method: addDefaultSystemGroup	1130
989. Method: addDefaultSystemGroups	1131
990. Method: addRole	1132
991. Method: create	1133
992. Method: create	1134
993. Method: delete	1135
994. Method: disable	1136
995. Method: enable	1137
996. Method: getCreateDefaultSystemGroup	1138
997. Method: getDetails	1139
998. Method: listAssignableRoles	1140
999. Method: listAssignedSystemGroups	1141
1000. Method: listDefaultSystemGroups	1142
1001. Method: listRoles	1143
1002. Method: listUsers	1144
1003. Method: removeAssignedSystemGroup	1145
1004. Method: removeAssignedSystemGroups	1146
1005. Method: removeDefaultSystemGroup	1147
1006. Method: removeDefaultSystemGroups	1148
1007. Method: removeRole	1149
1008. Method: setCreateDefaultSystemGroup	1150
1009. Method: setDetails	1151
1010. Method: setErrataNotifications	1152
1011. Method: setReadOnly	1153
1012. Method: usePamAuthentication	1154
user.external	1155
1013. Available methods	1156
1014. Description	1157
1015. Method: createExternalGroupToRoleMap	1158
1016. Method: createExternalGroupToSystemGroupMap	1159
1017. Method: deleteExternalGroupToRoleMap	1160
1018. Method: deleteExternalGroupToSystemGroupMap	1161
1019. Method: getDefaultOrg	1162
1020. Method: getExternalGroupToRoleMap	1163
1021. Method: getExternalGroupToSystemGroupMap	1164
1022. Method: getKeepTemporaryRoles	1165
1023. Method: getUseOrgUnit	1166
1024. Method: listExternalGroupToRoleMaps	1167
1025. Method: listExternalGroupToSystemGroupMaps	1168
1026. Method: setDefaultOrg	1169
1027. Method: setExternalGroupRoles	1170

1028. Method: setExternalGroupSystemGroups	1171
1029. Method: setKeepTemporaryRoles	1172
1030. Method: setUseOrgUnit	1173
virtualhostmanager	1174
1031. Available methods	1175
1032. Description	1176
1033. Method: create	1177
1034. Method: delete	1178
1035. Method: getDetail	1179
1036. Method: getModuleParameters	1180
1037. Method: listAvailableVirtualHostGathererModules	1181
1038. Method: listVirtualHostManagers	1182

Chapter 1. What programming languages are supported by the SUSE Manager API?

Any language that provides an XMLRPC client interface will work with the SUSE Manager API. While Perl and Python are two of the most commonly used, an XMLRPC client implementation is available for every common language.

Chapter 2. When trying to call a specific function, the error "Fault returned from XML RPC Server, fault code -1: Could not find method METHOD in class..." is given. What is wrong?

Typically this is seen when either a function name is being called that doesn't exist, the number of parameters for a particular function is incorrect, or the type of a passed parameter is incorrect (Such as an array is expected, but a String is passed). Check all of these things.

Chapter 3. Should I call an API method using the naming scheme "methodName" or "method_name"?

Both of these are valid names for the same method, so use whichever you prefer.

actionchain

Chapter 4. Available methods

- `addConfigurationDeployment`
- `addErrorataUpdate`
- `addErrorataUpdate`
- `addErrorataUpdate`
- `addErrorataUpdate`
- `addPackageInstall`
- `addPackageRemoval`
- `addPackageUpgrade`
- `addPackageVerify`
- `addScriptRun`
- `addScriptRun`
- `addSystemReboot`
- `createChain`
- `deleteChain`
- `listChainActions`
- `listChains`
- `removeAction`
- `renameChain`
- `scheduleChain`

Chapter 5. Description

Provides the namespace for the Action Chain methods.

Namespace:

actionchain

Chapter 6. Method: addConfigurationDeployment

HTTP **POST**

Description:

Adds an action to deploy a configuration file to an Action Chain.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain
- int sid - System ID
- array revisionSpecifiers
 - struct config revision specifier
 - string "channelLabel" - Channel label
 - string "filePath" - Path of the configuration file
 - int "revision" - Revision number

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 7. Method: addErrataUpdate

HTTP **POST**

Description:

Adds Errata update to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- int array errataIds - Errata ID
- string chainLabel - Label of the chain

Returns:

- int actionId - The action id of the scheduled action

Chapter 8. Method: addErrataUpdate

HTTP **POST**

Description:

Adds Errata update to an Action Chain.

Parameters:

- string sessionKey
- int array sids - System IDs
- int array errataIds - Errata ID
- string chainLabel - Label of the chain

Returns:

- int actionId - The action id of the scheduled action

Chapter 9. Method: addErrataUpdate

HTTP **POST**

Description:

Adds Errata update to an Action Chain.

Parameters:

- string sessionKey
- int array sids - System IDs
- int array errataIds - Errata ID
- string chainLabel - Label of the chain
- boolean onlyRelevant - If true, InvalidErrataException is thrown if an errata does not apply to a system.

Returns:

- int actionId - The action ID of the scheduled action

Chapter 10. Method: addPackageInstall

HTTP **POST**

Description:

Adds package installation action to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- int array packagelds
- string chainLabel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 11. Method: addPackageRemoval

HTTP **POST**

Description:

Adds an action to remove installed packages on the system to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- int array packagelds
- string chainLabel - Label of the chain

Returns:

- int actionId - The action id of the scheduled action or exception

Chapter 12. Method: addPackageUpgrade

HTTP **POST**

Description:

Adds an action to upgrade installed packages on the system to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- int array packagelds
- string chainLabel - Label of the chain

Returns:

- int actionId - The id of the action or throw an exception

Chapter 13. Method: addPackageVerify

HTTP **POST**

Description:

Adds an action to verify installed packages on the system to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- int array packageIds
- string chainLabel - Label of the chain

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 14. Method: addScriptRun

HTTP **POST**

Description:

Add an action with label to run a script to an Action Chain. NOTE: The script body must be Base64 encoded!

Parameters:

- string sessionKey
- int sid - System ID
- string chainLabel - Label of the chain
- string scriptLabel - Label of the script
- string uid - User ID on the particular system
- string gid - Group ID on the particular system
- int timeout - Timeout
- string scriptBody - Base64 encoded script body

Returns:

- int actionId - The id of the action or throw an exception

Chapter 15. Method: addScriptRun

HTTP **POST**

Description:

Add an action to run a script to an Action Chain. NOTE: The script body must be Base64 encoded!

Parameters:

- string sessionKey
- int sid - System ID
- string chainLabel - Label of the chain
- string uid - User ID on the particular system
- string gid - Group ID on the particular system
- int timeout - Timeout
- string scriptBody - Base64 encoded script body

Returns:

- int actionId - The id of the action or throw an exception

Chapter 16. Method: addSystemReboot

HTTP **POST**

Description:

Add system reboot to an Action Chain.

Parameters:

- string sessionKey
- int sid - System ID
- string chainLabel - Label of the chain

Returns:

- int actionId - The action id of the scheduled action

Chapter 17. Method: createChain

HTTP **POST**

Description:

Create an Action Chain.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain

Returns:

- int actionId - The ID of the created action chain

Chapter 18. Method: deleteChain

HTTP **POST**

Description:

Delete action chain by label.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 19. Method: listChainActions

HTTP **GET**

Description:

List all actions in the particular Action Chain.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain

Returns:

- array :
 - struct entry
 - int "id" - Action ID
 - string "label" - Label of an Action
 - string "created" - Created date/time
 - string "earliest" - Earliest scheduled date/time
 - string "type" - Type of the action
 - string "modified" - Modified date/time
 - string "cuid" - Creator UID

Chapter 20. Method: listChains

HTTP **GET**

Description:

List currently available action chains.

Parameters:

- string sessionKey

Returns:

- array :
 - struct chain
 - string "label" - Label of an Action Chain
 - string "entrycount" - Number of entries in the Action Chain

Chapter 21. Method: removeAction

HTTP **POST**

Description:

Remove an action from an Action Chain.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain
- int actionId - Action ID

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 22. Method: renameChain

HTTP **POST**

Description:

Rename an Action Chain.

Parameters:

- string sessionKey
- string previousLabel - Previous chain label
- string newLabel - New chain label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 23. Method: scheduleChain

HTTP **POST**

Description:

Schedule the Action Chain so that its actions will actually occur.

Parameters:

- string sessionKey
- string chainLabel - Label of the chain
- dateTime.iso8601 date - Earliest date

Returns:

- int - 1 on success, exception thrown otherwise.

activationkey

Chapter 24. Available methods

- `addChildChannels`
- `addConfigChannels`
- `addEntitlements`
- `addPackages`
- `addServerGroups`
- `checkConfigDeployment`
- `clone`
- `create`
- `create`
- `delete`
- `disableConfigDeployment`
- `enableConfigDeployment`
- `getDetails`
- `listActivatedSystems`
- `listActivationKeys`
- `listChannels`
- `listConfigChannels`
- `removeChildChannels`
- `removeConfigChannels`
- `removeEntitlements`
- `removePackages`
- `removeServerGroups`
- `setConfigChannels`
- `setDetails`

Chapter 25. Description

Contains methods to access common activation key functions available from the web interface.

Namespace:

activationkey

Chapter 26. Method: addChildChannels

HTTP **POST**

Description:

Add child channels to an activation key.

Parameters:

- string sessionKey
- string key
- string array childChannellLabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 27. Method: addConfigChannels

HTTP **POST**

Description:

Given a list of activation keys and configuration channels, this method adds given configuration channels to either the top or the bottom (whichever you specify) of an activation key's configuration channels list. The ordering of the configuration channels provided in the add list is maintained while adding. If one of the configuration channels in the 'add' list already exists in an activation key, the configuration channel will be re-ranked to the appropriate place.

Parameters:

- string sessionKey
- string array keys
- string array configChannelLabels - List of configuration channel labels in the ranked order.
- boolean addToTop
 - true - To prepend the given channels to the beginning of the activation key's config channel list
 - false - To append the given channels to the end of the activation key's config channel list

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 28. Method: addEntitlements

HTTP **POST**

Description:

Add add-on System Types to an activation key.

Parameters:

- string sessionKey
- string key
- string array entitlements - Add-on system type labels to associate with the key.
 - container_build_host
 - monitoring_entitled
 - osimage_build_host
 - virtualization_host
 - ansible_control_node

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 29. Method: addPackages

HTTP **POST**

Description:

Add packages to an activation key.

Parameters:

- string sessionKey
- string key
- array packages
 - struct package
 - string "name" - Package name
 - string "arch" - Arch label - Optional

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 30. Method: addServerGroups

HTTP **POST**

Description:

Add server groups to an activation key.

Parameters:

- string sessionKey
- string key
- int array serverGroupIds

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 31. Method: checkConfigDeployment

HTTP **POST**

Description:

Check configuration file deployment status for the activation key specified.

Parameters:

- string sessionKey
- string key

Returns:

- int status - 1 if enabled, 0 if disabled, exception thrown otherwise

Chapter 32. Method: clone

HTTP **POST**

Description:

Clone an existing activation key.

Parameters:

- string sessionKey
- string key - Key to be cloned.
- string cloneDescription - Description of the cloned key.

Returns:

- string The new activation key

Chapter 33. Method: create

HTTP **POST**

Description:

Create a new activation key. The activation key parameter passed in will be prefixed with the organization ID, and this value will be returned from the create call.

Eg. If the caller passes in the key "foo" and belong to an organization with the ID 100, the actual activation key will be "100-foo".

This call allows for the setting of a usage limit on this activation key. If unlimited usage is desired see the similarly named API method with no usage limit argument.

Parameters:

- string `sessionKey`
- string `key` - Leave empty to have new key autogenerated.
- string `description`
- string `baseChannelLabel` - Leave empty to accept default.
- int `usageLimit` - If unlimited usage is desired, use the create API that does not include the parameter.
- string array `entitlements` - Add-on system type labels to associate with the key.
 - `container_build_host`
 - `monitoring_entitled`
 - `osimage_build_host`
 - `virtualization_host`
 - `ansible_control_node`
- boolean `universalDefault`

Returns:

- string The new activation key

Chapter 34. Method: create

HTTP **POST**

Description:

Create a new activation key with unlimited usage. The activation key parameter passed in will be prefixed with the organization ID, and this value will be returned from the create call.

Eg. If the caller passes in the key "foo" and belong to an organization with the ID 100, the actual activation key will be "100-foo".

Parameters:

- string `sessionKey`
- string `key` - Leave empty to have new key autogenerated.
- string `description`
- string `baseChannelLabel` - Leave empty to accept default.
- string array `entitlements` - Add-on entitlement label to associate with the key.
 - `virtualization_host`
- boolean `universalDefault`

Returns:

- string The new activation key

Chapter 35. Method: delete

HTTP **POST**

Description:

Delete an activation key.

Parameters:

- string sessionKey
- string key

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 36. Method: disableConfigDeployment

HTTP **POST**

Description:

Disable configuration file deployment for the specified activation key.

Parameters:

- string sessionKey
- string key

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 37. Method: enableConfigDeployment

HTTP **POST**

Description:

Enable configuration file deployment for the specified activation key.

Parameters:

- string sessionKey
- string key

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 38. Method: getDetails

HTTP **GET**

Description:

Lookup an activation key's details.

Parameters:

- string sessionKey
- string key

Returns:

- * struct activation key
 - string "key"
 - string "description"
 - int "usage_limit"
 - string "base_channel_label"
 - string array "child_channel_labels" - childChannelLabel
 - string array "entitlements" - entitlementLabel
 - string array "server_group_ids" - serverGroupId
 - string array "package_names" - packageName - (deprecated by packages)
 - array "packages"
 - struct package
 - string "name" - packageName
 - string "arch" - archLabel - optional
 - boolean "universal_default"
 - boolean "disabled"
 - string "contact_method" - One of the following:
 - default
 - ssh-push

- ssh-push-tunnel

Available since API version: 10.2

Chapter 39. Method: listActivatedSystems

HTTP **GET**

Description:

List the systems activated with the key provided.

Parameters:

- string sessionKey
- string key

Returns:

- array :
 - struct system structure
 - int "id" - System id
 - string "hostname"
 - dateTime.iso8601 "last_checkin" - Last time server successfully checked in

Chapter 40. Method: listActivationKeys

HTTP **GET**

Description:

List activation keys that are visible to the user.

Parameters:

- string sessionKey

Returns:

- array :
- struct activation key
 - string "key"
 - string "description"
 - int "usage_limit"
 - string "base_channel_label"
 - string array "child_channel_labels" - childChannelLabel
 - string array "entitlements" - entitlementLabel
 - string array "server_group_ids" - serverGroupId
 - string array "package_names" - packageName - (deprecated by packages)
 - array "packages"
 - struct package
 - string "name" - packageName
 - string "arch" - archLabel - optional
 - boolean "universal_default"
 - boolean "disabled"
 - string "contact_method" - One of the following:
 - default
 - ssh-push

- ssh-push-tunnel

Available since API version: 10.2

Chapter 41. Method: listChannels

HTTP **GET**

Description:

List the channels for the given activation key with temporary authentication tokens to access them. Authentication is done via a machine specific password.

Parameters:

- string minionId - The id of the minion to authenticate with.
- string machinePassword - password specific to a machine.
- string activationKey - activation key to use channels from.

Returns:

- array :
- struct channelInfo
 - string "label" - Channel label
 - string "name" - Channel name
 - string "url" - Channel url
 - string "token" - Channel access token

Chapter 42. Method: listConfigChannels

HTTP **GET**

Description:

List configuration channels associated to an activation key.

Parameters:

- string sessionKey
- string key

Returns:

- array :
- struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 43. Method: removeChildChannels

HTTP **POST**

Description:

Remove child channels from an activation key.

Parameters:

- string sessionKey
- string key
- string array childChannellLabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 44. Method: removeConfigChannels

HTTP **POST**

Description:

Remove configuration channels from the given activation keys.

Parameters:

- string sessionKey
- string array keys
- string array configChannellLabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 45. Method: removeEntitlements

HTTP **POST**

Description:

Remove entitlements (by label) from an activation key. Currently only `virtualization_host` add-on entitlement is permitted.

Parameters:

- string `sessionKey`
- string `key`
- string array `entitlements`
 - `virtualization_host`

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 46. Method: removePackages

HTTP **POST**

Description:

Remove package names from an activation key.

Parameters:

- string sessionKey
- string key
- array packages
 - struct package
 - string "name" - Package name
 - string "arch" - Arch label - Optional

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 47. Method: removeServerGroups

HTTP **POST**

Description:

Remove server groups from an activation key.

Parameters:

- string sessionKey
- string key
- int array serverGroupIds

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 48. Method: setConfigChannels

HTTP **POST**

Description:

Replace the existing set of configuration channels on the given activation keys. Channels are ranked by their order in the array.

Parameters:

- string sessionKey
- string array keys
- string array configChannelLabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 49. Method: setDetails

HTTP **POST**

Description:

Update the details of an activation key.

Parameters:

- string sessionKey
- string key
- struct details
 - string "description" - optional
 - string "base_channel_label" - optional - to set default base channel set to empty string or 'none'
 - int "usage_limit" - optional
 - boolean "unlimited_usage_limit" - Set true for unlimited usage and to override usage_limit
 - boolean "universal_default" - optional
 - boolean "disabled" - optional
 - string "contact_method" - One of the following:
 - default
 - ssh-push
 - ssh-push-tunnel

Returns:

- int - 1 on success, exception thrown otherwise.

admin

Chapter 50. Available methods

- create
- create
- delete
- getDetails
- list
- setDetails

Chapter 51. Description

Provides methods to access and modify PAYG ssh connection data

Namespace:

admin

Chapter 52. Method: create

HTTP **POST**

Description:

Create a new ssh connection data to extract data from

Parameters:

- string sessionKey
- string description
- string host - hostname or IP address to the instance, will fail if already in use.
- int port
- string username
- string password
- string key - private key to use in authentication
- string keyPassword

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 53. Method: create

HTTP **POST**

Description:

Create a new ssh connection data to extract data from

Parameters:

- string sessionKey
- string description
- string host - hostname or IP address to the instance, will fail if already in use.
- int port
- string username
- string password
- string key - private key to use in authentication
- string keyPassword
- string bastionHost - hostname or IP address to a bastion host
- int bastionPort
- string bastionUsername
- string bastionPassword
- string bastionKey - private key to use in bastion authentication
- string bastionKeyPassword

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 54. Method: delete

HTTP **POST**

Description:

Returns a list of ssh connection data registered.

Parameters:

- string sessionKey
- string host - hostname or IP address of the instance

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 55. Method: getDetails

HTTP **POST**

Description:

Returns a list of ssh connection data registered.

Parameters:

- string sessionKey
- string host - hostname or IP address of the instance, will fail if host doesn't exist.

Returns:

- * struct SSH data
 - string "description"
 - string "hostname"
 - int "port"
 - string "username"
 - string "bastion_hostname"
 - int "bastion_port"
 - string "bastion_username"

Chapter 56. Method: list

HTTP **POST**

Description:

Returns a list of ssh connection data registered.

Parameters:

- string sessionKey

Returns:

- array :
- struct SSH data
 - string "description"
 - string "hostname"
 - int "port"
 - string "username"
 - string "bastion_hostname"
 - int "bastion_port"
 - string "bastion_username"

Chapter 57. Method: setDetails

HTTP **POST**

Description:

Updates the details of a ssh connection data

Parameters:

- string sessionKey
- string host - hostname or IP address to the instance, will fail if host doesn't exist.
- struct details - user details
 - string "description"
 - int "port"
 - string "username"
 - string "password"
 - string "key"
 - string "key_password"
 - string "bastion_host"
 - int "bastion_port"
 - string "bastion_username"
 - string "bastion_password"
 - string "bastion_key"
 - string "bastion_key_password"

Returns:

- int - 1 on success, exception thrown otherwise.

admin.configuration

Chapter 58. Available methods

- `configure`

Chapter 59. Description

Provides methods to configure the `#product()` server.

Namespace:

`admin.configuration`

Chapter 60. Method: configure

HTTP **POST**

Description:

Configure server.

Parameters:

- string sessionKey
- map content - the Uyuni configuration formula data

Returns:

- int - 1 on success, exception thrown otherwise.

admin.monitoring

Chapter 61. Available methods

- `disable`
- `enable`
- `getStatus`

Chapter 62. Description

Provides methods to manage the monitoring of the `#product()` server.

Namespace:

`admin.monitoring`

Chapter 63. Method: disable

HTTP **POST**

Description:

Disable monitoring.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Exporters
 - string "node"
 - string "tomcat"
 - string "taskomatic"
 - string "postgres"
 - string "self_monitoring"

Chapter 64. Method: enable

HTTP **POST**

Description:

Enable monitoring.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Exporters
 - string "node"
 - string "tomcat"
 - string "taskomatic"
 - string "postgres"
 - string "self_monitoring"

Chapter 65. Method: getStatus

HTTP **GET**

Description:

Get the status of each Prometheus exporter.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Exporters
 - string "node"
 - string "tomcat"
 - string "taskomatic"
 - string "postgres"
 - string "self_monitoring"

ansible

Chapter 66. Available methods

- `createAnsiblePath`
- `discoverPlaybooks`
- `fetchPlaybookContents`
- `introspectInventory`
- `listAnsiblePaths`
- `lookupAnsiblePathById`
- `removeAnsiblePath`
- `schedulePlaybook`
- `schedulePlaybook`
- `schedulePlaybook`
- `schedulePlaybook`
- `updateAnsiblePath`

Chapter 67. Description

Provides methods to manage Ansible systems

Namespace:

ansible

Chapter 68. Method: createAnsiblePath

HTTP **POST**

Description:

Create ansible path

Parameters:

- string `sessionKey`
- struct `props`
 - string `"type"` - The ansible path type: 'inventory' or 'playbook'
 - int `"server_id"` - ID of control node server
 - string `"path"` - The local path to inventory/playbook

Returns:

- * struct `ansible path`
 - int `"path id"`
 - string `"type label"`
 - int `"id of the ansible control node system"`
 - string `"local path to inventory or playbook"`

Chapter 69. Method: discoverPlaybooks

HTTP **POST**

Description:

Discover playbooks under given playbook path with given pathId

Parameters:

- string sessionKey
- int pathId - path id

Returns:

- struct playbooks
 - struct playbook
- struct ansible path
 - int "path id"
 - string "type label"
 - int "id of the ansible control node system"
 - string "local path to inventory or playbook"

Chapter 70. Method: fetchPlaybookContents

HTTP **POST**

Description:

Fetch the playbook content from the control node using a synchronous salt call.

Parameters:

- string sessionKey
- int pathId - playbook path id
- string playbookRelPath - relative path of playbook (inside path specified by pathId)

Returns:

- string contents - Text contents of the playbook

Chapter 71. Method: introspectInventory

HTTP **POST**

Description:

Introspect inventory under given inventory path with given pathId and return it in a structured way

Parameters:

- string sessionKey
- int pathId - path id

Returns:

- struct Inventory in a nested structure
 - object Inventory item - Inventory item (can be nested)

Chapter 72. Method: listAnsiblePaths

HTTP **GET**

Description:

List ansible paths for server (control node)

Parameters:

- string sessionKey
- int controlNodeId - id of ansible control node server

Returns:

- array :
- struct ansible path
 - int "path id"
 - string "type label"
 - int "id of the ansible control node system"
 - string "local path to inventory or playbook"

Chapter 73. Method: lookupAnsiblePathById

HTTP **GET**

Description:

Lookup ansible path by path id

Parameters:

- string sessionKey
- int pathId - path id

Returns:

- * struct ansible path
 - int "path id"
 - string "type label"
 - int "id of the ansible control node system"
 - string "local path to inventory or playbook"

Chapter 74. Method: removeAnsiblePath

HTTP **POST**

Description:

Create ansible path

Parameters:

- string sessionKey
- int pathId - path id

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 75. Method: schedulePlaybook

HTTP **POST**

Description:

Schedule a playbook execution

Parameters:

- string sessionKey
- string playbookPath
- string inventoryPath - path to Ansible inventory or empty
- int controlNodeId - system ID of the control node
- dateTime.iso8601 earliestOccurrence - earliest the execution command can be sent to the control node. ignored when actionChainLabel is used
- string actionChainLabel - label of an action chain to use, or None

Returns:

- int id - ID of the playbook execution action created

Chapter 76. Method: schedulePlaybook

HTTP **POST**

Description:

Schedule a playbook execution

Parameters:

- string sessionKey
- string playbookPath
- string inventoryPath - path to Ansible inventory or empty
- int controlNodeId - system ID of the control node
- dateTime.iso8601 earliestOccurrence - earliest the execution command can be sent to the control node. ignored when actionChainLabel is used
- string actionChainLabel - label of an action chain to use, or None
- boolean testMode - 'true' if the playbook shall be executed in test mode

Returns:

- int id - ID of the playbook execution action created

Chapter 77. Method: schedulePlaybook

HTTP **POST**

Description:

Schedule a playbook execution

Parameters:

- string sessionKey
- string playbookPath
- string inventoryPath - path to Ansible inventory or empty
- int controlNodeId - system ID of the control node
- dateTime.iso8601 earliestOccurrence - earliest the execution command can be sent to the control node. ignored when actionChainLabel is used
- string actionChainLabel - label of an action chain to use, or None
- struct ansibleArgs
 - boolean "flushCache"

Returns:

- int id - ID of the playbook execution action created

Chapter 78. Method: schedulePlaybook

HTTP **POST**

Description:

Schedule a playbook execution

Parameters:

- string sessionKey
- string playbookPath
- string inventoryPath - path to Ansible inventory or empty
- int controlNodeId - system ID of the control node
- dateTime.iso8601 earliestOccurrence - earliest the execution command can be sent to the control node. ignored when actionChainLabel is used
- string actionChainLabel - label of an action chain to use, or None
- boolean testMode - 'true' if the playbook shall be executed in test mode
- struct ansibleArgs
 - boolean "flushCache"

Returns:

- int id - ID of the playbook execution action created

Chapter 79. Method: updateAnsiblePath

HTTP **POST**

Description:

Create ansible path

Parameters:

- string sessionKey
- int pathId - path id
- struct props
 - string "path" - The local path to inventory/playbook

Returns:

- * struct ansible path
 - int "path id"
 - string "type label"
 - int "id of the ansible control node system"
 - string "local path to inventory or playbook"

api

Chapter 80. Available methods

- `getApiCallList`
- `getApiNamespaceCallList`
- `getApiNamespaces`
- `getVersion`
- `systemVersion`

Chapter 81. Description

Methods providing information about the API.

Namespace:

api

Chapter 82. Method: getApiCallList

HTTP **GET**

Description:

Lists all available api calls grouped by namespace

Parameters:

- string sessionKey

Returns:

- struct method_info
 - string "name" - method name
 - string "parameters" - method parameters
 - string "exceptions" - method exceptions
 - string "return" - method return type

Chapter 83. Method: getApiNamespaceCallList

HTTP **GET**

Description:

Lists all available api calls for the specified namespace

Parameters:

- string sessionKey
- string namespace

Returns:

- struct method_info
 - string "name" - method name
 - string "parameters" - method parameters
 - string "exceptions" - method exceptions
 - string "return" - method return type

Chapter 84. Method: getApiNamespaces

HTTP **GET**

Description:

Lists available API namespaces

Parameters:

- string sessionKey

Returns:

- struct namespace
 - string "namespace" - API namespace
 - string "handler" - API Handler

Chapter 85. Method: getVersion

HTTP **GET**

Description:

Returns the version of the API.

Parameters:

Returns:

- string version

Chapter 86. Method: systemVersion

HTTP **GET**

Description:

Returns the server version.

Parameters:

Returns:

- string version

SUSE Manager API Documentation

SUSE Manager API version: 27 :homepage: <http://www.suse.com/products/suse-manager>

Welcome to the SUSE Manager API. By using the included API calls, you can easily automate many of the tasks you perform everyday. All API calls are grouped by common functionality.

Chapter 87. Legal Notice

Copyright (c) 2014 Red Hat, Inc. Copyright (c) 2019 SUSE LLC

This software is licensed to you under the GNU General Public License, version 2 (GPLv2). There is NO WARRANTY for this software, express or implied, including the implied warranties of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. You should have received a copy of GPLv2 along with this software; if not, see <http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt>.

Red Hat trademarks are not licensed under GPLv2. No permission is granted to use or replicate Red Hat trademarks that are incorporated in this software or its documentation.

- [actionchain.pdf](#)
- [activationkey.pdf](#)
- [admin.pdf](#)
- [admin.configuration.pdf](#)
- [admin.monitoring.pdf](#)
- [ansible.pdf](#)
- [api.pdf](#)
- [audit.pdf](#)
- [auth.pdf](#)
- [channel.pdf](#)
- [channel.access.pdf](#)
- [channel.appstreams.pdf](#)
- [channel.org.pdf](#)
- [channel.software.pdf](#)
- [configchannel.pdf](#)
- [contentmanagement.pdf](#)
- [distchannel.pdf](#)
- [errata.pdf](#)

-
- [formula.pdf](#)
 - [image.pdf](#)
 - [image.delta.pdf](#)
 - [image.profile.pdf](#)
 - [image.store.pdf](#)
 - [kickstart.pdf](#)
 - [kickstart.filepreservation.pdf](#)
 - [kickstart.keys.pdf](#)
 - [kickstart.profile.pdf](#)
 - [kickstart.profile.keys.pdf](#)
 - [kickstart.profile.software.pdf](#)
 - [kickstart.profile.system.pdf](#)
 - [kickstart.snippet.pdf](#)
 - [kickstart.tree.pdf](#)
 - [maintenance.pdf](#)
 - [org.pdf](#)
 - [org.trusts.pdf](#)
 - [packages.pdf](#)
 - [packages.provider.pdf](#)
 - [packages.search.pdf](#)
 - [preferences.locale.pdf](#)
 - [proxy.pdf](#)
 - [recurring.pdf](#)
 - [recurring.custom.pdf](#)
 - [recurring.highstate.pdf](#)
 - [saltkey.pdf](#)
 - [schedule.pdf](#)

-
- [subscriptionmatching.pinnedsubscription.pdf](#)
 - [sync.content.pdf](#)
 - [sync.master.pdf](#)
 - [sync.slave.pdf](#)
 - [system.pdf](#)
 - [system.appstreams.pdf](#)
 - [system.config.pdf](#)
 - [system.custominfo.pdf](#)
 - [system.monitoring.pdf](#)
 - [system.provisioning.powermanagement.pdf](#)
 - [system.provisioning.snapshot.pdf](#)
 - [system.scap.pdf](#)
 - [system.search.pdf](#)
 - [systemgroup.pdf](#)
 - [user.pdf](#)
 - [user.external.pdf](#)
 - [virtualhostmanager.pdf](#)

Generated on: 5/25/2024

audit

Chapter 88. Available methods

- `listImagesByPatchStatus`
- `listImagesByPatchStatus`
- `listSystemsByPatchStatus`
- `listSystemsByPatchStatus`

Chapter 89. Description

Methods to audit systems.

Namespace:

audit

Chapter 90. Method: listImagesByPatchStatus

HTTP **GET**

Description:

List visible images with their patch status regarding a given CVE identifier. Please note that the query code relies on data that is pre-generated by the 'cve-server-channels' taskomatic job.

Parameters:

- string sessionKey
- string cveldentifier

Returns:

- array :
- struct CVE audit image
 - int "image_id"
 - string "patch_status"
 - **AFFECTED_PATCH_INAPPLICABLE** - affected, patch available in unassigned channel
 - **AFFECTED_PATCH_APPLICABLE** - affected, patch available in assigned channel
 - **NOT_AFFECTED** - not affected
 - **PATCHED** - patched
 - channel_labels array "string" - labels of channels that contain an unapplied patch
 - errata_advisories array "string" - advisories of erratas that patch the specified vulnerability

Chapter 91. Method: listImagesByPatchStatus

HTTP **GET**

Description:

List visible images with their patch status regarding a given CVE identifier. Filter the results by passing in a list of patch status labels. Please note that the query code relies on data that is pre-generated by the 'cve-server-channels' taskomatic job.

Parameters:

- string `sessionKey`
- string `cveIdentifier`
- string array `patchStatusLabels`
 - `AFFECTED_PATCH_INAPPLICABLE` - Affected, patch available in unassigned channel
 - `AFFECTED_PATCH_APPLICABLE` - Affected, patch available in assigned channel
 - `NOT_AFFECTED` - Not affected
 - `PATCHED` - Patched

Returns:

- array :
- struct CVE audit image
 - int `"image_id"`
 - string `"patch_status"`
 - `AFFECTED_PATCH_INAPPLICABLE` - affected, patch available in unassigned channel
 - `AFFECTED_PATCH_APPLICABLE` - affected, patch available in assigned channel
 - `NOT_AFFECTED` - not affected
 - `PATCHED` - patched
 - `channel_labels` array "string" - labels of channels that contain an unapplied patch
 - `errata_advisories` array "string" - advisories of erratas that patch the specified vulnerability

Chapter 92. Method: listSystemsByPatchStatus

HTTP **GET**

Description:

List visible systems with their patch status regarding a given CVE identifier. Please note that the query code relies on data that is pre-generated by the 'cve-server-channels' taskomatic job.

Parameters:

- string `sessionKey`
- string `cveIdentifier`

Returns:

- array :
- struct CVE audit system
 - int "system_id"
 - string "patch_status"
 - `AFFECTED_PATCH_INAPPLICABLE` - affected, patch available in unassigned channel
 - `AFFECTED_PATCH_APPLICABLE` - affected, patch available in assigned channel
 - `NOT_AFFECTED` - not affected
 - `PATCHED` - patched
 - `channel_labels` array "string" - labels of channels that contain an unapplied patch
 - `errata_advisories` array "string" - advisories of erratas that patch the specified vulnerability

Chapter 93. Method:

listSystemsByPatchStatus

HTTP **GET**

Description:

List visible systems with their patch status regarding a given CVE identifier. Filter the results by passing in a list of patch status labels. Please note that the query code relies on data that is pre-generated by the 'cve-server-channels' taskomatic job.

Parameters:

- string `sessionKey`
- string `cveIdentifier`
- string array `patchStatusLabels`
 - `AFFECTED_PATCH_INAPPLICABLE` - Affected, patch available in unassigned channel
 - `AFFECTED_PATCH_APPLICABLE` - Affected, patch available in assigned channel
 - `NOT_AFFECTED` - Not affected
 - `PATCHED` - Patched

Returns:

- array :
- struct CVE audit system
 - int `"system_id"`
 - string `"patch_status"`
 - `AFFECTED_PATCH_INAPPLICABLE` - affected, patch available in unassigned channel
 - `AFFECTED_PATCH_APPLICABLE` - affected, patch available in assigned channel
 - `NOT_AFFECTED` - not affected
 - `PATCHED` - patched
 - `channel_labels` array "string" - labels of channels that contain an unapplied patch
 - `errata_advisories` array "string" - advisories of erratas that patch the specified vulnerability

auth

Chapter 94. Available methods

- login
- login
- logout

Chapter 95. Description

This namespace provides methods to authenticate with the system's management server.

Namespace:

auth

Chapter 96. Method: login

HTTP **POST**

Description:

Login using a username and password. Returns the session key used by most other API methods.

Parameters:

- string username
- string password

Returns:

- string sessionKey

Chapter 97. Method: login

HTTP **POST**

Description:

Login using a username and password. Returns the session key used by other methods.

Parameters:

- string username
- string password
- int duration - Length of session.

Returns:

- string sessionKey

Chapter 98. Method: logout

HTTP **POST**

Description:

Logout the user with the given session key.

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

channel.access

Chapter 99. Available methods

- `disableUserRestrictions`
- `enableUserRestrictions`
- `getOrgSharing`
- `setOrgSharing`

Chapter 100. Description

Provides methods to retrieve and alter channel access restrictions.

Namespace:

channel.access

Chapter 101. Method: disableUserRestrictions

HTTP **POST**

Description:

Disable user restrictions for the given channel. If disabled, all users within the organization may subscribe to the channel.

Parameters:

- string sessionKey
- string channelLabel - label of the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 102. Method: enableUserRestrictions

HTTP **POST**

Description:

Enable user restrictions for the given channel. If enabled, only selected users within the organization may subscribe to the channel.

Parameters:

- string sessionKey
- string channelLabel - label of the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 103. Method: getOrgSharing

HTTP **GET**

Description:

Get organization sharing access control.

Parameters:

- string sessionKey
- string channelLabel - label of the channel

Returns:

- string access - The access value (one of the following: 'public', 'private', or 'protected')

Chapter 104. Method: setOrgSharing

HTTP **POST**

Description:

Set organization sharing access control.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string access - Access (one of the following: 'public', 'private', or 'protected')

Returns:

- int - 1 on success, exception thrown otherwise.

channel

Chapter 105. Available methods

- `listAllChannels`
- `listManageableChannels`
- `listMyChannels`
- `listPopularChannels`
- `listRetiredChannels`
- `listSharedChannels`
- `listSoftwareChannels`
- `listVendorChannels`

Chapter 106. Description

Provides method to get back a list of Software Channels.

Namespace:

channel

Chapter 107. Method: listAllChannels

HTTP **GET**

Description:

List all software channels that the user's organization is entitled to.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 108. Method: listManageableChannels

HTTP **GET**

Description:

List all software channels that the user is entitled to manage.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 109. Method: listMyChannels

HTTP **GET**

Description:

List all software channels that belong to the user's organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 110. Method: listPopularChannels

HTTP **GET**

Description:

List the most popular software channels. Channels that have at least the number of systems subscribed as specified by the popularity count will be returned.

Parameters:

- string sessionKey
- int popularityCount

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 111. Method: listRetiredChannels

HTTP **GET**

Description:

List all retired software channels. These are channels that the user's organization is entitled to, but are no longer supported because they have reached their 'end-of-life' date.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 112. Method: listSharedChannels

HTTP **GET**

Description:

List all software channels that may be shared by the user's organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

Chapter 113. Method: listSoftwareChannels

HTTP **GET**

Description:

List all visible software channels.

Parameters:

- string sessionKey

Returns:

- array :
 - struct channel
 - string "label"
 - string "name"
 - string "parent_label"
 - string "end_of_life"
 - string "arch"

Chapter 114. Method: listVendorChannels

HTTP **GET**

Description:

Lists all the vendor software channels that the user's organization is entitled to.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel info
 - int "id"
 - string "label"
 - string "name"
 - string "provider_name"
 - int "packages"
 - int "systems"
 - string "arch_name"

channel.appstreams

Chapter 115. Available methods

- `isModular`
- `listModular`
- `listModuleStreams`

Chapter 116. Description

Provides methods to handle appstreams for channels.

Namespace:

channel.appstreams

Chapter 117. Method: isModular

HTTP **GET**

Description:

Check if channel is modular.

Parameters:

- string sessionKey
- string channelLabel

Returns:

- boolean result - true if the channel is modular

Chapter 118. Method: listModular

HTTP **GET**

Description:

List modular channels in users organization.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `channel`
 - int `"id"`
 - string `"name"`
 - string `"label"`
 - string `"arch_name"`
 - string `"arch_label"`
 - string `"summary"`
 - string `"description"`
 - string `"checksum_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"maintainer_name"`
 - string `"maintainer_email"`
 - string `"maintainer_phone"`
 - string `"support_policy"`
 - string `"gpg_key_url"`
 - string `"gpg_key_id"`
 - string `"gpg_key_fp"`
 - `dateTime.iso8601` `"yumrepo_last_sync"` - (optional)

-
- string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 119. Method: listModuleStreams

HTTP **GET**

Description:

List available module streams for a given channel.

Parameters:

- string sessionKey
- string channelLabel

Returns:

- array :
- struct AppStream
 - string "stream"
 - string "module"
 - string "arch"

channel.org

Chapter 120. Available methods

- `disableAccess`
- `enableAccess`
- `list`

Chapter 121. Description

Provides methods to retrieve and alter organization trust relationships for a channel.

Namespace:

channel.org

Chapter 122. Method: disableAccess

HTTP **POST**

Description:

Disable access to the channel for the given organization.

Parameters:

- string sessionKey
- string label - label of the channel
- int orgId - ID of org being removed access

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 123. Method: enableAccess

HTTP **POST**

Description:

Enable access to the channel for the given organization.

Parameters:

- string sessionKey
- string label - label of the channel
- int orgId - ID of org being granted access

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 124. Method: list

HTTP **POST**

Description:

List the organizations associated with the given channel that may be trusted.

Parameters:

- string sessionKey
- string label - label of the channel

Returns:

- array :
 - struct org
 - int "org_id"
 - string "org_name"
 - boolean "access_enabled"

channel.software

Chapter 125. Available methods

- `addPackages`
- `addRepoFilter`
- `alignMetadata`
- `applyChannelState`
- `associateRepo`
- `clearRepoFilters`
- `clone`
- `create`
- `create`
- `create`
- `create`
- `createRepo`
- `createRepo`
- `createRepo`
- `delete`
- `disassociateRepo`
- `getChannelLastBuildById`
- `getDetails`
- `getDetails`
- `getRepoDetails`
- `getRepoDetails`
- `getRepoSyncCronExpression`
- `isExisting`
- `isGloballySubscribable`
- `isUserManageable`

-
- `isUserSubscribable`
 - `listAllPackages`
 - `listAllPackages`
 - `listAllPackages`
 - `listArches`
 - `listChannelRepos`
 - `listChildren`
 - `listErrata`
 - `listErrata`
 - `listErrata`
 - `listErrata`
 - `listErrata`
 - `listErrataByType`
 - `listErrataNeedingSync`
 - `listLatestPackages`
 - `listPackagesWithoutChannel`
 - `listRepoFilters`
 - `listSubscribedSystems`
 - `listSystemChannels`
 - `listUserRepos`
 - `mergeErrata`
 - `mergeErrata`
 - `mergeErrata`
 - `mergePackages`
 - `mergePackages`
 - `regenerateNeededCache`
 - `regenerateNeededCache`
 - `regenerateYumCache`

-
- `removeErrata`
 - `removePackages`
 - `removeRepo`
 - `removeRepo`
 - `removeRepoFilter`
 - `setContactDetails`
 - `setDetails`
 - `setDetails`
 - `setGloballySubscribable`
 - `setRepoFilters`
 - `setUserManageable`
 - `setUserSubscribable`
 - `syncErrata`
 - `syncRepo`
 - `syncRepo`
 - `syncRepo`
 - `syncRepo`
 - `syncRepo`
 - `syncRepo`
 - `updateRepo`
 - `updateRepoLabel`
 - `updateRepoLabel`
 - `updateRepoSsl`
 - `updateRepoSsl`
 - `updateRepoUrl`
 - `updateRepoUrl`

Chapter 126. Description

Provides methods to access and modify many aspects of a channel.

Namespace:

channel.software

Chapter 127. Method: addPackages

HTTP **POST**

Description:

Adds a given list of packages to the given channel.

Parameters:

- string sessionKey
- string channelLabel - target channel
- int array packageIds - ID of a package to add to the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 128. Method: addRepoFilter

HTTP **POST**

Description:

Adds a filter for a given repo.

Parameters:

- string sessionKey
- string label - repository label
- struct filterProps
 - string "filter" - string to filter on
 - string "flag" - + for include, - for exclude

Returns:

- int order - sort order for new filter

Chapter 129. Method: alignMetadata

HTTP **POST**

Description:

Align the metadata of a channel to another channel.

Parameters:

- string sessionKey
- string channelFromLabel - the label of the source channel
- string channelToLabel - the label of the target channel
- string metadataType - the metadata type. Only 'modules' supported currently.

Returns:

- int result code - 1 when metadata has been aligned, 0 otherwise

Chapter 130. Method: applyChannelState

HTTP **POST**

Description:

Refresh pillar data and then schedule channels state on the provided systems

Parameters:

- string sessionKey
- int array sids

Returns:

- int array actionId

Chapter 131. Method: associateRepo

HTTP **POST**

Description:

Associates a repository with a channel

Parameters:

- string sessionKey
- string channelLabel - channel label
- string repoLabel - repository label

Returns:

- * struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"
 - string "gpg_key_fp"

-
- dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 132. Method: clearRepoFilters

HTTP **POST**

Description:

Removes the filters for a repo

Parameters:

- string sessionKey
- string label - repository label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 133. Method: clone

HTTP **POST**

Description:

Clone a channel. If `arch_label` is omitted, the arch label of the original channel will be used. If `parent_label` is omitted, the clone will be a base channel.

Parameters:

- string `sessionKey`
- string `originalLabel`
- struct `channelDetails`
 - string `"name"`
 - string `"label"`
 - string `"summary"`
 - string `"parent_label"` - (optional)
 - string `"arch_label"` - (optional)
 - string `"gpg_key_url"` - (optional), `gpg_url` might be used as well
 - string `"gpg_key_id"` - (optional), `gpg_id` might be used as well
 - string `"gpg_key_fp"` - (optional), `gpg_fingerprint` might be used as well
 - string `"gpg_check"` - (optional)
 - string `"description"` - (optional)
 - string `"checksum"` - either `sha1` or `sha256`
- boolean `originalState`

Returns:

- int `id` - the cloned channel ID

Chapter 134. Method: create

HTTP **POST**

Description:

Creates a software channel

Parameters:

- string `sessionKey`
- string `label` - label of the new channel
- string `name` - name of the new channel
- string `summary` - summary of the channel
- string `archLabel` - the label of the architecture the channel corresponds to, run `channel.software.listArches` API for complete listing
- string `parentLabel` - label of the parent of this channel, an empty string if it does not have one
- string `checksumType` - checksum type for this channel, used for yum repository metadata generation
 - `sha1` - offers widest compatibility with clients
 - `sha256` - offers highest security, but is compatible only with newer clients: Fedora 11 and newer, or Enterprise Linux 6 and newer.
- struct `gpgKey`
 - string `"url"` - GPG key URL
 - string `"id"` - GPG key ID
 - string `"fingerprint"` - GPG key Fingerprint
- boolean `gpgCheck` - true if the GPG check should be enabled by default, false otherwise

Returns:

- int `status` - 1 if the creation operation succeeded, 0 otherwise

Available since API version: 10.9

Chapter 135. Method: create

HTTP **POST**

Description:

Creates a software channel

Parameters:

- string `sessionKey`
- string `label` - label of the new channel
- string `name` - name of the new channel
- string `summary` - summary of the channel
- string `archLabel` - the label of the architecture the channel corresponds to, run `channel.software.listArches` API for complete listing
- string `parentLabel` - label of the parent of this channel, an empty string if it does not have one
- string `checksumType` - checksum type for this channel, used for yum repository metadata generation
 - `sha1` - offers widest compatibility with clients
 - `sha256` - offers highest security, but is compatible only with newer clients: Fedora 11 and newer, or Enterprise Linux 6 and newer.
- struct `gpgKey`
 - string `"url"` - GPG key URL
 - string `"id"` - GPG key ID
 - string `"fingerprint"` - GPG key Fingerprint

Returns:

- int `status` - 1 if the creation operation succeeded, 0 otherwise

Available since API version: 10.9

Chapter 136. Method: create

HTTP **POST**

Description:

Creates a software channel

Parameters:

- string `sessionKey`
- string `label` - label of the new channel
- string `name` - name of the new channel
- string `summary` - summary of the channel
- string `archLabel` - the label of the architecture the channel corresponds to, run `channel.software.listArches` API for complete listing
- string `parentLabel` - label of the parent of this channel, an empty string if it does not have one
- string `checksumType` - checksum type for this channel, used for yum repository metadata generation
 - `sha1` - offers widest compatibility with clients
 - `sha256` - offers highest security, but is compatible only with newer clients: Fedora 11 and newer, or Enterprise Linux 6 and newer.

Returns:

- int `status` - 1 if the creation operation succeeded, 0 otherwise

Available since API version: 10.9

Chapter 137. Method: create

HTTP **POST**

Description:

Creates a software channel

Parameters:

- string `sessionKey`
- string `label` - label of the new channel
- string `name` - name of the new channel
- string `summary` - summary of the channel
- string `archLabel` - the label of the architecture the channel corresponds to, run `channel.software.listArches` API for complete listing
- string `parentLabel` - label of the parent of this channel, an empty string if it does not have one

Returns:

- int `status` - 1 if the creation operation succeeded, 0 otherwise

Chapter 138. Method: createRepo

HTTP **POST**

Description:

Creates a repository

Parameters:

- string sessionKey
- string label - repository label
- string type - repository type (yum, uln...)
- string url - repository url

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 139. Method: createRepo

HTTP **POST**

Description:

Creates a repository

Parameters:

- string sessionKey
- string label - repository label
- string type - repository type (yum, uln...)
- string url - repository url
- string sslCaCert - SSL CA cert description
- string sslCliCert - SSL Client cert description
- string sslCliKey - SSL Client key description

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 140. Method: createRepo

HTTP **POST**

Description:

Creates a repository

Parameters:

- string `sessionKey`
- string `label` - repository label
- string `type` - repository type (only YUM is supported)
- string `url` - repository url
- string `sslCaCert` - SSL CA cert description, or an empty string
- string `sslCliCert` - SSL Client cert description, or an empty string
- string `sslCliKey` - SSL Client key description, or an empty string
- boolean `hasSignedMetadata` - true if the repository has signed metadata, false otherwise

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 141. Method: delete

HTTP **POST**

Description:

Deletes a custom software channel

Parameters:

- string sessionKey
- string channelLabel - channel to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 142. Method: disassociateRepo

HTTP **POST**

Description:

Disassociates a repository from a channel

Parameters:

- string sessionKey
- string channelLabel - channel label
- string repoLabel - repository label

Returns:

- * struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"
 - string "gpg_key_fp"

-
- dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 143. Method: getChannelLastBuildById

HTTP **GET**

Description:

Returns the last build date of the repomd.xml file for the given channel as a localised string.

Parameters:

- string sessionKey
- int id - id of channel wanted

Returns:

- date date - the last build date of the repomd.xml file as a localised string

Chapter 144. Method: getDetails

HTTP **GET**

Description:

Returns details of the given channel as a map

Parameters:

- string `sessionKey`
- string `channelLabel` - channel to query

Returns:

- * struct `channel`
 - int `"id"`
 - string `"name"`
 - string `"label"`
 - string `"arch_name"`
 - string `"arch_label"`
 - string `"summary"`
 - string `"description"`
 - string `"checksum_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"maintainer_name"`
 - string `"maintainer_email"`
 - string `"maintainer_phone"`
 - string `"support_policy"`
 - string `"gpg_key_url"`
 - string `"gpg_key_id"`
 - string `"gpg_key_fp"`
 - `dateTime.iso8601` `"yumrepo_last_sync"` - (optional)

-
- string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 145. Method: getDetails

HTTP **GET**

Description:

Returns details of the given channel as a map

Parameters:

- string `sessionKey`
- int `id` - channel to query

Returns:

- * struct `channel`
 - int `"id"`
 - string `"name"`
 - string `"label"`
 - string `"arch_name"`
 - string `"arch_label"`
 - string `"summary"`
 - string `"description"`
 - string `"checksum_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"maintainer_name"`
 - string `"maintainer_email"`
 - string `"maintainer_phone"`
 - string `"support_policy"`
 - string `"gpg_key_url"`
 - string `"gpg_key_id"`
 - string `"gpg_key_fp"`
 - `dateTime.iso8601` `"yumrepo_last_sync"` - (optional)

-
- string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 146. Method: getRepoDetails

HTTP **GET**

Description:

Returns details of the given repository

Parameters:

- string sessionKey
- string repoLabel - repo to query

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 147. Method: getRepoDetails

HTTP **GET**

Description:

Returns details of the given repository

Parameters:

- string sessionKey
- int id - repository ID

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 148. Method: getRepoSyncCronExpression

HTTP **GET**

Description:

Returns repo synchronization cron expression

Parameters:

- string sessionKey
- string channelLabel - channel label

Returns:

- string expression - quartz expression

Chapter 149. Method: isExisting

HTTP **GET**

Description:

Returns whether is existing

Parameters:

- string sessionKey
- string channelLabel - label of the channel

Returns:

- boolean result - true if the channel exists

Chapter 150. Method: isGloballySubscribable

HTTP **GET**

Description:

Returns whether the channel is subscribable by any user in the organization

Parameters:

- string sessionKey
- string channelLabel - channel to query

Returns:

- int subscribable - 1 if true, 0 otherwise

Chapter 151. Method: isUserManageable

HTTP **GET**

Description:

Returns whether the channel may be managed by the given user.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string login - login of the target user

Returns:

- int status - 1 if manageable, 0 if not

Chapter 152. Method: isUserSubscribable

HTTP **GET**

Description:

Returns whether the channel may be subscribed to by the given user.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string login - login of the target user

Returns:

- int status - 1 if subscribable, 0 if not

Chapter 153. Method: listAllPackages

HTTP **GET**

Description:

Lists all packages in the channel, regardless of package version, between the given dates.

Parameters:

- string `sessionKey`
- string `channelLabel` - channel to query
- `dateTime.iso8601` `startDate`
- `dateTime.iso8601` `endDate`

Returns:

- array :
- struct `package`
 - string `"name"`
 - string `"version"`
 - string `"release"`
 - string `"epoch"`
 - string `"checksum"`
 - string `"checksum_type"`
 - int `"id"`
 - string `"arch_label"`
 - string `"last_modified_date"`
 - string `"last_modified"` - (deprecated)

Chapter 154. Method: listAllPackages

HTTP **GET**

Description:

Lists all packages in the channel, regardless of version whose last modified date is greater than given date.

Parameters:

- string sessionKey
- string channelLabel - channel to query
- dateTime.iso8601 startDate

Returns:

- array :
- struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - string "checksum"
 - string "checksum_type"
 - int "id"
 - string "arch_label"
 - string "last_modified_date"
 - string "last_modified" - (deprecated)

Chapter 155. Method: listAllPackages

HTTP **GET**

Description:

Lists all packages in the channel, regardless of the package version

Parameters:

- string sessionKey
- string channelLabel - channel to query

Returns:

- array :
- struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - string "checksum"
 - string "checksum_type"
 - int "id"
 - string "arch_label"
 - string "last_modified_date"
 - string "last_modified" - (deprecated)

Chapter 156. Method: listArches

HTTP **GET**

Description:

Lists the potential software channel architectures that can be created

Parameters:

- string sessionKey

Returns:

- array :
- struct channel arch
 - string "name"
 - string "label"

Chapter 157. Method: listChannelRepos

HTTP **GET**

Description:

Lists associated repos with the given channel

Parameters:

- string sessionKey
- string channelLabel - channel label

Returns:

- array :
- struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 158. Method: listChildren

HTTP **GET**

Description:

List the children of a channel

Parameters:

- string `sessionKey`
- string `channelLabel` - the label of the channel

Returns:

- array :
- struct `channel`
 - int `"id"`
 - string `"name"`
 - string `"label"`
 - string `"arch_name"`
 - string `"arch_label"`
 - string `"summary"`
 - string `"description"`
 - string `"checksum_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"maintainer_name"`
 - string `"maintainer_email"`
 - string `"maintainer_phone"`
 - string `"support_policy"`
 - string `"gpg_key_url"`
 - string `"gpg_key_id"`
 - string `"gpg_key_fp"`

-
- dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 159. Method: listErrata

HTTP **GET**

Description:

List the errata applicable to a channel after given startDate

Parameters:

- string sessionKey
- string channelLabel - channel to query
- dateTime.iso8601 startDate

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 160. Method: listErrata

HTTP **GET**

Description:

List the errata applicable to a channel between startDate and endDate.

Parameters:

- string sessionKey
- string channelLabel - channel to query
- dateTime.iso8601 startDate
- dateTime.iso8601 endDate

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 161. Method: listErrata

HTTP **GET**

Description:

List the errata applicable to a channel between startDate and endDate.

Parameters:

- string sessionKey
- string channelLabel - channel to query
- dateTime.iso8601 startDate
- dateTime.iso8601 endDate
- boolean lastModified - select by last modified or not

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 162. Method: listErrata

HTTP **GET**

Description:

List the errata applicable to a channel

Parameters:

- string sessionKey
- string channelLabel - channel to query

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 163. Method: listErrataByType

HTTP **GET**

Description:

List the errata of a specific type that are applicable to a channel

Parameters:

- string `sessionKey`
- string `channelLabel` - channel to query
- string `advisoryType` - type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', 'Bug Fix Advisory')

Returns:

- array :
 - struct `errata`
 - string `"advisory"` - name of the advisory
 - string `"issue_date"` - date format follows YYYY-MM-DD HH24:MI:SS
 - string `"update_date"` - date format follows YYYY-MM-DD HH24:MI:SS
 - string `"synopsis"`
 - string `"advisory_type"`
 - string `"last_modified_date"` - date format follows YYYY-MM-DD HH24:MI:SS

Chapter 164. Method: listErrataNeedingSync

HTTP **GET**

Description:

If you have synced a new channel then patches will have been updated with the packages that are in the newly synced channel. A cloned erratum will not have been automatically updated however. If you cloned a channel that includes those cloned errata and should include the new packages, they will not be included when they should. This method lists the errata that will be updated if you run the syncErrata method.

Parameters:

- string sessionKey
- string channelLabel - channel to update

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 165. Method: listLatestPackages

HTTP **GET**

Description:

Lists the packages with the latest version (including release and epoch) for the given channel

Parameters:

- string sessionKey
- string channelLabel - channel to query

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"

Chapter 166. Method: listPackagesWithoutChannel

HTTP **GET**

Description:

Lists all packages that are not associated with a channel. Typically these are custom packages.

Parameters:

- string sessionKey

Returns:

- array :
- struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"
 - dateTime.iso8601 "last_modified"
 - string "path" - the path on that file system that the package resides
 - boolean "part_of_retracted_patch" - true if the package is a part of a retracted patch
 - string "provider" - the provider of the package, determined by the gpg key it was signed with.

Chapter 167. Method: listRepoFilters

HTTP **GET**

Description:

Lists the filters for a repo

Parameters:

- string sessionKey
- string label - repository label

Returns:

- array :
- struct filter
 - int "sortOrder"
 - string "filter"
 - string "flag"

Chapter 168. Method: listSubscribedSystems

HTTP **GET**

Description:

Returns list of subscribed systems for the given channel label

Parameters:

- string sessionKey
- string channelLabel - channel to query

Returns:

- array :
 - struct system
 - int "id"
 - string "name"

Chapter 169. Method: listSystemChannels

HTTP **GET**

Description:

Returns a list of channels that a system is subscribed to for the given system id

Parameters:

- string sessionKey
- int sid – system ID

Returns:

- array :
 - struct channel
 - string "id"
 - string "label"
 - string "name"

Chapter 170. Method: listUserRepos

HTTP **GET**

Description:

Returns a list of ContentSource (repos) that the user can see

Parameters:

- string sessionKey

Returns:

- array :
 - struct map
 - long "id" - ID of the repo
 - string "label" - label of the repo
 - string "sourceUrl" - URL of the repo

Chapter 171. Method: mergeErrata

HTTP **POST**

Description:

Merges all errata from one channel into another

Parameters:

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull errata from
- string `mergeToLabel` - the label to push the errata into

Returns:

- array :
- struct `errata`
 - int `"id"` - errata ID
 - string `"date"` - the date erratum was created
 - string `"advisory_type"` - type of the advisory
 - string `"advisory_status"` - status of the advisory
 - string `"advisory_name"` - name of the advisory
 - string `"advisory_synopsis"` - summary of the erratum

Chapter 172. Method: mergeErrata

HTTP **POST**

Description:

Merges all errata from one channel into another based upon a given start/end date.

Parameters:

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull errata from
- string `mergeToLabel` - the label to push the errata into
- string `startDate`
- string `endDate`

Returns:

- array :
- struct `errata`
 - int `"id"` - errata ID
 - string `"date"` - the date erratum was created
 - string `"advisory_type"` - type of the advisory
 - string `"advisory_status"` - status of the advisory
 - string `"advisory_name"` - name of the advisory
 - string `"advisory_synopsis"` - summary of the erratum

Chapter 173. Method: mergeErrata

HTTP **POST**

Description:

Merges a list of errata from one channel into another

Parameters:

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull errata from
- string `mergeToLabel` - the label to push the errata into
- string array `errataNames` - the advisory name of the errata to merge

Returns:

- array :
- struct `errata`
 - int `"id"` - errata ID
 - string `"date"` - the date erratum was created
 - string `"advisory_type"` - type of the advisory
 - string `"advisory_status"` - status of the advisory
 - string `"advisory_name"` - name of the advisory
 - string `"advisory_synopsis"` - summary of the erratum

Chapter 174. Method: mergePackages

HTTP **POST**

Description:

Merges all packages from one channel into another

Parameters:

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull packages from
- string `mergeToLabel` - the label to push the packages into

Returns:

- array :
- struct `package`
 - string `"name"`
 - string `"version"`
 - string `"release"`
 - string `"epoch"`
 - int `"id"`
 - string `"arch_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"path"` - the path on that file system that the package resides
 - boolean `"part_of_retracted_patch"` - true if the package is a part of a retracted patch
 - string `"provider"` - the provider of the package, determined by the gpg key it was signed with.

Chapter 175. Method: mergePackages

HTTP **POST**

Description:

Merges all packages from one channel into another

Parameters:

- string `sessionKey`
- string `mergeFromLabel` - the label of the channel to pull packages from
- string `mergeToLabel` - the label to push the packages into
- boolean `alignModules` - align modular data of the target channel to the source channel (RHEL8 and higher)

Returns:

- array :
- struct `package`
 - string `"name"`
 - string `"version"`
 - string `"release"`
 - string `"epoch"`
 - int `"id"`
 - string `"arch_label"`
 - `dateTime.iso8601` `"last_modified"`
 - string `"path"` - the path on that file system that the package resides
 - boolean `"part_of_retracted_patch"` - true if the package is a part of a retracted patch
 - string `"provider"` - the provider of the package, determined by the gpg key it was signed with.

Chapter 176. Method: regenerateNeededCache

HTTP **POST**

Description:

Completely clear and regenerate the needed Errata and Package cache for all systems subscribed to the specified channel. This should be used only if you believe your cache is incorrect for all the systems in a given channel. This will schedule an asynchronous action to actually do the processing.

Parameters:

- string `sessionKey`
- string `channelLabel` - the label of the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 177. Method: regenerateNeededCache

HTTP **POST**

Description:

Completely clear and regenerate the needed Errata and Package cache for all systems subscribed. You must be a `#product()` Admin to perform this action. This will schedule an asynchronous action to actually do the processing.

Parameters:

- string `sessionKey`

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 178. Method: regenerateYumCache

HTTP **POST**

Description:

Regenerate yum cache for the specified channel.

Parameters:

- string sessionKey
- string channelLabel - the label of the channel
- boolean force - force cache regeneration

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 179. Method: removeErrata

HTTP **POST**

Description:

Removes a given list of errata from the given channel.

Parameters:

- string sessionKey
- string channelLabel - target channel
- string array errataNames - name of an erratum to remove
- boolean removePackages - true to remove packages from the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 180. Method: removePackages

HTTP **POST**

Description:

Removes a given list of packages from the given channel.

Parameters:

- string sessionKey
- string channellabel - target channel
- int array packageids - ID of a package to remove from the channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 181. Method: removeRepo

HTTP **POST**

Description:

Removes a repository

Parameters:

- string sessionKey
- long id - ID of repo to be removed

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 182. Method: removeRepo

HTTP **POST**

Description:

Removes a repository

Parameters:

- string sessionKey
- string label - label of repo to be removed

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 183. Method: removeRepoFilter

HTTP **POST**

Description:

Removes a filter for a given repo.

Parameters:

- string sessionKey
- string label - repository label
- struct filterProps
 - string "filter" - string to filter on
 - string "flag" - + for include, - for exclude

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 184. Method: setContactDetails

HTTP **POST**

Description:

Set contact/support information for given channel.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string maintainerName - name of the channel maintainer
- string maintainerEmail - email of the channel maintainer
- string maintainerPhone - phone number of the channel maintainer
- string supportPolicy - channel support policy

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 185. Method: setDetails

HTTP **POST**

Description:

Allows to modify channel attributes

Parameters:

- string sessionKey
- string channelLabel - channel label
- struct details
 - string "checksum_label" - new channel repository checksum label (optional)
 - string "name" - new channel name (optional)
 - string "summary" - new channel summary (optional)
 - string "description" - new channel description (optional)
 - string "maintainer_name" - new channel maintainer name (optional)
 - string "maintainer_email" - new channel email address (optional)
 - string "maintainer_phone" - new channel phone number (optional)
 - string "gpg_key_url" - new channel gpg key url (optional)
 - string "gpg_key_id" - new channel gpg key id (optional)
 - string "gpg_key_fp" - new channel gpg key fingerprint (optional)
 - string "gpg_check" - enable/disable gpg check (optional)

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 186. Method: setDetails

HTTP **POST**

Description:

Allows to modify channel attributes

Parameters:

- string sessionKey
- int channelId - channel id
- struct details
 - string "checksum_label" - new channel repository checksum label (optional)
 - string "name" - new channel name (optional)
 - string "summary" - new channel summary (optional)
 - string "description" - new channel description (optional)
 - string "maintainer_name" - new channel maintainer name (optional)
 - string "maintainer_email" - new channel email address (optional)
 - string "maintainer_phone" - new channel phone number (optional)
 - string "gpg_key_url" - new channel gpg key url (optional)
 - string "gpg_key_id" - new channel gpg key id (optional)
 - string "gpg_key_fp" - new channel gpg key fingerprint (optional)
 - string "gpg_check" - enable/disable gpg check (optional)

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 187. Method: setGloballySubscribable

HTTP **POST**

Description:

Set globally subscribable attribute for given channel.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- boolean value - true if the channel is to be globally subscribable. False otherwise.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 188. Method: setRepoFilters

HTTP **POST**

Description:

Replaces the existing set of filters for a given repo. Filters are ranked by their order in the array.

Parameters:

- string sessionKey
- string label - repository label
- array filterProps
 - struct filter properties
 - string "filter" - string to filter on
 - string "flag" - + for include, - for exclude

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 189. Method: setUserManageable

HTTP **POST**

Description:

Set the manageable flag for a given channel and user. If value is set to 'true', this method will give the user manage permissions to the channel. Otherwise, that privilege is revoked.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string login - login of the target user
- boolean value - value of the flag to set

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 190. Method: setUserSubscribable

HTTP **POST**

Description:

Set the subscribable flag for a given channel and user. If value is set to 'true', this method will give the user subscribe permissions to the channel. Otherwise, that privilege is revoked.

Parameters:

- string sessionKey
- string channelLabel - label of the channel
- string login - login of the target user
- boolean value - value of the flag to set

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 191. Method: syncErrata

HTTP **POST**

Description:

If you have synced a new channel then patches will have been updated with the packages that are in the newly synced channel. A cloned erratum will not have been automatically updated however. If you cloned a channel that includes those cloned errata and should include the new packages, they will not be included when they should. This method updates all the errata in the given cloned channel with packages that have recently been added, and ensures that all the packages you expect are in the channel. It also updates cloned errata attributes like advisoryStatus.

Parameters:

- string sessionKey
- string channelLabel - channel to update

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 192. Method: syncRepo

HTTP **POST**

Description:

Trigger immediate repo synchronization

Parameters:

- string sessionKey
- string array channellabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 193. Method: syncRepo

HTTP **POST**

Description:

Trigger immediate repo synchronization

Parameters:

- string sessionKey
- string channelLabel - channel label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 194. Method: syncRepo

HTTP **POST**

Description:

Trigger immediate repo synchronization

Parameters:

- string sessionKey
- string channelLabel - channel label
- struct params
 - boolean "sync-kickstart" - create kickstartable tree - Optional
 - boolean "no-errata" - do not sync errata - Optional
 - boolean "fail" - terminate upon any error - Optional
 - boolean "latest" - only download latest packages - Optional

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 195. Method: syncRepo

HTTP **POST**

Description:

Schedule periodic repo synchronization

Parameters:

- string sessionKey
- string channelLabel - channel label
- string cronExpr - cron expression, if empty all periodic schedules will be disabled

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 196. Method: syncRepo

HTTP **POST**

Description:

Schedule periodic repo synchronization

Parameters:

- string sessionKey
- string channelLabel - channel label
- string cronExpr - cron expression, if empty all periodic schedules will be disabled
- struct params
 - boolean "sync-kickstart" - create kickstartable tree - Optional
 - boolean "no-errata" - do not sync errata - Optional
 - boolean "fail" - terminate upon any error - Optional
 - boolean "latest" - only download latest packages - Optional

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 197. Method: updateRepo

HTTP **POST**

Description:

Updates a ContentSource (repo)

Parameters:

- string sessionKey
- int id - repository ID
- string label - new repository label
- string url - new repository URL

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 198. Method: updateRepoLabel

HTTP **POST**

Description:

Updates repository label

Parameters:

- string sessionKey
- int id - repository ID
- string label - new repository label

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 199. Method: updateRepoLabel

HTTP **POST**

Description:

Updates repository label

Parameters:

- string sessionKey
- string label - repository label
- string newLabel - new repository label

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 200. Method: updateRepoSsl

HTTP **POST**

Description:

Updates repository SSL certificates

Parameters:

- string sessionKey
- int id - repository ID
- string sslCaCert - SSL CA cert description
- string sslCliCert - SSL Client cert description
- string sslCliKey - SSL Client key description

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 201. Method: updateRepoSsl

HTTP **POST**

Description:

Updates repository SSL certificates

Parameters:

- string sessionKey
- string label - repository label
- string sslCaCert - SSL CA cert description
- string sslCliCert - SSL Client cert description
- string sslCliKey - SSL Client key description

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 202. Method: updateRepoUrl

HTTP **POST**

Description:

Updates repository source URL

Parameters:

- string sessionKey
- int id - repository ID
- string url - new repository URL

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

Chapter 203. Method: updateRepoUrl

HTTP **POST**

Description:

Updates repository source URL

Parameters:

- string sessionKey
- string label - repository label
- string url - new repository URL

Returns:

- * struct channel
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"
 - boolean "hasSignedMetadata"
 - array "sslContentSources"
- struct content source SSL
 - string "sslCaDesc"
 - string "sslCertDesc"
 - string "sslKeyDesc"

configchannel

Chapter 204. Available methods

- `channelExists`
- `create`
- `create`
- `create`
- `createOrUpdatePath`
- `createOrUpdateSymlink`
- `deleteChannels`
- `deleteFileRevisions`
- `deleteFiles`
- `deployAllSystems`
- `deployAllSystems`
- `deployAllSystems`
- `deployAllSystems`
- `getDetails`
- `getDetails`
- `getEncodedFileRevision`
- `getFileRevision`
- `getFileRevisions`
- `listAssignedSystemGroups`
- `listFiles`
- `listGlobals`
- `listSubscribedSystems`
- `lookupChannelInfo`
- `lookupFileInfo`
- `lookupFileInfo`

-
- `scheduleFileComparisons`
 - `syncSaltFilesOnDisk`
 - `update`
 - `updateInitSl`

Chapter 205. Description

Provides methods to access and modify many aspects of configuration channels.

Namespace:

configchannel

Chapter 206. Method: channelExists

HTTP **POST**

Description:

Check for the existence of the config channel provided.

Parameters:

- string sessionKey
- string label - channel to check for

Returns:

- int existence - 1 if exists, 0 otherwise

Chapter 207. Method: create

HTTP **POST**

Description:

Create a new global config channel. Caller must be at least a config admin or an organization admin.

Parameters:

- string sessionKey
- string label
- string name
- string description

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 208. Method: create

HTTP **POST**

Description:

Create a new global config channel. Caller must be at least a config admin or an organization admin.

Parameters:

- string sessionKey
- string label
- string name
- string description
- string type - the channel type either 'normal' or 'state'

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 209. Method: create

HTTP **POST**

Description:

Create a new global config channel. Caller must be at least a config admin or an organization admin.

Parameters:

- string sessionKey
- string label
- string name
- string description
- string type - the channel type either 'normal' or 'state'
- struct pathInfo - the path info
 - string "contents" - contents of the init.sls file
 - boolean "contents_enc64" - identifies base64 encoded content(default: disabled)

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"

- int "priority"

Chapter 210. Method: createOrUpdatePath

HTTP **POST**

Description:

Create a new file or directory with the given path, or update an existing path.

Parameters:

- string `sessionKey`
- string `label` - the channel label
- string `path`
- boolean `isDir` - true if the path is a directory, False if it is a file
- struct `pathInfo`
 - string `"contents"` - contents of the file (text or base64 encoded if binary or want to preserve control characters like LF, CR etc.)(only for non-directories)
 - boolean `"contents_enc64"` - identifies base64 encoded content (default: disabled, only for non-directories)
 - string `"owner"` - owner of the file/directory
 - string `"group"` - group name of the file/directory
 - string `"permissions"` - octal file/directory permissions (eg: 644)
 - string `"selinux_ctx"` - SELinux Security context (optional)
 - string `"macro-start-delimiter"` - config file macro start delimiter. Use null or empty string to accept the default. (only for non-directories)
 - string `"macro-end-delimiter"` - config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
 - int `"revision"` - next revision number, auto increment for null
 - boolean `"binary"` - mark the binary content, if True, base64 encoded content is expected (only for non-directories)

Returns:

- * struct configuration revision information

-
- string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 211. Method: createOrUpdateSymlink

HTTP **POST**

Description:

Create a new symbolic link with the given path, or update an existing path in config channel of 'normal' type.

Parameters:

- string sessionKey
- string label
- string path
- struct pathInfo
 - string "target_path" - the target path for the symbolic link
 - string "selinux_ctx" - SELinux Security context (optional)
 - int "revision" - next revision number, skip this field for automatic revision number assignment

Returns:

- * struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision

-
- `dateTime.iso8601 "creation"` - Creation Date
 - `dateTime.iso8601 "modified"` - Last Modified Date
 - `string "owner"` - File Owner. Present for files or directories only.
 - `string "group"` - File Group. Present for files or directories only.
 - `int "permissions"` - File Permissions (Deprecated). Present for files or directories only.
 - `string "permissions_mode"` - File Permissions. Present for files or directories only.
 - `string "selinux_ctx"` - SELinux Context (optional).
 - `boolean "binary"` - `true/false` , Present for files only.
 - `string "sha256"` - File's sha256 signature. Present for files only.
 - `string "macro-start-delimiter"` - Macro start delimiter for a config file. Present for text files only.
 - `string "macro-end-delimiter"` - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 212. Method: deleteChannels

HTTP **POST**

Description:

Delete a list of global config channels. Caller must be a config admin.

Parameters:

- string sessionKey
- string array labels - configuration channel labels to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 213. Method: deleteFileRevisions

HTTP **POST**

Description:

Delete specified revisions of a given configuration file

Parameters:

- string sessionKey
- string label - label of config channel to lookup on
- string filePath - configuration file path
- int array revisions - list of revisions to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 214. Method: deleteFiles

HTTP **POST**

Description:

Remove file paths from a global channel.

Parameters:

- string sessionKey
- string label - channel to remove the files from
- string array paths

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 215. Method: deployAllSystems

HTTP **POST**

Description:

Schedule an immediate configuration deployment for all systems subscribed to a particular configuration channel.

Parameters:

- string sessionKey
- string label - the configuration channel's label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 216. Method: deployAllSystems

HTTP **POST**

Description:

Schedule a configuration deployment for all systems subscribed to a particular configuration channel.

Parameters:

- string sessionKey
- string label - the configuration channel's label
- dateTime.iso8601 date - the date to schedule the action

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 217. Method: deployAllSystems

HTTP **POST**

Description:

Schedule a configuration deployment of a certain file for all systems subscribed to a particular configuration channel.

Parameters:

- string sessionKey
- string label - the configuration channel's label
- string filePath - the configuration file path

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 218. Method: deployAllSystems

HTTP **POST**

Description:

Schedule a configuration deployment of a certain file for all systems subscribed to a particular configuration channel.

Parameters:

- string sessionKey
- string label - the configuration channel's label
- string filePath - the configuration file path
- dateTime.iso8601 date - the date to schedule the action

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 219. Method: getDetails

HTTP **GET**

Description:

Lookup config channel details.

Parameters:

- string sessionKey
- string label

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 220. Method: getDetails

HTTP **GET**

Description:

Lookup config channel details.

Parameters:

- string sessionKey
- int id - the channel ID

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 221. Method: getEncodedFileRevision

HTTP **GET**

Description:

Get revision of the specified configuration file and transmit the contents as base64 encoded.

Parameters:

- string sessionKey
- string label - label of config channel to lookup on
- string filePath - config file path to examine
- int revision - config file revision to examine

Returns:

- * struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.

-
- int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Chapter 222. Method: getFileRevision

HTTP **GET**

Description:

Get revision of the specified config file

Parameters:

- string sessionKey
- string label - label of config channel to lookup on
- string filePath - config file path to examine
- int revision - config file revision to examine

Returns:

- * struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.

-
- int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Chapter 223. Method: getFileRevisions

HTTP **GET**

Description:

Get list of revisions for specified config file

Parameters:

- string `sessionKey`
- string `label` - label of config channel to lookup on
- string `filePath` - config file path to examine

Returns:

- array :
- struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the `contents_enc64` attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - `dateTime.iso8601` "creation" - Creation Date
 - `dateTime.iso8601` "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.

-
- int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Chapter 224. Method: listAssignedSystemGroups

HTTP **GET**

Description:

Return a list of Groups where a given configuration channel is assigned to

Parameters:

- string sessionKey
- string label - label of the config channel to list assigned groups

Returns:

- array :
- struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 225. Method: listFiles

HTTP **GET**

Description:

Return a list of files in a channel.

Parameters:

- string sessionKey
- string label - label of config channel to list files on

Returns:

- array :
- struct configuration file information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - dateTime.iso8601 "last_modified" - Last Modified Date

Chapter 226. Method: listGlobals

HTTP **GET**

Description:

List all the global config channels accessible to the logged-in user.

Parameters:

- string sessionKey

Returns:

- array :
- struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - string "type"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 227. Method: listSubscribedSystems

HTTP **GET**

Description:

Return a list of systems subscribed to a configuration channel

Parameters:

- string sessionKey
- string label - label of the config channel to list subscribed systems

Returns:

- array :
- struct system
 - int "id"
 - string "name"

Chapter 228. Method: lookupChannelInfo

HTTP **GET**

Description:

Lists details on a list of channels given their channel labels.

Parameters:

- string sessionKey
- string array labels - the channel labels

Returns:

- array :
- struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 229. Method: lookupFileInfo

HTTP **GET**

Description:

Given a list of paths and a channel, returns details about the latest revisions of the paths.

Parameters:

- string `sessionKey`
- string `label` - label of config channel to lookup on
- string array `paths` - list of paths to examine

Returns:

- array :
- struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the `contents_enc64` attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - `dateTime.iso8601` "creation" - Creation Date
 - `dateTime.iso8601` "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.

-
- int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 230. Method: lookupFileInfo

HTTP **GET**

Description:

Given a path, revision number, and a channel, returns details about the latest revisions of the paths.

Parameters:

- string sessionKey
- string label - label of config channel to lookup on
- string path - path of file/directory
- int revision - the revision number

Returns:

- * struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.

-
- string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.12

Chapter 231. Method: scheduleFileComparisons

HTTP **POST**

Description:

Schedule a comparison of the latest revision of a file against the version deployed on a list of systems.

Parameters:

- string sessionKey
- string label - label of config channel
- string path - file path
- long array sids

Returns:

- int actionId - the action ID of the scheduled action

Chapter 232. Method: syncSaltFilesOnDisk

HTTP **POST**

Description:

Synchronize all files on the disk to the current state of the database.

Parameters:

- string sessionKey
- string array labels - configuration channel labels to synchronize files from

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 233. Method: update

HTTP **POST**

Description:

Update a global config channel. Caller must be at least a config admin or an organization admin, or have access to a system containing this config channel.

Parameters:

- string sessionKey
- string label
- string name
- string description

Returns:

- * struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 234. Method: updateInitSls

HTTP **POST**

Description:

Update the init.sls file for the given state channel. User can only update contents, nothing else.

Parameters:

- string `sessionKey`
- string `label` - the channel label
- struct `pathInfo`
 - string `"contents"` - contents of the init.sls file
 - boolean `"contents_enc64"` - identifies base64 encoded content(default: disabled)
 - int `"revision"` - next revision number, auto increment for null

Returns:

- * struct configuration revision information
 - string `"type"`
 - file
 - directory
 - symlink
 - string `"path"` - File Path
 - string `"target_path"` - Symbolic link Target File Path. Present for Symbolic links only.
 - string `"channel"` - Channel Name
 - string `"contents"` - File contents (base64 encoded according to the `contents_enc64` attribute)
 - boolean `"contents_enc64"` - Identifies base64 encoded content
 - int `"revision"` - File Revision
 - `dateTime.iso8601` `"creation"` - Creation Date
 - `dateTime.iso8601` `"modified"` - Last Modified Date

-
- string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

contentmanagement

Chapter 235. Available methods

- `attachFilter`
- `attachSource`
- `attachSource`
- `buildProject`
- `buildProject`
- `createAppStreamFilters`
- `createEnvironment`
- `createFilter`
- `createProject`
- `detachFilter`
- `detachSource`
- `listFilterCriteria`
- `listFilters`
- `listProjectEnvironments`
- `listProjectFilters`
- `listProjectSources`
- `listProjects`
- `lookupEnvironment`
- `lookupFilter`
- `lookupProject`
- `lookupSource`
- `promoteProject`
- `removeEnvironment`
- `removeFilter`
- `removeProject`

-
- `updateEnvironment`
 - `updateFilter`
 - `updateProject`

Chapter 236. Description

Provides methods to access and modify Content Lifecycle Management related entities (Projects, Environments, Filters, Sources).

Namespace:

contentmanagement

Chapter 237. Method: attachFilter

HTTP **POST**

Description:

Attach a Filter to a Project

Parameters:

- string sessionKey
- string projectLabel - Project label
- int filterId - filter ID to attach

Returns:

- * struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 238. Method: attachSource

HTTP **POST**

Description:

Attach a Source to a Project

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string sourceType - Project Source type, e.g. 'software'
- string sourceLabel - Project Source label
- int sourcePosition - Project Source position

Returns:

- * struct content project source information
 - string "contentProjectLabel"
 - string "type"
 - string "state"
 - string "channelLabel" - (if type is SW_CHANNEL) the label of channel associated with the source

Chapter 239. Method: attachSource

HTTP **POST**

Description:

Attach a Source to a Project

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string sourceType - Project Source type, e.g. 'software'
- string sourceLabel - Project Source label

Returns:

- * struct content project source information
 - string "contentProjectLabel"
 - string "type"
 - string "state"
 - string "channelLabel" - (if type is SW_CHANNEL) the label of channel associated with the source

Chapter 240. Method: buildProject

HTTP **POST**

Description:

Build a Project

Parameters:

- string sessionKey
- string projectLabel - Project label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 241. Method: buildProject

HTTP **POST**

Description:

Build a Project

Parameters:

- string sessionKey
- string projectLabel - Project label
- string message - log message to be assigned to the build

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 242. Method: createAppStreamFilters

HTTP **POST**

Description:

Create Filters for AppStream Modular Channel and attach them to CLM Project

Parameters:

- string sessionKey
- string prefix - Filter name prefix
- string channelLabel - Modular Channel label
- string projectLabel - Project label

Returns:

- array :
- struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 243. Method: createEnvironment

HTTP **POST**

Description:

Create a Content Environment and appends it behind given Content Environment

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string predecessorLabel - Predecessor Environment label
- string envLabel - new Content Environment label
- string name - new Content Environment name
- string description - new Content Environment description

Returns:

- * struct content environment information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - int "version"
 - string "status"
 - dateTime.iso8601 "lastBuildDate" - last build/promote date
 - string "contentProjectLabel"
 - string "previousEnvironmentLabel"
 - string "nextEnvironmentLabel"

Chapter 244. Method: createFilter

HTTP POST

Description:

Create a Content Filter #paragraph_end() #paragraph() The following filters are available (you can get the list in machine-readable format using the listFilterCriteria() endpoint): #paragraph_end() #paragraph() Package filtering: #itemlist() #item("by name - field: name; matchers: contains or matches") #item("by name, epoch, version, release and architecture - field: nevra; matcher: equals") #itemlist_end() #paragraph_end() #paragraph() Errata/Patch filtering: #itemlist() #item("by advisory name - field: advisory_name; matcher: equals or matches") #item("by type - field: advisory_type (e.g. 'Security Advisory'); matcher: equals") #item("by synopsis - field: synopsis; matcher: equals, contains or matches") #item("by keyword - field: keyword; matcher: contains") #item("by date - field: issue_date; matcher: greater or greaterreq; value needs to be in ISO format e.g 2022-12-10T12:00:00Z") #item("by affected package name - field: package_name; matcher: contains_pkg_name or matches_pkg_name") #item("by affected package with version - field: package_nevra; matcher: contains_pkg_lt_evr, contains_pkg_le_evr, contains_pkg_eq_evr, contains_pkg_ge_evr or contains_pkg_gt_evr") #itemlist_end() #paragraph_end() #paragraph() Appstream module/stream filtering: #itemlist() #item("by module name, stream - field: module_stream; matcher: equals; value: modulaname:stream") #itemlist_end() Note: Only 'allow' rule is supported for appstream filters. #paragraph_end() #paragraph() Note: The 'matches' matcher works on Java regular expressions.

Parameters:

- string sessionKey
- string name - Filter name
- string rule - Filter rule ('deny' or 'allow')
- string entityType - Filter entityType ('package' or 'erratum')
- struct criteria
 - string "matcher" - The matcher type of the filter (e.g. 'contains')
 - string "field" - The entity field to match (e.g. 'name')
 - string "value" - The field value to match (e.g. 'kernel')

Returns:

-
- * struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 245. Method: createProject

HTTP **POST**

Description:

Create Content Project

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string name - Content Project name
- string description - Content Project description

Returns:

- * struct content project information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - dateTime.iso8601 "lastBuildDate"
 - int "orgId"
 - string "firstEnvironment"

Chapter 246. Method: detachFilter

HTTP **POST**

Description:

Detach a Filter from a Project

Parameters:

- string sessionKey
- string projectLabel - Project label
- int filterId - filter ID to detach

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 247. Method: detachSource

HTTP **POST**

Description:

Detach a Source from a Project

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string sourceType - Project Source type, e.g. 'software'
- string sourceLabel - Project Source label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 248. Method: listFilterCriteria

HTTP **GET**

Description:

List of available filter criteria

Parameters:

- string sessionKey

Returns:

- array :
 - struct Filter Criteria
 - string "type"
 - string "matcher"
 - string "field"

Chapter 249. Method: listFilters

HTTP **GET**

Description:

List all Content Filters visible to given user

Parameters:

- string `sessionKey`

Returns:

- array :
- struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 250. Method: listProjectEnvironments

HTTP **GET**

Description:

List Environments in a Content Project with the respect to their ordering

Parameters:

- string sessionKey
- string projectLabel - Content Project label

Returns:

- array :
- struct content environment information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - int "version"
 - string "status"
 - dateTime.iso8601 "lastBuildDate" - last build/promote date
 - string "contentProjectLabel"
 - string "previousEnvironmentLabel"
 - string "nextEnvironmentLabel"

Chapter 251. Method: listProjectFilters

HTTP **GET**

Description:

List all Filters associated with a Project

Parameters:

- string sessionKey
- string projectLabel - Project label

Returns:

- array :
- struct assigned content filter information
 - string "state"
- struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 252. Method: listProjectSources

HTTP **GET**

Description:

List Content Project Sources

Parameters:

- string sessionKey
- string projectLabel - Content Project label

Returns:

- array :
- struct content project source information
 - string "contentProjectLabel"
 - string "type"
 - string "state"
 - string "channelLabel" - (if type is SW_CHANNEL) the label of channel associated with the source

Chapter 253. Method: listProjects

HTTP **GET**

Description:

List Content Projects visible to user

Parameters:

- string sessionKey

Returns:

- array :
- struct content project information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - dateTime.iso8601 "lastBuildDate"
 - int "orgId"
 - string "firstEnvironment"

Chapter 254. Method: lookupEnvironment

HTTP **GET**

Description:

Look up Content Environment based on Content Project and Content Environment label

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string envLabel - Content Environment label

Returns:

- * struct content environment information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - int "version"
 - string "status"
 - dateTime.iso8601 "lastBuildDate" - last build/promote date
 - string "contentProjectLabel"
 - string "previousEnvironmentLabel"
 - string "nextEnvironmentLabel"

Chapter 255. Method: lookupFilter

HTTP **GET**

Description:

Lookup a Content Filter by ID

Parameters:

- string sessionKey
- int filterId - Filter ID

Returns:

- * struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 256. Method: lookupProject

HTTP **GET**

Description:

Look up Content Project with given label

Parameters:

- string sessionKey
- string projectLabel - Content Project label

Returns:

- * struct content project information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - dateTime.iso8601 "lastBuildDate"
 - int "orgId"
 - string "firstEnvironment"

Chapter 257. Method: lookupSource

HTTP **GET**

Description:

Look up Content Project Source

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string sourceType - Project Source type, e.g. 'software'
- string sourceLabel - Project Source label

Returns:

- * struct content project source information
 - string "contentProjectLabel"
 - string "type"
 - string "state"
 - string "channelLabel" - (if type is SW_CHANNEL) the label of channel associated with the source

Chapter 258. Method: promoteProject

HTTP **POST**

Description:

Promote an Environment in a Project

Parameters:

- string sessionKey
- string projectLabel - Project label
- string envLabel - Environment label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 259. Method: removeEnvironment

HTTP **POST**

Description:

Remove a Content Environment

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string envLabel - Content Environment label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 260. Method: removeFilter

HTTP **POST**

Description:

Remove a Content Filter

Parameters:

- string sessionKey
- int filterId - Filter ID

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 261. Method: removeProject

HTTP **POST**

Description:

Remove Content Project

Parameters:

- string sessionKey
- string projectLabel - Content Project label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 262. Method: updateEnvironment

HTTP **POST**

Description:

Update Content Environment with given label

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- string envLabel - Content Environment label
- struct props
 - string "name" - Content Environment name
 - string "description" - Content Environment description

Returns:

- * struct content environment information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - int "version"
 - string "status"
 - dateTime.iso8601 "lastBuildDate" - last build/promote date
 - string "contentProjectLabel"
 - string "previousEnvironmentLabel"
 - string "nextEnvironmentLabel"

Chapter 263. Method: updateFilter

HTTP **POST**

Description:

Update a Content Filter #paragraph_end() #paragraph() See also: createFilter(), listFilterCriteria()

Parameters:

- string sessionKey
- int filterId - Filter ID
- string name - New filter name
- string rule - New filter rule ('deny' or 'allow')
- struct criteria
 - string "matcher" - The matcher type of the filter (e.g. 'contains')
 - string "field" - The entity field to match (e.g. 'name')
 - string "value" - The field value to match (e.g. 'kernel')

Returns:

- * struct content filter information
 - int "id"
 - string "name"
 - int "orgId"
 - entityType "entity type (e.g. 'package')"
 - rule "rule (e.g. 'deny')"
 - struct criteria
 - string "matcher" - the matcher type of the filter (e.g. 'contains')
 - string "field" - the entity field to match (e.g. 'name')
 - string "value" - the field value to match (e.g. 'kernel')

Chapter 264. Method: updateProject

HTTP **POST**

Description:

Update Content Project with given label

Parameters:

- string sessionKey
- string projectLabel - Content Project label
- struct props
 - string "name" - Content Project name
 - string "description" - Content Project description

Returns:

- * struct content project information
 - int "id"
 - string "label"
 - string "name"
 - string "description"
 - dateTime.iso8601 "lastBuildDate"
 - int "orgId"
 - string "firstEnvironment"

distchannel

Chapter 265. Available methods

- `listDefaultMaps`
- `listMapsForOrg`
- `listMapsForOrg`
- `setMapForOrg`

Chapter 266. Description

Provides methods to access and modify distribution channel information

Namespace:

distchannel

Chapter 267. Method: listDefaultMaps

HTTP **GET**

Description:

Lists the default distribution channel maps

Parameters:

- string sessionKey

Returns:

- array :
- struct distribution channel map
 - string "os" - operating system
 - string "release" - OS Release
 - string "arch_name" - channel architecture
 - string "channel_label" - channel label
 - string "org_specific" - 'Y' organization specific, 'N' default

Chapter 268. Method: listMapsForOrg

HTTP **GET**

Description:

Lists distribution channel maps valid for the user's organization

Parameters:

- string sessionKey

Returns:

- array :
- struct distribution channel map
 - string "os" - operating system
 - string "release" - OS Release
 - string "arch_name" - channel architecture
 - string "channel_label" - channel label
 - string "org_specific" - 'Y' organization specific, 'N' default

Chapter 269. Method: listMapsForOrg

HTTP **GET**

Description:

Lists distribution channel maps valid for an organization, #product() admin rights needed.

Parameters:

- string sessionKey
- int orgId

Returns:

- array :
- struct distribution channel map
 - string "os" - operating system
 - string "release" - OS Release
 - string "arch_name" - channel architecture
 - string "channel_label" - channel label
 - string "org_specific" - 'Y' organization specific, 'N' default

Chapter 270. Method: setMapForOrg

HTTP **POST**

Description:

Sets, overrides (/removes if channelLabel empty) a distribution channel map within an organization

Parameters:

- string sessionKey
- string os
- string release
- string archName
- string channelLabel

Returns:

- int - 1 on success, exception thrown otherwise.

errata

Chapter 271. Available methods

- `addPackages`
- `applicableToChannels`
- `bugzillaFixes`
- `clone`
- `cloneAsOriginal`
- `cloneAsOriginalAsync`
- `cloneAsync`
- `create`
- `delete`
- `findByCve`
- `getDetails`
- `listAffectedSystems`
- `listCves`
- `listKeywords`
- `listPackages`
- `publish`
- `publishAsOriginal`
- `removePackages`
- `setDetails`

Chapter 272. Description

Provides methods to access and modify errata.

Namespace:

errata

Chapter 273. Method: addPackages

HTTP **POST**

Description:

Add a set of packages to an erratum with the given advisory name. This method will only allow for modification of custom errata created either through the UI or API.

Parameters:

- string sessionKey
- string advisoryName
- int array packagelds

Returns:

- int the number of packages added, exception otherwise

Chapter 274. Method: applicableToChannels

HTTP **GET**

Description:

Returns a list of channels applicable to the errata with the given advisory name. For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the list of channels applicable of both of them.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- array :
 - struct channel
 - int "channel_id"
 - string "label"
 - string "name"
 - string "parent_channel_label"

Chapter 275. Method: bugzillaFixes

HTTP **POST**

Description:

Get the Bugzilla fixes for an erratum matching the given advisoryName. The bugs will be returned in a struct where the bug id is the key. i.e. 208144="errata.bugzillaFixes Method Returns different results than docs say" For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the list of Bugzilla fixes of both of them.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- struct Bugzilla info
 - string "bugzilla_id" - actual bug number is the key into the struct
 - string "bug_summary" - summary who's key is the bug id

Chapter 276. Method: clone

HTTP **POST**

Description:

Clone a list of errata into the specified channel.

Parameters:

- string `sessionKey`
- string `channelLabel`
- string array `advisoryNames` - the advisory names of the errata to clone

Returns:

- array :
- struct `errata`
 - int `"id"` - errata ID
 - string `"date"` - the date erratum was created
 - string `"advisory_type"` - type of the advisory
 - string `"advisory_status"` - status of the advisory
 - string `"advisory_name"` - name of the advisory
 - string `"advisory_synopsis"` - summary of the erratum

Chapter 277. Method: cloneAsOriginal

HTTP **POST**

Description:

Clones a list of errata into a specified cloned channel according the original erratas.

Parameters:

- string sessionKey
- string channelLabel
- string array advisoryNames - the advisory names of the errata to clone

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 278. Method: cloneAsOriginalAsync

HTTP **POST**

Description:

Asynchronously clones a list of errata into a specified cloned channel according the original erratas

Parameters:

- string sessionKey
- string channelLabel
- string array advisoryNames - the advisory names of the errata to clone

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 279. Method: cloneAsync

HTTP **POST**

Description:

Asynchronously clone a list of errata into the specified channel.

Parameters:

- string sessionKey
- string channelLabel
- string array advisoryNames - the advisory names of the errata to clone

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 280. Method: create

HTTP **POST**

Description:

Create a custom errata

Parameters:

- string `sessionKey`
- struct `errataInfo`
 - string `"synopsis"`
 - string `"advisory_name"`
 - int `"advisory_release"`
 - string `"advisory_type"` - Type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', or 'Bug Fix Advisory')
 - string `"advisory_status"` - Status of advisory (one of the following: 'final', 'testing', 'stable' or 'retracted')
 - string `"product"`
 - string `"errataFrom"`
 - string `"topic"`
 - string `"description"`
 - string `"references"`
 - string `"notes"`
 - string `"solution"`
 - string `"severity"` - Severity of advisory (one of the following: 'Low', 'Moderate', 'Important', 'Critical' or 'Unspecified')
- array `bugs`
 - struct `bug`
 - int `"id"` - Bug Id
 - string `"summary"`

- string "url"

- string array keywords - list of keywords to associate with the errata
- int array packageIds
- string array channelLabels - list of channels the errata should be published to

Returns:

- * struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 281. Method: delete

HTTP **POST**

Description:

Delete an erratum. This method will only allow for deletion of custom errata created either through the UI or API.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 282. Method: findByCve

HTTP **GET**

Description:

Lookup the details for errata associated with the given CVE (e.g. CVE-2008-3270)

Parameters:

- string sessionKey
- string cveName

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 283. Method: getDetails

HTTP **GET**

Description:

Retrieves the details for the erratum matching the given advisory name.

Parameters:

- string `sessionKey`
- string `advisoryName`

Returns:

- struct `erratum`
 - int `"id"`
 - string `"issue_date"`
 - string `"update_date"`
 - string `"last_modified_date"` - last time the erratum was modified.
 - string `"synopsis"`
 - int `"release"`
 - string `"advisory_status"`
 - string `"vendor_advisory"`
 - string `"type"`
 - string `"product"`
 - string `"errataFrom"`
 - string `"topic"`
 - string `"description"`
 - string `"references"`
 - string `"notes"`
 - string `"solution"`
 - boolean `"reboot_suggested"` - A boolean flag signaling whether a system reboot is

advisable following the application of the errata. Typical example is upon kernel update.

- boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 284. Method: listAffectedSystems

HTTP **GET**

Description:

Return the list of systems affected by the errata with the given advisory name. For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the affected systems by both of them.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 285. Method: listCves

HTTP **GET**

Description:

Returns a list of http://cve.mitre.org/_blankCVEs applicable to the errata with the given advisory name. For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the list of CVEs of both of them.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- string array CVE name

Chapter 286. Method: listKeywords

HTTP **GET**

Description:

Get the keywords associated with an erratum matching the given advisory name. For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the keywords of both of them.

Parameters:

- string sessionKey
- string advisoryName

Returns:

- string array keyword associated with erratum.

Chapter 287. Method: listPackages

HTTP **GET**

Description:

Returns a list of the packages affected by the errata with the given advisory name. For those errata that are present in both vendor and user organizations under the same advisory name, this method retrieves the packages of both of them.

Parameters:

- string `sessionKey`
- string `advisoryName`

Returns:

- array :
 - struct `package`
 - int `"id"`
 - string `"name"`
 - string `"epoch"`
 - string `"version"`
 - string `"release"`
 - string `"arch_label"`
 - string array `"providing_channels"` - - Channel label providing this package.
 - string `"build_host"`
 - string `"description"`
 - string `"checksum"`
 - string `"checksum_type"`
 - string `"vendor"`
 - string `"summary"`
 - string `"cookie"`

-
- string "license"
 - string "path"
 - string "file"
 - string "build_date"
 - string "last_modified_date"
 - string "size"
 - string "payload_size"

Chapter 288. Method: publish

HTTP **POST**

Description:

Adds an existing errata to a set of channels.

Parameters:

- string sessionKey
- string advisoryName
- string array channelLabels - list of channel labels to add to

Returns:

- * struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 289. Method: publishAsOriginal

HTTP **POST**

Description:

Adds an existing cloned errata to a set of cloned channels according to its original erratum

Parameters:

- string sessionKey
- string advisoryName
- string array channelLabels - list of channel labels to add to

Returns:

- * struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 290. Method: removePackages

HTTP **POST**

Description:

Remove a set of packages from an erratum with the given advisory name. This method will only allow for modification of custom errata created either through the UI or API.

Parameters:

- string sessionKey
- string advisoryName
- int array packagelds

Returns:

- int the number of packages removed, exception otherwise

Chapter 291. Method: setDetails

HTTP **POST**

Description:

Set erratum details. All arguments are optional and will only be modified if included in the struct. This method will only allow for modification of custom errata created either through the UI or API.

Parameters:

- string sessionKey
- string advisoryName
- struct details
 - string "synopsis"
 - string "advisory_name"
 - int "advisory_release"
 - string "advisory_type" - Type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', or 'Bug Fix Advisory')
 - string "product"
 - dateTime.iso8601 "issue_date"
 - dateTime.iso8601 "update_date"
 - string "errataFrom"
 - string "topic"
 - string "description"
 - string "references"
 - string "notes"
 - string "solution"
 - string "severity" - Severity of advisory (one of the following: 'Low', 'Moderate', 'Important', 'Critical' or 'Unspecified')
 - array "bugs" - 'bugs' is the key into the struct
 - struct bug

-
- int "id" - Bug Id
 - string "summary"
 - string "url"
 - string array "keywords" - list of keywords to associate with the errata
 - string array "cves" - list of CVEs to associate with the errata

Returns:

- int - 1 on success, exception thrown otherwise.

formula

Chapter 292. Available methods

- `getCombinedFormulaDataByServerIds`
- `getCombinedFormulasByServerId`
- `getFormulasByGroupId`
- `getFormulasByServerId`
- `getGroupFormulaData`
- `getSystemFormulaData`
- `listFormulas`
- `setFormulasOfGroup`
- `setFormulasOfServer`
- `setGroupFormulaData`
- `setSystemFormulaData`

Chapter 293. Description

Provides methods to access and modify formulas.

Namespace:

formula

Chapter 294. Method: getCombinedFormulaDataByServerIds

HTTP **GET**

Description:

Return the list of formulas a server and all his groups have.

Parameters:

- string sessionKey
- string formulaName
- int array sids

Returns:

- array :
- struct formula data
 - int "system_id"
 - string "minion_id"
 - struct "formula_values" - saved formula values

Chapter 295. Method: getCombinedFormulasByServerId

HTTP **GET**

Description:

Return the list of formulas a server and all his groups have.

Parameters:

- string sessionKey
- int sid - the system ID

Returns:

- string array the list of formulas

Chapter 296. Method: getFormulasByGroupId

HTTP **GET**

Description:

Return the list of formulas a server group has.

Parameters:

- string sessionKey
- int systemGroupId

Returns:

- string array the list of formulas

Chapter 297. Method: getFormulasByServerId

HTTP **GET**

Description:

Return the list of formulas directly applied to a server.

Parameters:

- string sessionKey
- int sid - the system ID

Returns:

- string array the list of formulas

Chapter 298. Method: getGroupFormulaData

HTTP **GET**

Description:

Get the saved data for the specific formula against specific group

Parameters:

- string sessionKey
- int groupId
- string formulaName

Returns:

- struct the saved formula data

Chapter 299. Method: `getSystemFormulaData`

HTTP `GET`

Description:

Get the saved data for the specific formula against specific server

Parameters:

- string `sessionKey`
- int `sid` - the system ID
- string `formulaName`

Returns:

- struct the saved formula data

Chapter 300. Method: listFormulas

HTTP **GET**

Description:

Return the list of formulas currently installed.

Parameters:

- string sessionKey

Returns:

- string array the list of formulas

Chapter 301. Method: setFormulasOfGroup

HTTP **POST**

Description:

Set the formulas of a server group.

Parameters:

- string sessionKey
- int systemGroupId
- string array formulas

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 302. Method: setFormulasOfServer

HTTP **POST**

Description:

Set the formulas of a server.

Parameters:

- string sessionKey
- int sid - the system ID
- string array formulas

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 303. Method: setGroupFormulaData

HTTP **POST**

Description:

Set the formula form for the specified group.

Parameters:

- string sessionKey
- int groupId
- string formulaName
- struct content - struct containing the values for each field in the form

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 304. Method: setSystemFormulaData

HTTP **POST**

Description:

Set the formula form for the specified server.

Parameters:

- string sessionKey
- int systemId
- string formulaName
- struct content - struct content with the values for each field in the form

Returns:

- int - 1 on success, exception thrown otherwise.

image

Chapter 305. Available methods

- `addImageFile`
- `delete`
- `deleteImageFile`
- `getCustomValues`
- `getDetails`
- `getPillar`
- `getRelevantErrata`
- `importContainerImage`
- `importImage`
- `importOSImage`
- `listImages`
- `listPackages`
- `scheduleImageBuild`
- `setPillar`

Chapter 306. Description

Provides methods to access and modify images.

Namespace:

image

Chapter 307. Method: addImageFile

HTTP **POST**

Description:

Delete image file

Parameters:

- string sessionKey
- int imageId - ID of the image
- string file - the file name, it must exist in the store
- string type - the image type
- boolean external - the file is external

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 308. Method: delete

HTTP **POST**

Description:

Delete an image

Parameters:

- string sessionKey
- int imageId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 309. Method: deleteImageFile

HTTP **POST**

Description:

Delete image file

Parameters:

- string sessionKey
- int imageId - ID of the image
- string file - the file name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 310. Method: getCustomValues

HTTP **GET**

Description:

Get the custom data values defined for the image

Parameters:

- string sessionKey
- int imageId

Returns:

- struct the map of custom labels to custom values
 - string "custom info label"
 - string "value"

Chapter 311. Method: getDetails

HTTP **GET**

Description:

Get details of an image

Parameters:

- string sessionKey
- int imageId

Returns:

- * struct image overview information
 - int "id"
 - string "name" - image name
 - string "type" - image type
 - string "version" - image tag/version
 - int "revision" - image build revision number
 - string "arch" - image architecture
 - boolean "external" - true if the image is built externally, false otherwise
 - string "checksum"
 - string "profileLabel"
 - string "storeLabel"
 - string "buildStatus" - One of:
 - queued
 - picked up
 - completed
 - failed
 - string "inspectStatus" - Available if the build is successful. One of:
 - queued

-
- picked up
 - completed
 - failed
 - int "buildServerId"
 - int "securityErrata"
 - int "bugErrata"
 - int "enhancementErrata"
 - int "outdatedPackages"
 - int "installedPackages"
 - struct "files" - image files
 - boolean "obsolete" - true if the image has been replaced in the store

Chapter 312. Method: getPillar

HTTP **GET**

Description:

Get pillar data of an image. The "size" entries are converted to string.

Parameters:

- string sessionKey
- int imageId

Returns:

- struct the pillar data

Chapter 313. Method: getRelevantErrata

HTTP **GET**

Description:

Returns a list of all errata that are relevant for the image

Parameters:

- string sessionKey
- int imageId

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 314. Method: importContainerImage

HTTP **POST**

Description:

Import an image and schedule an inspect afterwards

Parameters:

- string sessionKey
- string name - image name as specified in the store
- string version - version to import or empty
- int buildHostId - system ID of the build host
- string storeLabel
- string activationKey - activation key to get the channel data from
- dateTime.iso8601 earliestOccurrence - earliest the following inspect can run

Returns:

- int the ID of the inspect action created

Chapter 315. Method: importImage (Deprecated)

HTTP **POST**

Description:

Import an image and schedule an inspect afterwards

Deprecated - Schedule a Container image import

Parameters:

- string sessionKey
- string name - image name as specified in the store
- string version - version to import or empty
- int buildHostId - system ID of the build host
- string storeLabel
- string activationKey - activation key to get the channel data from
- dateTime.iso8601 earliestOccurrence - earliest the following inspect can run

Returns:

- int the ID of the inspect action created

Chapter 316. Method: importOSImage

HTTP **POST**

Description:

Import an image and schedule an inspect afterwards

Parameters:

- string sessionKey
- string name - image name as specified in the store
- string version - version to import
- string arch - image architecture

Returns:

- int the ID of the image

Chapter 317. Method: listImages

HTTP **GET**

Description:

List available images

Parameters:

- string sessionKey

Returns:

- array :
- struct image information
 - int "id"
 - string "name" - image name
 - string "version" - image tag/version
 - int "revision" - image build revision number
 - string "arch" - image architecture
 - boolean "external" - true if the image is built externally, false otherwise
 - string "storeLabel"
 - string "checksum"
 - string "obsolete"

Chapter 318. Method: listPackages

HTTP **GET**

Description:

List the installed packages on the given image

Parameters:

- string sessionKey
- int imageId

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - string "arch"

Chapter 319. Method: scheduleImageBuild

HTTP **POST**

Description:

Schedule an image build

Parameters:

- string sessionKey
- string profileLabel
- string version - version to build or empty
- int buildHostId - system id of the build host
- dateTime.iso8601 earliestOccurrence - earliest the build can run.

Returns:

- int the ID of the build action created

Chapter 320. Method: setPillar

HTTP **POST**

Description:

Set pillar data of an image. The "size" entries should be passed as string.

Parameters:

- string sessionKey
- int imageld
- struct pillarData

Returns:

- int - 1 on success, exception thrown otherwise.

image.delta

Chapter 321. Available methods

- `createDeltaImage`
- `getDetails`
- `listDeltas`

Chapter 322. Description

Provides methods to access and modify delta images.

Namespace:

image.delta

Chapter 323. Method: createDeltaImage

HTTP **POST**

Description:

Import an image and schedule an inspect afterwards. The "size" entries in the pillar should be passed as string.

Parameters:

- string sessionKey
- int sourceImageId
- int targetImageId
- string file
- struct pillar

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 324. Method: getDetails

HTTP **GET**

Description:

Get details of an Image

Parameters:

- string sessionKey
- int sourceImageId
- int targetImageId

Returns:

- * struct delta image information
 - int "source_id"
 - int "target_id"
 - string "file" - file path
 - struct "pillar" - pillar data

Chapter 325. Method: listDeltas

HTTP **GET**

Description:

List available DeltaImages

Parameters:

- string sessionKey

Returns:

- array :
- struct delta image information
 - int "source_id"
 - int "target_id"
 - string "file" - file path
 - struct "pillar" - pillar data

image.profile

Chapter 326. Available methods

- create
- create
- delete
- deleteCustomValues
- getCustomValues
- getDetails
- listImageProfileTypes
- listImageProfiles
- setCustomValues
- setDetails

Chapter 327. Description

Provides methods to access and modify image profiles.

Namespace:

image.profile

Chapter 328. Method: create

HTTP **POST**

Description:

Create a new image profile

Parameters:

- string sessionKey
- string label
- string type
- string storeLabel
- string path
- string activationKey - optional
- string kiwiOptions

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 329. Method: create

HTTP **POST**

Description:

Create a new image profile

Parameters:

- string sessionKey
- string label
- string type
- string storeLabel
- string path
- string activationKey - optional

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 330. Method: delete

HTTP **POST**

Description:

Delete an image profile

Parameters:

- string sessionKey
- string label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 331. Method: deleteCustomValues

HTTP **POST**

Description:

Delete the custom values defined for the specified image profile. (Note: Attempt to delete values of non-existing keys throws exception. Attempt to delete value of existing key which has assigned no values doesn't throw exception.)

Parameters:

- string sessionKey
- string label
- string array keys - the custom data keys

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 332. Method: getCustomValues

HTTP **GET**

Description:

Get the custom data values defined for the image profile

Parameters:

- string sessionKey
- string label

Returns:

- struct the map of custom labels to custom values
 - string "custom info label"
 - string "value"

Chapter 333. Method: getDetails

HTTP **GET**

Description:

Get details of an image profile

Parameters:

- string sessionKey
- string label

Returns:

- * struct image profile information
 - string "label"
 - string "imageType"
 - string "imageStore"
 - string "activationKey"
 - string "path" - in case type support path

Chapter 334. Method: listImageProfileTypes

HTTP **GET**

Description:

List available image store types

Parameters:

- string sessionKey

Returns:

- string array the list of image profile types

Chapter 335. Method: listImageProfiles

HTTP **GET**

Description:

List available image profiles

Parameters:

- string sessionKey

Returns:

- array :
- struct image profile information
 - string "label"
 - string "imageType"
 - string "imageStore"
 - string "activationKey"
 - string "path" - in case type support path

Chapter 336. Method: setCustomValues

HTTP **POST**

Description:

Set custom values for the specified image profile

Parameters:

- string sessionKey
- string label
- struct values - the map of custom labels to custom values
 - string "custom info label"
 - string "value"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 337. Method: setDetails

HTTP **POST**

Description:

Set details of an image profile

Parameters:

- string sessionKey
- string label
- struct details
 - string "storeLabel"
 - string "path"
 - string "activationKey" - set empty string to unset

Returns:

- int - 1 on success, exception thrown otherwise.

image.store

Chapter 338. Available methods

- create
- delete
- getDetails
- listImageStoreTypes
- listImageStores
- setDetails

Chapter 339. Description

Provides methods to access and modify image stores.

Namespace:

image.store

Chapter 340. Method: create

HTTP **POST**

Description:

Create a new image store

Parameters:

- string sessionKey
- string label
- string uri
- string storeType
- struct credentials - optional
 - string "username"
 - string "password"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 341. Method: delete

HTTP **POST**

Description:

Delete an image store

Parameters:

- string sessionKey
- string label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 342. Method: getDetails

HTTP **GET**

Description:

Get details of an image store

Parameters:

- string sessionKey
- string label

Returns:

- * struct image store information
 - string "label"
 - string "uri"
 - string "storetype"
 - boolean "hasCredentials"
 - string "username"

Chapter 343. Method: listImageStoreTypes

HTTP **GET**

Description:

List available image store types

Parameters:

- string sessionKey

Returns:

- array :
- struct image store type information
 - int "id"
 - string "label"
 - string "name"

Chapter 344. Method: listImageStores

HTTP **GET**

Description:

List available image stores

Parameters:

- string sessionKey

Returns:

- array :
- struct image store information
 - string "label"
 - string "uri"
 - string "storetype"
 - boolean "hasCredentials"
 - string "username"

Chapter 345. Method: setDetails

HTTP **POST**

Description:

Set details of an image store

Parameters:

- string sessionKey
- string label
- struct details - image store details
 - string "uri"
 - string "username" - pass empty string to unset credentials
 - string "password"

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart

Chapter 346. Available methods

- `cloneProfile`
- `createProfile`
- `createProfile`
- `createProfileWithCustomUrl`
- `createProfileWithCustomUrl`
- `deleteProfile`
- `disableProfile`
- `findKickstartForIp`
- `importFile`
- `importFile`
- `importFile`
- `importRawFile`
- `importRawFile`
- `isProfileDisabled`
- `listAllIpRanges`
- `listAutoinstallableChannels`
- `listKickstartableChannels`
- `listKickstarts`
- `renameProfile`

Chapter 347. Description

Provides methods to create kickstart files

Namespace:

kickstart

Chapter 348. Method: cloneProfile

HTTP **POST**

Description:

Clone a Kickstart Profile

Parameters:

- string sessionKey
- string ksLabelToClone - Label of the kickstart profile to clone
- string newKsLabel - label of the cloned profile

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 349. Method: createProfile

HTTP **POST**

Description:

Create a kickstart profile.

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- string kickstartHost - Kickstart hostname (of a SUSE Manager server or proxy) used to construct the default download URL for the new kickstart profile.
- string rootPassword - Root password.
- string updateType - Should the profile update itself to use the newest tree available? Possible values are: none (default) or all (includes custom Kickstart Trees).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 350. Method: createProfile

HTTP **POST**

Description:

Create a kickstart profile.

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- string kickstartHost - Kickstart hostname (of a SUSE Manager server or proxy) used to construct the default download URL for the new kickstart profile.
- string rootPassword - Root password.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 351. Method: createProfileWithCustomUrl

HTTP **POST**

Description:

Create a kickstart profile.

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- boolean downloadUrl - Download URL, or 'default' to use the kickstart tree's default URL.
- string rootPassword - Root password.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 352. Method: createProfileWithCustomUrl

HTTP **POST**

Description:

Create a kickstart profile.

Parameters:

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.
- string `virtualizationType` - `none`, `para_host`, `qemu`, `xenfv` or `xenpv`.
- string `kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- boolean `downloadUrl` - Download URL, or 'default' to use the kickstart tree's default URL.
- string `rootPassword` - Root password.
- string `updateType` - Should the profile update itself to use the newest tree available? Possible values are: `none` (default) or `all` (includes custom Kickstart Trees).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 353. Method: deleteProfile

HTTP **POST**

Description:

Delete a kickstart profile

Parameters:

- string sessionKey
- string ksLabel - The label of the kickstart profile you want to remove

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 354. Method: disableProfile

HTTP **POST**

Description:

Enable/Disable a Kickstart Profile

Parameters:

- string sessionKey
- string profileLabel - Label for the kickstart tree you want to en/disable
- string disabled - true to disable the profile

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 355. Method: findKickstartForIp

HTTP **GET**

Description:

Find an associated kickstart for a given ip address.

Parameters:

- string sessionKey
- string ipAddress - The ip address to search for (i.e. 192.168.0.1)

Returns:

- string label - label of the kickstart. Empty string if not found

Chapter 356. Method: importFile

HTTP **POST**

Description:

Import a kickstart profile.

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- string kickstartFileContents - Contents of the kickstart file to import.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 357. Method: importFile

HTTP **POST**

Description:

Import a kickstart profile.

Parameters:

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.
- string `virtualizationType` - none, para_host, qemu, xenfv or xenpv.
- string `kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- string `kickstartHost` - Kickstart hostname (of a SUSE Manager server or proxy) used to construct the default download URL for the new kickstart profile. Using this option signifies that this default URL will be used instead of any `url/nfs/cdrom/harddrive` commands in the kickstart file itself.
- string `kickstartFileContents` - Contents of the kickstart file to import.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 358. Method: importFile

HTTP **POST**

Description:

Import a kickstart profile.

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- string kickstartHost - Kickstart hostname (of a SUSE Manager server or proxy) used to construct the default download URL for the new kickstart profile. Using this option signifies that this default URL will be used instead of any url/nfs/cdrom/harddrive commands in the kickstart file itself.
- string kickstartFileContents - Contents of the kickstart file to import.
- string updateType - Should the profile update itself to use the newest tree available? Possible values are: none (default) or all (includes custom Kickstart Trees).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 359. Method: importRawFile

HTTP **POST**

Description:

Import a raw kickstart file into `#product()`.

Parameters:

- string `sessionKey`
- string `profileLabel` - Label for the new kickstart profile.
- string `virtualizationType` - none, para_host, qemu, xenfv or xenpv.
- string `kickstartableTreeLabel` - Label of a kickstartable tree to associate the new profile with.
- string `kickstartFileContents` - Contents of the kickstart file to import.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 360. Method: importRawFile

HTTP **POST**

Description:

Import a raw kickstart file into #product().

Parameters:

- string sessionKey
- string profileLabel - Label for the new kickstart profile.
- string virtualizationType - none, para_host, qemu, xenfv or xenpv.
- string kickstartableTreeLabel - Label of a kickstartable tree to associate the new profile with.
- string kickstartFileContents - Contents of the kickstart file to import.
- string updateType - Should the profile update itself to use the newest tree available? Possible values are: none (default) or all (includes custom Kickstart Trees).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 361. Method: isProfileDisabled

HTTP **GET**

Description:

Returns whether a kickstart profile is disabled

Parameters:

- string sessionKey
- string profileLabel - kickstart profile label

Returns:

- boolean disabled - true if profile is disabled

Chapter 362. Method: listAllIpRanges

HTTP **GET**

Description:

List all Ip Ranges and their associated kickstarts available in the user's org.

Parameters:

- string sessionKey

Returns:

- array :
- struct kickstart IP range
 - string "ksLabel" - the kickstart label associated with the IP range
 - string "max" - the max IP of the range
 - string "min" - the min IP of the range

Chapter 363. Method: listAutoinstallableChannels

HTTP **GET**

Description:

List autoinstallable channels for the logged in user.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"
 - string "gpg_key_fp"

-
- dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 364. Method: listKickstartableChannels

HTTP **GET**

Description:

List kickstartable channels for the logged in user.

Parameters:

- string sessionKey

Returns:

- array :
- struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"
 - string "gpg_key_fp"

-
- dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 365. Method: listKickstarts

HTTP **GET**

Description:

Provides a list of kickstart profiles visible to the user's org

Parameters:

- string sessionKey

Returns:

- array :
- struct kickstart
 - string "label"
 - string "tree_label"
 - string "name"
 - boolean "advanced_mode"
 - boolean "org_default"
 - boolean "active"
 - string "update_type"

Chapter 366. Method: renameProfile

HTTP **POST**

Description:

Rename a kickstart profile in `#product()`.

Parameters:

- string `sessionKey`
- string `originalLabel` - Label for the kickstart profile you want to rename
- string `newLabel` - new label to change to

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.filepreservation

Chapter 367. Available methods

- create
- delete
- getDetails
- listAllFilePreservations

Chapter 368. Description

Provides methods to retrieve and manipulate kickstart file preservation lists.

Namespace:

kickstart.filepreservation

Chapter 369. Method: create

HTTP **POST**

Description:

Create a new file preservation list.

Parameters:

- string sessionKey
- string name - name of the file list to create
- string array files - file names to include

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 370. Method: delete

HTTP **POST**

Description:

Delete a file preservation list.

Parameters:

- string sessionKey
- string name - name of the file list to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 371. Method: getDetails

HTTP **GET**

Description:

Returns all the data associated with the given file preservation list.

Parameters:

- string sessionKey
- string name - name of the file list to retrieve details for

Returns:

- * struct file list
 - string "name"
 - string array "file_names" - the list of file names

Chapter 372. Method: listAllFilePreservations

HTTP **GET**

Description:

List all file preservation lists for the organization associated with the user logged into the given session

Parameters:

- string sessionKey

Returns:

- array :
- struct file preservation
 - int "id"
 - string "name"
 - dateTime.iso8601 "created"
 - dateTime.iso8601 "last_modified"

kickstart.keys

Chapter 373. Available methods

- create
- delete
- getDetails
- listAllKeys
- update

Chapter 374. Description

Provides methods to manipulate kickstart keys.

Namespace:

kickstart.keys

Chapter 375. Method: create

HTTP **POST**

Description:

creates a new key with the given parameters

Parameters:

- string sessionKey
- string description
- string type - valid values are GPG or SSL
- string content

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 376. Method: delete

HTTP **POST**

Description:

deletes the key identified by the given parameters

Parameters:

- string sessionKey
- string description

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 377. Method: getDetails

HTTP **GET**

Description:

returns all the data associated with the given key

Parameters:

- string sessionKey
- string description

Returns:

- struct key
 - string "description"
 - string "type"
 - string "content"

Chapter 378. Method: listAllKeys

HTTP **GET**

Description:

list all keys for the org associated with the user logged into the given session

Parameters:

- string sessionKey

Returns:

- array :
 - struct key
 - string "description"
 - string "type"

Chapter 379. Method: update

HTTP **POST**

Description:

Updates type and content of the key identified by the description

Parameters:

- string sessionKey
- string description
- string type - valid values are GPG or SSL
- string content

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.profile

Chapter 380. Available methods

- `addIpRange`
- `addScript`
- `addScript`
- `addScript`
- `compareActivationKeys`
- `compareAdvancedOptions`
- `comparePackages`
- `downloadKickstart`
- `downloadRenderedKickstart`
- `getAdvancedOptions`
- `getAvailableRepositories`
- `getCfgPreservation`
- `getChildChannels`
- `getCustomOptions`
- `getKickstartTree`
- `getRepositories`
- `getUpdateType`
- `getVariables`
- `getVirtualizationType`
- `listIpRanges`
- `listScripts`
- `orderScripts`
- `removeIpRange`
- `removeScript`
- `setAdvancedOptions`

-
- `setCfgPreservation`
 - `setChildChannels`
 - `setCustomOptions`
 - `setKickstartTree`
 - `setLogging`
 - `setRepositories`
 - `setUpdateType`
 - `setVariables`
 - `setVirtualizationType`

Chapter 381. Description

Provides methods to access and modify many aspects of a kickstart profile.

Namespace:

kickstart.profile

Chapter 382. Method: addIpRange

HTTP **POST**

Description:

Add an ip range to a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The label of the kickstart
- string min - The ip address making up the minimum of the range (i.e. 192.168.0.1)
- string max - The ip address making up the maximum of the range (i.e. 192.168.0.254)

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 383. Method: addScript

HTTP **POST**

Description:

Add a pre/post script to a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The kickstart label to add the script to.
- string name - The kickstart script name.
- string contents - The full script to add.
- string interpreter - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- string type - The type of script (either 'pre' or 'post').
- boolean chroot - Whether to run the script in the chrooted install location (recommended) or not.

Returns:

- int id - the id of the added script

Chapter 384. Method: addScript

HTTP **POST**

Description:

Add a pre/post script to a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The kickstart label to add the script to.
- string name - The kickstart script name.
- string contents - The full script to add.
- string interpreter - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- string type - The type of script (either 'pre' or 'post').
- boolean chroot - Whether to run the script in the chrooted install location (recommended) or not.
- boolean template - Enable templating using cobbler.

Returns:

- int id - the id of the added script

Chapter 385. Method: addScript

HTTP **POST**

Description:

Add a pre/post script to a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The kickstart label to add the script to.
- string name - The kickstart script name.
- string contents - The full script to add.
- string interpreter - The path to the interpreter to use (i.e. /bin/bash). An empty string will use the kickstart default interpreter.
- string type - The type of script (either 'pre' or 'post').
- boolean chroot - Whether to run the script in the chrooted install location (recommended) or not.
- boolean template - Enable templating using cobbler.
- boolean erroronfail - Whether to throw an error if the script fails or not

Returns:

- int id - the id of the added script

Chapter 386. Method: compareActivationKeys

HTTP **POST**

Description:

Returns a list for each kickstart profile; each list will contain activation keys not present on the other profile.

Parameters:

- string sessionKey
- string kickstartLabel1
- string kickstartLabel2

Returns:

- struct Comparison Info
 - array "kickstartLabel1" - Actual label of the first kickstart profile is the key into the struct
 - array :
- struct activation key
 - string "key"
 - string "description"
 - int "usage_limit"
 - string "base_channel_label"
 - string array "child_channel_labels" - childChannellabel
 - string array "entitlements" - entitlementLabel
 - string array "server_group_ids" - serverGroupid
 - string array "package_names" - packageName - (deprecated by packages)
 - array "packages"
 - struct package
 - string "name" - packageName
 - string "arch" - archLabel - optional

-
- boolean "universal_default"
 - boolean "disabled"
 - string "contact_method" - One of the following:
 - default
 - ssh-push
 - ssh-push-tunnel
 - array "kickstartLabel2" - Actual label of the second kickstart profile is the key into the struct
 - array :
 - struct activation key
 - string "key"
 - string "description"
 - int "usage_limit"
 - string "base_channel_label"
 - string array "child_channel_labels" - childChannelLabel
 - string array "entitlements" - entitlementLabel
 - string array "server_group_ids" - serverGroupId
 - string array "package_names" - packageName - (deprecated by packages)
 - array "packages"
 - struct package
 - string "name" - packageName
 - string "arch" - archLabel - optional
 - boolean "universal_default"
 - boolean "disabled"
 - string "contact_method" - One of the following:
 - default
 - ssh-push
 - ssh-push-tunnel

Chapter 387. Method: compareAdvancedOptions

HTTP **POST**

Description:

Returns a list for each kickstart profile; each list will contain the properties that differ between the profiles and their values for that specific profile .

Parameters:

- string sessionKey
- string kickstartLabel1
- string kickstartLabel2

Returns:

- struct Comparison Info
 - array "kickstartLabel1" - Actual label of the first kickstart profile is the key into the struct
 - array :
- struct value
 - string "name"
 - string "value"
 - boolean "enabled"
 - array "kickstartLabel2" - Actual label of the second kickstart profile is the key into the struct
 - array :
- struct value
 - string "name"
 - string "value"
 - boolean "enabled"

Chapter 388. Method: comparePackages

HTTP **POST**

Description:

Returns a list for each kickstart profile; each list will contain package names not present on the other profile.

Parameters:

- string sessionKey
- string kickstartLabel1
- string kickstartLabel2

Returns:

- struct Comparison Info
 - array "kickstartLabel1" - Actual label of the first kickstart profile is the key into the struct
 - string array package name
 - array "kickstartLabel2" - Actual label of the second kickstart profile is the key into the struct
 - string array package name

Chapter 389. Method: downloadKickstart

HTTP **POST**

Description:

Download the full contents of a kickstart file.

Parameters:

- string sessionKey
- string ksLabel - The label of the kickstart to download.
- string host - The host to use when referring to the SUSE Manager server. Usually this should be the FQDN, but could be the ip address or shortname as well.

Returns:

- string ks - The contents of the kickstart file. Note: if an activation key is not associated with the kickstart file, registration will not occur in the generated %post section. If one is associated, it will be used for registration

Chapter 390. Method: downloadRenderedKickstart

HTTP **POST**

Description:

Downloads the Cobbler-rendered Kickstart file.

Parameters:

- string `sessionKey`
- string `ksLabel` - The label of the kickstart to download.

Returns:

- string `ks` - The contents of the kickstart file

Chapter 391. Method: getAdvancedOptions

HTTP **GET**

Description:

Get advanced options for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.

Returns:

- array :
- struct option
 - string "name"
 - string "arguments"

Chapter 392. Method: getAvailableRepositories

HTTP **GET**

Description:

Lists available OS repositories to associate with the provided kickstart profile.

Parameters:

- string sessionKey
- string ksLabel

Returns:

- string array repositoryLabel

Chapter 393. Method: getCfgPreservation

HTTP **GET**

Description:

Get ks.cfg preservation option for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.

Returns:

- boolean preserve - The value of the option. True means that ks.cfg will be copied to /root, false means that it will not

Chapter 394. Method: getChildChannels

HTTP **GET**

Description:

Get the child channels for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile.

Returns:

- string array channelLabel

Chapter 395. Method: getCustomOptions

HTTP **GET**

Description:

Get custom options for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel

Returns:

- array :
- struct option
 - int "id"
 - string "arguments"

Chapter 396. Method: getKickstartTree

HTTP **GET**

Description:

Get the kickstart tree for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.

Returns:

- string kstreeLabel - Label of the kickstart tree.

Chapter 397. Method: getRepositories

HTTP **GET**

Description:

Lists all OS repositories associated with provided kickstart profile.

Parameters:

- string sessionKey
- string ksLabel

Returns:

- string array repositoryLabel

Chapter 398. Method: getUpdateType

HTTP **GET**

Description:

Get the update type for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile.

Returns:

- string update_type - Update type for this Kickstart Profile.

Chapter 399. Method: getVariables

HTTP **GET**

Description:

Returns a list of variables associated with the specified kickstart profile

Parameters:

- string sessionKey
- string ksLabel

Returns:

- struct kickstart variable
 - string "key"
 - string or int "value"

Chapter 400. Method: getVirtualizationType

HTTP **GET**

Description:

For given kickstart profile label returns label of virtualization type it's using

Parameters:

- string sessionKey
- string ksLabel

Returns:

- string virtLabel - Label of virtualization type.

Chapter 401. Method: listIpRanges

HTTP **GET**

Description:

List all ip ranges for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The label of the kickstart

Returns:

- array :
- struct kickstart IP range
 - string "ksLabel" - the kickstart label associated with the IP range
 - string "max" - the max IP of the range
 - string "min" - the min IP of the range

Chapter 402. Method: listScripts

HTTP **GET**

Description:

List the pre and post scripts for a kickstart profile in the order they will run during the kickstart.

Parameters:

- string sessionKey
- string ksLabel - The label of the kickstart

Returns:

- array :
- struct kickstart script
 - int "id"
 - string "name"
 - string "contents"
 - string "script_type" - the type of script ('pre' or 'post')
 - string "interpreter" - the scripting language interpreter to use for this script. An empty string indicates the default kickstart shell.
 - boolean "chroot" - true if the script will be executed within the chroot environment
 - boolean "erroronfail" - true if the script will throw an error if it fails
 - boolean "template" - true if templating using cobbler is enabled
 - boolean "beforeRegistration" - true if script will run before the server registers and performs server actions

Chapter 403. Method: orderScripts

HTTP **POST**

Description:

Change the order that kickstart scripts will run for this kickstart profile. Scripts will run in the order they appear in the array. There are three arrays, one for all pre scripts, one for the post scripts that run before registration and server actions happen, and one for post scripts that run after registration and server actions. All scripts must be included in one of these lists, as appropriate.

Parameters:

- string `sessionKey`
- string `ksLabel` - The label of the kickstart
- int array `preScripts` - IDs of the ordered pre scripts
- int array `postScriptsBeforeRegistration` - IDs of the ordered post scripts that will run before registration
- int array `postScriptsAfterRegistration` - IDs of the ordered post scripts that will run after registration

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 404. Method: removeIpRange

HTTP **POST**

Description:

Remove an ip range from a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The kickstart label of the ip range you want to remove
- string ipAddress - An Ip Address that falls within the range that you are wanting to remove. The min or max of the range will work.

Returns:

- int status - 1 on successful removal, 0 if range wasn't found for the specified kickstart, exception otherwise

Chapter 405. Method: removeScript

HTTP **POST**

Description:

Remove a script from a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - The kickstart from which to remove the script from.
- int scriptId - The id of the script to remove.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 406. Method: setAdvancedOptions

HTTP **POST**

Description:

Set advanced options for a kickstart profile. 'md5_crypt_rootpw' is not supported anymore. If 'sha256_crypt_rootpw' is set to 'True', 'root_pw' is taken as plaintext and will sha256 encrypted on server side, otherwise a hash encoded password (according to the auth option) is expected

Parameters:

- string sessionKey
- string ksLabel
- array options
 - struct advanced options
 - string "name" - Name of the advanced option. Valid Option names: autostep, interactive, install, upgrade, text, network, cdrom, harddrive, nfs, url, lang, langsupport, keyboard, mouse, device, deviceprobe, zerombr, clearpart, bootloader, timezone, auth, rootpw, selinux, reboot, firewall, xconfig, skipx, key, ignoredisk, autopart, cmdline, firstboot, graphical, iscsi, iscsiname, logging, monitor, multipath, poweroff, halt, services, shutdown, user, vnc, zfc, driverdisk, sha256_crypt_rootpw
 - string "arguments" - Arguments of the option

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 407. Method: setCfgPreservation

HTTP **POST**

Description:

Set ks.cfg preservation option for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.
- boolean preserve - whether or not ks.cfg and all %include fragments will be copied to /root.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 408. Method: setChildChannels

HTTP **POST**

Description:

Set the child channels for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.
- string array channelLabels - List of labels of child channels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 409. Method: setCustomOptions

HTTP **POST**

Description:

Set custom options for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel
- string array options

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 410. Method: setKickstartTree

HTTP **POST**

Description:

Set the kickstart tree for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.
- string kstreeLabel - Label of new kickstart tree.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 411. Method: setLogging

HTTP **POST**

Description:

Set logging options for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.
- boolean pre - whether or not to log the pre section of a kickstart to `/root/ks-pre.log`
- boolean post - whether or not to log the post section of a kickstart to `/root/ks-post.log`

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 412. Method: setRepositories

HTTP **POST**

Description:

Associates OS repository to a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel
- string array repoLabels

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 413. Method: setUpdateType

HTTP **POST**

Description:

Set the update type for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - Label of kickstart profile to be changed.
- string updateType - The new update type to set. Possible values are 'all' and 'none'.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 414. Method: setVariables

HTTP **POST**

Description:

Associates list of kickstart variables with the specified kickstart profile

Parameters:

- string sessionKey
- string ksLabel
- struct variables
 - string "key"
 - string or int "value"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 415. Method: setVirtualizationType

HTTP **POST**

Description:

For given kickstart profile label sets its virtualization type.

Parameters:

- string sessionKey
- string ksLabel
- string typeLabel - One of the following: 'none', 'qemu', 'para_host', 'xenpv', 'xenfv'

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.profile.keys

Chapter 416. Available methods

- `addActivationKey`
- `getActivationKeys`
- `removeActivationKey`

Chapter 417. Description

Provides methods to access and modify the list of activation keys associated with a kickstart profile.

Namespace:

kickstart.profile.keys

Chapter 418. Method: addActivationKey

HTTP **POST**

Description:

Add an activation key association to the kickstart profile

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string key - the activation key

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 419. Method: getActivationKeys

HTTP **GET**

Description:

Lookup the activation keys associated with the kickstart profile.

Parameters:

- string `sessionKey`
- string `ksLabel` - the kickstart profile label

Returns:

- array :
- struct activation key
 - string "key"
 - string "description"
 - int "usage_limit"
 - string "base_channel_label"
 - string array "child_channel_labels" - childChannelLabel
 - string array "entitlements" - entitlementLabel
 - string array "server_group_ids" - serverGroupId
 - string array "package_names" - packageName - (deprecated by packages)
 - array "packages"
 - struct package
 - string "name" - packageName
 - string "arch" - archLabel - optional
 - boolean "universal_default"
 - boolean "disabled"
 - string "contact_method" - One of the following:
 - default

-
- ssh-push
 - ssh-push-tunnel

Chapter 420. Method: removeActivationKey

HTTP **POST**

Description:

Remove an activation key association from the kickstart profile

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string key - the activation key

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.profile.software

Chapter 421. Available methods

- `appendToSoftwareList`
- `getSoftwareDetails`
- `getSoftwareList`
- `setSoftwareDetails`
- `setSoftwareList`
- `setSoftwareList`

Chapter 422. Description

Provides methods to access and modify the software list associated with a kickstart profile.

Namespace:

kickstart.profile.software

Chapter 423. Method: appendToSoftwareList

HTTP **POST**

Description:

Append the list of software packages to a kickstart profile. Duplicate packages will be ignored.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile
- string array packageList - the list of package names to be added to the profile

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 424. Method: getSoftwareDetails

HTTP **GET**

Description:

Gets kickstart profile software details.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile

Returns:

- struct kickstart packages info
 - string "noBase" - install @Base package group
 - string "ignoreMissing" - ignore missing packages

Chapter 425. Method: getSoftwareList

HTTP **GET**

Description:

Get a list of a kickstart profile's software packages.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile

Returns:

- string array the list of the kickstart profile's software packages

Chapter 426. Method: setSoftwareDetails

HTTP **POST**

Description:

Sets kickstart profile software details.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile
- struct params - kickstart packages info
 - string "noBase" - install @Base package group
 - string "ignoreMissing" - ignore missing packages

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 427. Method: setSoftwareList

HTTP **POST**

Description:

Set the list of software packages for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile
- string array packageList - the list of package names to be set on the profile

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 428. Method: setSoftwareList

HTTP **POST**

Description:

Set the list of software packages for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile
- string array packageList - a list of package names to be set on the profile
- boolean ignoreMissing - ignore missing packages if true
- boolean noBase - don't install @Base package group if true

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.profile.system

Chapter 429. Available methods

- `addFilePreservations`
- `addKeys`
- `checkConfigManagement`
- `checkRemoteCommands`
- `disableConfigManagement`
- `disableRemoteCommands`
- `enableConfigManagement`
- `enableRemoteCommands`
- `getLocale`
- `getPartitioningScheme`
- `getRegistrationType`
- `getSELinux`
- `listFilePreservations`
- `listKeys`
- `removeFilePreservations`
- `removeKeys`
- `setLocale`
- `setPartitioningScheme`
- `setRegistrationType`
- `setSELinux`

Chapter 430. Description

Provides methods to set various properties of a kickstart profile.

Namespace:

kickstart.profile.system

Chapter 431. Method: addFilePreservations

HTTP **POST**

Description:

Adds the given list of file preservations to the specified kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string array filePreservations - the list identifying the file preservations to add

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 432. Method: addKeys

HTTP **POST**

Description:

Adds the given list of keys to the specified kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string array descriptions - the list identifying the keys to add

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 433. Method: checkConfigManagement

HTTP **POST**

Description:

Check the configuration management status for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- boolean true if configuration management is enabled; otherwise, false

Chapter 434. Method: checkRemoteCommands

HTTP **POST**

Description:

Check the remote commands status flag for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- boolean true if remote commands support is enabled; otherwise, false

Chapter 435. Method: disableConfigManagement

HTTP **POST**

Description:

Disables the configuration management flag in a kickstart profile so that a system created using this profile will be NOT be configuration capable.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 436. Method: disableRemoteCommands

HTTP **POST**

Description:

Disables the remote command flag in a kickstart profile so that a system created using this profile will be capable of running remote commands

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 437. Method: enableConfigManagement

HTTP **POST**

Description:

Enables the configuration management flag in a kickstart profile so that a system created using this profile will be configuration capable.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 438. Method: enableRemoteCommands

HTTP **POST**

Description:

Enables the remote command flag in a kickstart profile so that a system created using this profile will be capable of running remote commands

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 439. Method: getLocale

HTTP **GET**

Description:

Retrieves the locale for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- struct locale info
 - string "locale"
 - boolean "useUtc"
 - true - the hardware clock uses UTC
 - false - the hardware clock does not use UTC

Chapter 440. Method: getPartitioningScheme

HTTP **GET**

Description:

Get the partitioning scheme for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the label of a kickstart profile

Returns:

- string array a list of partitioning commands used to setup the partitions, logical volumes and volume groups

Chapter 441. Method: getRegistrationType

HTTP **POST**

Description:

returns the registration type of a given kickstart profile. Registration Type can be one of reactivation/deletion/none These types determine the behaviour of the registration when using this profile for reprovisioning.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- string the registration type
 - reactivation
 - deletion
 - none

Chapter 442. Method: getSELinux

HTTP **GET**

Description:

Retrieves the SELinux enforcing mode property of a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- string enforcing mode
 - enforcing
 - permissive
 - disabled

Chapter 443. Method: listFilePreservations

HTTP **GET**

Description:

Returns the set of all file preservations associated with the given kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- array :
- struct file list
 - string "name"
 - string array "file_names" - the list of file names

Chapter 444. Method: listKeys

HTTP **GET**

Description:

Returns the set of all keys associated with the given kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label

Returns:

- array :
 - struct key
 - string "description"
 - string "type"
 - string "content"

Chapter 445. Method: removeFilePreservations

HTTP **POST**

Description:

Removes the given list of file preservations from the specified kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string array filePreservations - the list identifying the file preservations to add

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 446. Method: removeKeys

HTTP **POST**

Description:

Removes the given list of keys from the specified kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string array descriptions - the list identifying the keys to remove

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 447. Method: setLocale

HTTP **POST**

Description:

Sets the locale for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string locale - the locale
- boolean useUtc
 - true - the hardware clock uses UTC
 - false - the hardware clock does not use UTC

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 448. Method: setPartitioningScheme

HTTP **POST**

Description:

Set the partitioning scheme for a kickstart profile.

Parameters:

- string sessionKey
- string ksLabel - the label of the kickstart profile to update
- string array scheme - the partitioning scheme is a list of partitioning command strings used to setup the partitions, volume groups and logical volumes.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 449. Method: setRegistrationType

HTTP **POST**

Description:

Sets the registration type of a given kickstart profile. Registration Type can be one of reactivation/deletion/none These types determine the behaviour of the re registration when using this profile.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string registrationType
 - reactivation - to try and generate a reactivation key and use that to register the system when reprovisioning a system.
 - deletion - to try and delete the existing system profile and reregister the system being reprovisioned as new
 - none - to preserve the status quo and leave the current system as a duplicate on a reprovision.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 450. Method: setSELinux

HTTP **POST**

Description:

Sets the SELinux enforcing mode property of a kickstart profile so that a system created using this profile will have the appropriate SELinux enforcing mode.

Parameters:

- string sessionKey
- string ksLabel - the kickstart profile label
- string enforcingMode - the SELinux enforcing mode
 - enforcing
 - permissive
 - disabled

Returns:

- int - 1 on success, exception thrown otherwise.

kickstart.snippet

Chapter 451. Available methods

- createOrUpdate
- delete
- listAll
- listCustom
- listDefault

Chapter 452. Description

Provides methods to create kickstart files

Namespace:

kickstart.snippet

Chapter 453. Method: createOrUpdate

HTTP **POST**

Description:

Will create a snippet with the given name and contents if it doesn't exist. If it does exist, the existing snippet will be updated.

Parameters:

- string sessionKey
- string name
- string contents

Returns:

- * struct snippet
 - string "name"
 - string "contents"
 - string "fragment" - the string to include in a kickstart file that will generate this snippet
 - string "file" - the local path to the file containing this snippet

Chapter 454. Method: delete

HTTP **POST**

Description:

Delete the specified snippet. If the snippet is not found, 0 is returned.

Parameters:

- string sessionKey
- string name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 455. Method: listAll

HTTP **GET**

Description:

List all cobbler snippets for the logged in user

Parameters:

- string sessionKey

Returns:

- array :
- struct snippet
 - string "name"
 - string "contents"
 - string "fragment" - the string to include in a kickstart file that will generate this snippet
 - string "file" - the local path to the file containing this snippet

Chapter 456. Method: listCustom

HTTP **GET**

Description:

List only custom snippets for the logged in user. These snippets are editable.

Parameters:

- string sessionKey

Returns:

- array :
- struct snippet
 - string "name"
 - string "contents"
 - string "fragment" - the string to include in a kickstart file that will generate this snippet
 - string "file" - the local path to the file containing this snippet

Chapter 457. Method: listDefault

HTTP **GET**

Description:

List only pre-made default snippets for the logged in user. These snippets are not editable.

Parameters:

- string sessionKey

Returns:

- array :
- struct snippet
 - string "name"
 - string "contents"
 - string "fragment" - the string to include in a kickstart file that will generate this snippet
 - string "file" - the local path to the file containing this snippet

kickstart.tree

Chapter 458. Available methods

- create
- create
- delete
- deleteTreeAndProfiles
- getDetails
- list
- listInstallTypes
- rename
- update
- update

Chapter 459. Description

Provides methods to access and modify the kickstart trees.

Namespace:

kickstart.tree

Chapter 460. Method: create

HTTP **POST**

Description:

Create a Kickstart Tree (Distribution) in #product().

Parameters:

- string sessionKey
- string treeLabel - The new kickstart tree label.
- string basePath - Path to the base or root of the kickstart tree.
- string channelLabel - Label of channel to associate with the kickstart tree.
- string installType - Label for KickstartInstallType (rhel_6, rhel_7, rhel_8, rhel_9, fedora_9).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 461. Method: create

HTTP **POST**

Description:

Create a Kickstart Tree (Distribution) in `#product()`.

Parameters:

- string `sessionKey`
- string `treeLabel` - The new kickstart tree label.
- string `basePath` - Path to the base or root of the kickstart tree.
- string `channelLabel` - Label of channel to associate with the kickstart tree.
- string `installType` - Label for `KickstartInstallType` (`rhel_2.1`, `rhel_3`, `rhel_4`, `rhel_5`, `fedora_9`).
- string `kernelOptions` - Options to be passed to the kernel when booting for the installation.
- string `postKernelOptions` - Options to be passed to the kernel when booting for the installation.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 462. Method: delete

HTTP **POST**

Description:

Delete a Kickstart Tree (Distribution) from #product().

Parameters:

- string sessionKey
- string treeLabel - Label for the kickstart tree to delete.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 463. Method: deleteTreeAndProfiles

HTTP **POST**

Description:

Delete a kickstarttree and any profiles associated with this kickstart tree. **WARNING:** This will delete all profiles associated with this kickstart tree!

Parameters:

- string sessionKey
- string treeLabel - Label for the kickstart tree to delete.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 464. Method: getDetails

HTTP **GET**

Description:

The detailed information about a kickstartable tree given the tree name.

Parameters:

- string sessionKey
- string treeLabel - Label of kickstartable tree to search.

Returns:

- * struct kickstartable tree
 - int "id"
 - string "label"
 - string "abs_path"
 - int "channel_id"
 - string "kernel_options"
 - string "post_kernel_options"
- struct kickstart install type
 - int "id"
 - string "label"
 - string "name"

Chapter 465. Method: list

HTTP **POST**

Description:

List the available kickstartable trees for the given channel.

Parameters:

- string sessionKey
- string channelLabel - Label of channel to search.

Returns:

- array :
- struct kickstartable tree
 - int "id"
 - string "label"
 - string "base_path"
 - int "channel_id"

Chapter 466. Method: listInstallTypes

HTTP **GET**

Description:

List the available kickstartable install types (rhel2,3,4,5 and fedora9+).

Parameters:

- string sessionKey

Returns:

- array :
- struct kickstart install type
 - int "id"
 - string "label"
 - string "name"

Chapter 467. Method: rename

HTTP **POST**

Description:

Rename a Kickstart Tree (Distribution) in #product().

Parameters:

- string sessionKey
- string originalLabel - Label for the kickstart tree to rename.
- string newLabel - The kickstart tree's new label.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 468. Method: update

HTTP **POST**

Description:

Edit a Kickstart Tree (Distribution) in #product().

Parameters:

- string sessionKey
- string treeLabel - Label for the kickstart tree.
- string basePath - Path to the base or root of the kickstart tree.
- string channelLabel - Label of channel to associate with kickstart tree.
- string installType - Label for KickstartInstallType (rhel_6, rhel_7, rhel_8, rhel_9, fedora_9).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 469. Method: update

HTTP **POST**

Description:

Edit a Kickstart Tree (Distribution) in #product().

Parameters:

- string sessionKey
- string treeLabel - Label for the kickstart tree.
- string basePath - Path to the base or root of the kickstart tree.
- string channelLabel - Label of channel to associate with kickstart tree.
- string installType - Label for KickstartInstallType (rhel_2.1, rhel_3, rhel_4, rhel_5, fedora_9).
- string kernelOptions - Options to be passed to the kernel when booting for the installation.
- string postKernelOptions - Options to be passed to the kernel when booting for the installation.

Returns:

- int - 1 on success, exception thrown otherwise.

maintenance

Chapter 470. Available methods

- `assignScheduleToSystems`
- `createCalendar`
- `createCalendarWithURL`
- `createSchedule`
- `createSchedule`
- `deleteCalendar`
- `deleteSchedule`
- `getCalendarDetails`
- `getScheduleDetails`
- `listCalendarLabels`
- `listScheduleNames`
- `listSystemsWithSchedule`
- `refreshCalendar`
- `retractScheduleFromSystems`
- `updateCalendar`
- `updateSchedule`

Chapter 471. Description

Provides methods to access and modify Maintenance Schedules related entities

Namespace:

maintenance

Chapter 472. Method: assignScheduleToSystems

HTTP **POST**

Description:

Assign schedule with given name to systems with given IDs. Throws a `PermissionCheckFailureException` when some of the systems are not accessible by the user. Throws a `InvalidParameterException` when some of the systems have pending actions that are not allowed in the maintenance mode.

Parameters:

- string `sessionKey`
- string `scheduleName` - The schedule name
- int array `sids` - system IDs
- string array `rescheduleStrategy` - available:
 - `Cancel` - cancel actions which are outside the maintenance windows
 - `Fail` - let assignment fail. No operation will be performed

Returns:

- int array number of involved systems

Chapter 473. Method: createCalendar

HTTP **POST**

Description:

Create a new maintenance calendar

Parameters:

- string sessionKey
- string label - maintenance calendar label
- string ical - ICal calendar data

Returns:

- array :
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 474. Method: createCalendarWithUrl

HTTP **POST**

Description:

Create a new maintenance calendar

Parameters:

- string sessionKey
- string label - maintenance calendar label
- string url - download URL for ICal calendar data

Returns:

- array :
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 475. Method: createSchedule

HTTP **POST**

Description:

Create a new maintenance Schedule

Parameters:

- string sessionKey
- string name - maintenance schedule name
- string type - schedule type: single, multi

Returns:

- array :
- struct maintenance schedule information
 - int "id"
 - int "orgId"
 - string "name"
 - string "type"
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 476. Method: createSchedule

HTTP **POST**

Description:

Create a new Maintenance Schedule

Parameters:

- string sessionKey
- string name - maintenance schedule name
- string type - schedule type: single, multi
- string calendar - maintenance calendar label

Returns:

- array :
- struct maintenance schedule information
 - int "id"
 - int "orgId"
 - string "name"
 - string "type"
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 477. Method: deleteCalendar

HTTP **POST**

Description:

Remove a maintenance calendar

Parameters:

- string sessionKey
- string label - maintenance calendar label
- boolean cancelScheduledActions - cancel actions of affected schedules

Returns:

- array :
- struct reschedule information
 - string "strategy" - selected strategy
 - string "for_schedule_name"
 - boolean "status"
 - string "message"
 - array "actions"
 - struct action information
 - int "id" - action ID
 - string "name" - action name
 - string "type" - action type
 - string "scheduler" - the user that scheduled the action (optional)
 - dateTime.iso8601 "earliest" - the earliest date and time the action will be performed
 - int "prerequisite" - ID of the prerequisite action (optional)
 - int array "affected_system_ids" - affected system IDs
 - string "details" - action details string

Chapter 478. Method: deleteSchedule

HTTP **POST**

Description:

Remove a maintenance schedule

Parameters:

- string sessionKey
- string name - maintenance schedule name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 479. Method: getCalendarDetails

HTTP **POST**

Description:

Lookup a specific maintenance schedule

Parameters:

- string sessionKey
- string label - maintenance calendar label

Returns:

- array :
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 480. Method: getScheduleDetails

HTTP **POST**

Description:

Lookup a specific maintenance schedule

Parameters:

- string sessionKey
- string name - maintenance Schedule Name

Returns:

- array :
- struct maintenance schedule information
 - int "id"
 - int "orgId"
 - string "name"
 - string "type"
- struct maintenance calendar information
 - int "id"
 - int "orgId"
 - string "label"
 - string "url" - calendar url if present
 - string "ical"

Chapter 481. Method: listCalendarLabels

HTTP **POST**

Description:

List schedule names visible to user

Parameters:

- string sessionKey

Returns:

- string array maintenance calendar labels

Chapter 482. Method: listScheduleNames

HTTP **POST**

Description:

List Schedule Names visible to user

Parameters:

- string sessionKey

Returns:

- string array maintenance schedule names

Chapter 483. Method: listSystemsWithSchedule

HTTP **POST**

Description:

List IDs of systems that have given schedule assigned Throws a `PermissionCheckFailureException` when some of the systems are not accessible by the user.

Parameters:

- string `sessionKey`
- string `scheduleName` - the schedule name

Returns:

- int array system IDs

Chapter 484. Method: refreshCalendar

HTTP **POST**

Description:

Refresh maintenance calendar data using the configured URL

Parameters:

- string `sessionKey`
- string `label` - maintenance calendar label
- string array `rescheduleStrategy` - available:
 - Cancel - cancel actions which are outside the maintenance windows
 - Fail - let update fail. The calendar stay untouched

Returns:

- array :
- struct reschedule information
 - string `"strategy"` - selected strategy
 - string `"for_schedule_name"`
 - boolean `"status"`
 - string `"message"`
 - array `"actions"`
 - struct action information
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)
 - `dateTime.iso8601` `"earliest"` - the earliest date and time the action will be performed
 - int `"prerequisite"` - ID of the prerequisite action (optional)

-
- int array "affected_system_ids" - affected system IDs
 - string "details" - action details string

Chapter 485. Method: retractScheduleFromSystems

HTTP **POST**

Description:

Retract schedule with given name from systems with given IDs. Throws a `PermissionCheckFailureException` when some of the systems are not accessible by the user.

Parameters:

- string `sessionKey`
- int array `sids` - system IDs

Returns:

- int array number of involved systems

Chapter 486. Method: updateCalendar

HTTP **POST**

Description:

Update a maintenance calendar

Parameters:

- string `sessionKey`
- string `label` - maintenance calendar label
- struct `details` - maintenance calendar details
 - string `"ical"` - new ical calendar data
 - string `"url"` - new calendar URL
- string array `rescheduleStrategy` - available:
 - `Cancel` - cancel actions which are outside the maintenance windows
 - `Fail` - let update fail. The calendar stay untouched

Returns:

- array :
- struct `reschedule information`
 - string `"strategy"` - selected strategy
 - string `"for_schedule_name"`
 - boolean `"status"`
 - string `"message"`
 - array `"actions"`
 - struct `action information`
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)

-
- `dateTime.iso8601 "earliest"` - the earliest date and time the action will be performed
 - `int "prerequisite"` - ID of the prerequisite action (optional)
 - `int array "affected_system_ids"` - affected system IDs
 - `string "details"` - action details string

Chapter 487. Method: updateSchedule

HTTP **POST**

Description:

Update a maintenance schedule

Parameters:

- string `sessionKey`
- string `name` - maintenance schedule name
- struct `details` - maintenance schedule details
 - string `"type"` - new schedule type
 - `single`
 - `multi`
 - string `"calendar"` - new calendar label
- string array `rescheduleStrategy` - available:
 - `Cancel` - cancel actions which are outside the maintenance windows
 - `Fail` - let update fail. The calendar stays untouched

Returns:

- * struct `reschedule` information
 - string `"strategy"` - selected strategy
 - string `"for_schedule_name"`
 - boolean `"status"`
 - string `"message"`
 - array `"actions"`
 - struct `action` information
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type

-
- string "scheduler" - the user that scheduled the action (optional)
 - dateTime.iso8601 "earliest" - the earliest date and time the action will be performed
 - int "prerequisite" - ID of the prerequisite action (optional)
 - int array "affected_system_ids" - affected system IDs
 - string "details" - action details string

org

Chapter 488. Available methods

- create
- createFirst
- delete
- getCImSyncPatchesConfig
- getDetails
- getDetails
- getPolicyForScapFileUpload
- getPolicyForScapResultDeletion
- isContentStagingEnabled
- isErrataEmailNotifsForOrg
- isOrgConfigManagedByOrgAdmin
- listOrgs
- listUsers
- migrateSystems
- setCImSyncPatchesConfig
- setContentStaging
- setErrataEmailNotifsForOrg
- setOrgConfigManagedByOrgAdmin
- setPolicyForScapFileUpload
- setPolicyForScapResultDeletion
- transferSystems
- updateName

Chapter 489. Description

Contains methods to access common organization management functions available from the web interface.

Namespace:

org

Chapter 490. Method: create

HTTP **POST**

Description:

Create a new organization and associated administrator account.

Parameters:

- string `sessionKey`
- string `orgName` - Organization name. Must meet same criteria as in the web UI.
- string `adminLogin` - New administrator login name.
- string `adminPassword` - New administrator password.
- string `prefix` - New administrator's prefix. Must match one of the values available in the web UI. (i.e. Dr., Mr., Mrs., Sr., etc.)
- string `firstName` - New administrator's first name.
- string `lastName` - New administrator's first name.
- string `email` - New administrator's e-mail.
- boolean `usePamAuth` - True if PAM authentication should be used for the new administrator account.

Returns:

- * struct `organization info`
 - int `"id"`
 - string `"name"`
 - int `"active_users"` - number of active users in the organization
 - int `"systems"` - number of systems in the organization
 - int `"trusts"` - number of trusted organizations
 - int `"system_groups"` - number of system groups in the organization (optional)
 - int `"activation_keys"` - number of activation keys in the organization (optional)
 - int `"kickstart_profiles"` - number of kickstart profiles in the organization (optional)

-
- int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

Chapter 491. Method: createFirst

HTTP **POST**

Description:

Create first organization and user after initial setup without authentication

Parameters:

- string orgName - Organization name. Must meet same criteria as in the web UI.
- string adminLogin - New administrator login name.
- string adminPassword - New administrator password.
- string firstName - New administrator's first name.
- string lastName - New administrator's first name.
- string email - New administrator's e-mail.

Returns:

- * struct organization info
 - int "id"
 - string "name"
 - int "active_users" - number of active users in the organization
 - int "systems" - number of systems in the organization
 - int "trusts" - number of trusted organizations
 - int "system_groups" - number of system groups in the organization (optional)
 - int "activation_keys" - number of activation keys in the organization (optional)
 - int "kickstart_profiles" - number of kickstart profiles in the organization (optional)
 - int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

Chapter 492. Method: delete

HTTP **POST**

Description:

Delete an organization. The default organization (i.e. orgId=1) cannot be deleted.

Parameters:

- string sessionKey
- int orgId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 493. Method: getClmSyncPatchesConfig

HTTP **GET**

Description:

Reads the content lifecycle management patch synchronization config option.

Parameters:

- string sessionKey
- int orgId

Returns:

- boolean status - Get the config option value

Chapter 494. Method: getDetails

HTTP **GET**

Description:

The detailed information about an organization given the organization ID.

Parameters:

- string sessionKey
- int orgId

Returns:

- * struct organization info
 - int "id"
 - string "name"
 - int "active_users" - number of active users in the organization
 - int "systems" - number of systems in the organization
 - int "trusts" - number of trusted organizations
 - int "system_groups" - number of system groups in the organization (optional)
 - int "activation_keys" - number of activation keys in the organization (optional)
 - int "kickstart_profiles" - number of kickstart profiles in the organization (optional)
 - int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

Chapter 495. Method: getDetails

HTTP **GET**

Description:

The detailed information about an organization given the organization name.

Parameters:

- string `sessionKey`
- string `name`

Returns:

- * struct organization info
 - int "id"
 - string "name"
 - int "active_users" - number of active users in the organization
 - int "systems" - number of systems in the organization
 - int "trusts" - number of trusted organizations
 - int "system_groups" - number of system groups in the organization (optional)
 - int "activation_keys" - number of activation keys in the organization (optional)
 - int "kickstart_profiles" - number of kickstart profiles in the organization (optional)
 - int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

Chapter 496. Method: getPolicyForScapFileUpload

HTTP **GET**

Description:

Get the status of SCAP detailed result file upload settings for the given organization.

Parameters:

- string sessionKey
- int orgId

Returns:

- struct scap_upload_info
 - boolean "enabled" - Aggregation of detailed SCAP results is enabled.
 - int "size_limit" - Limit (in Bytes) for a single SCAP file upload.

Chapter 497. Method: getPolicyForScapResultDeletion

HTTP **GET**

Description:

Get the status of SCAP result deletion settings for the given organization.

Parameters:

- string sessionKey
- int orgId

Returns:

- struct scap_deletion_info
 - boolean "enabled" - Deletion of SCAP results is enabled
 - int "retention_period" - Period (in days) after which a scan can be deleted (if enabled).

Chapter 498. Method: isContentStagingEnabled

HTTP **GET**

Description:

Get the status of content staging settings for the given organization. Returns true if enabled, false otherwise.

Parameters:

- string sessionKey
- int orgId

Returns:

- boolean status - Get the status of content staging settings

Chapter 499. Method: isErrataEmailNotifsForOrg

HTTP **GET**

Description:

Returns whether errata e-mail notifications are enabled for the organization

Parameters:

- string sessionKey
- int orgId

Returns:

- boolean status - Returns the status of the errata e-mail notification setting for the organization

Chapter 500. Method: isOrgConfigManagedByOrgAdmin

HTTP **GET**

Description:

Returns whether Organization Administrator is able to manage his organization configuration. This may have a high impact on general #product() performance.

Parameters:

- string sessionKey
- int orgId

Returns:

- boolean status - Returns the status org admin management setting

Chapter 501. Method: listOrgs

HTTP **GET**

Description:

Returns the list of organizations.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct organization info
 - int "id"
 - string "name"
 - int "active_users" - number of active users in the organization
 - int "systems" - number of systems in the organization
 - int "trusts" - number of trusted organizations
 - int "system_groups" - number of system groups in the organization (optional)
 - int "activation_keys" - number of activation keys in the organization (optional)
 - int "kickstart_profiles" - number of kickstart profiles in the organization (optional)
 - int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

Chapter 502. Method: listUsers

HTTP **GET**

Description:

Returns the list of users in a given organization.

Parameters:

- string sessionKey
- int orgId

Returns:

- array :
- struct user
 - string "login"
 - string "login_uc"
 - string "name"
 - string "email"
 - boolean "is_org_admin"

Chapter 503. Method: migrateSystems (Deprecated)

HTTP **POST**

Description:

Transfer systems from one organization to another. If executed by a #product() administrator, the systems will be transferred from their current organization to the organization specified by the toOrgId. If executed by an organization administrator, the systems must exist in the same organization as that administrator and the systems will be transferred to the organization specified by the toOrgId. In any scenario, the origination and destination organizations must be defined in a trust.

Note: This method is deprecated and will be removed in a future API version. Please use transferSystems instead.

Deprecated - being replaced by org.transferSystems(User loggedInUser, Integer toOrgId, List(Integer) sids)

Parameters:

- string sessionKey
- int toOrgId - ID of the organization where the system(s) will be transferred to.
- int array sids

Returns:

- int array serverIdTransferred

Chapter 504. Method: setClmSyncPatchesConfig

HTTP **POST**

Description:

Sets the content lifecycle management patch synchronization config option.

Parameters:

- string sessionKey
- int orgId
- boolean value - The config option value

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 505. Method: setContentStaging

HTTP **POST**

Description:

Set the status of content staging for the given organization.

Parameters:

- string sessionKey
- int orgId
- boolean enable - Use true/false to enable/disable

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 506. Method: setErrataEmailNotifsForOrg

HTTP **POST**

Description:

Dis/enables errata e-mail notifications for the organization

Parameters:

- string sessionKey
- int orgId
- boolean enable - Use true/false to enable/disable

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 507. Method: setOrgConfigManagedByOrgAdmin

HTTP **POST**

Description:

Sets whether Organization Administrator can manage his organization configuration. This may have a high impact on general #product() performance.

Parameters:

- string sessionKey
- int orgId
- boolean enable - Use true/false to enable/disable

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 508. Method: setPolicyForScapFileUpload

HTTP **POST**

Description:

Set the status of SCAP detailed result file upload settings for the given organization.

Parameters:

- string sessionKey
- int orgId
- struct newSettings
 - boolean "enabled" - Aggregation of detailed SCAP results is enabled.
 - int "size_limit" - Limit (in Bytes) for a single SCAP file upload.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 509. Method: setPolicyForScapResultDeletion

HTTP **POST**

Description:

Set the status of SCAP result deletion settings for the given organization.

Parameters:

- string sessionKey
- int orgId
- struct newSettings
 - boolean "enabled" - Deletion of SCAP results is enabled
 - int "retention_period" - Period (in days) after which a scan can be deleted (if enabled).

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 510. Method: transferSystems

HTTP **POST**

Description:

Transfer systems from one organization to another. If executed by a #product() administrator, the systems will be transferred from their current organization to the organization specified by the toOrgId. If executed by an organization administrator, the systems must exist in the same organization as that administrator and the systems will be transferred to the organization specified by the toOrgId. In any scenario, the origination and destination organizations must be defined in a trust.

Parameters:

- string sessionKey
- int toOrgId - ID of the organization where the system(s) will be transferred to.
- int array sids

Returns:

- int array serverIdTransferred

Chapter 511. Method: updateName

HTTP **POST**

Description:

Updates the name of an organization

Parameters:

- string sessionKey
- int orgId
- string name - Organization name. Must meet same criteria as in the web UI.

Returns:

- * struct organization info
 - int "id"
 - string "name"
 - int "active_users" - number of active users in the organization
 - int "systems" - number of systems in the organization
 - int "trusts" - number of trusted organizations
 - int "system_groups" - number of system groups in the organization (optional)
 - int "activation_keys" - number of activation keys in the organization (optional)
 - int "kickstart_profiles" - number of kickstart profiles in the organization (optional)
 - int "configuration_channels" - number of configuration channels in the organization (optional)
 - boolean "staging_content_enabled" - is staging content enabled in organization (optional)

org.trusts

Chapter 512. Available methods

- `addTrust`
- `getDetails`
- `listChannelsConsumed`
- `listChannelsProvided`
- `listOrgs`
- `listSystemsAffected`
- `listTrusts`
- `removeTrust`

Chapter 513. Description

Contains methods to access common organization trust information available from the web interface.

Namespace:

org.trusts

Chapter 514. Method: addTrust

HTTP **POST**

Description:

Add an organization to the list of trusted organizations.

Parameters:

- string sessionKey
- int orgId
- int trustOrgId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 515. Method: getDetails

HTTP **GET**

Description:

The trust details about an organization given the organization's ID.

Parameters:

- string sessionKey
- int orgId - Id of the trusted organization

Returns:

- struct org trust details
 - dateTime.iso8601 "created" - Date the organization was created
 - dateTime.iso8601 "trusted_since" - Date the organization was defined as trusted
 - int "channels_provided" - Number of channels provided by the organization.
 - int "channels_consumed" - Number of channels consumed by the organization.
 - int "systems_migrated_to" - (Deprecated by systems_transferred_to) Number of systems transferred to the organization.
 - int "systems_migrated_from" - (Deprecated by systems_transferred_from) Number of systems transferred from the organization.
 - int "systems_transferred_to" - Number of systems transferred to the organization.
 - int "systems_transferred_from" - Number of systems transferred from the organization.

Chapter 516. Method: listChannelsConsumed

HTTP **GET**

Description:

Lists all software channels that the organization given may consume from the user's organization.

Parameters:

- string sessionKey
- int orgId - Id of the trusted organization

Returns:

- array :
 - struct channel info
 - int "channel_id"
 - string "channel_name"
 - int "packages"
 - int "systems"

Chapter 517. Method: listChannelsProvided

HTTP **GET**

Description:

Lists all software channels that the organization given is providing to the user's organization.

Parameters:

- string sessionKey
- int orgId - Id of the trusted organization

Returns:

- array :
 - struct channel info
 - int "channel_id"
 - string "channel_name"
 - int "packages"
 - int "systems"

Chapter 518. Method: listOrgs

HTTP **GET**

Description:

List all organizations trusted by the user's organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct trusted organizations
 - int "org_id"
 - string "org_name"
 - int "shared_channels"

Chapter 519. Method: listSystemsAffected

HTTP **GET**

Description:

Get a list of systems within the trusted organization that would be affected if the trust relationship was removed. This basically lists systems that are sharing at least (1) package.

Parameters:

- string sessionKey
- int orgId
- string trustOrgId

Returns:

- array :
 - struct affected systems
 - int "systemId"
 - string "systemName"

Chapter 520. Method: listTrusts

HTTP **GET**

Description:

Returns the list of trusted organizations.

Parameters:

- string sessionKey
- int orgId

Returns:

- array :
- struct trusted organizations
 - int "orgId"
 - string "orgName"
 - boolean "trustEnabled"

Chapter 521. Method: removeTrust

HTTP **POST**

Description:

Remove an organization to the list of trusted organizations.

Parameters:

- string sessionKey
- int orgId
- int trustOrgId

Returns:

- int - 1 on success, exception thrown otherwise.

packages

Chapter 522. Available methods

- `findByNvrea`
- `getDetails`
- `getPackage`
- `getPackageUrl`
- `listChangelog`
- `listDependencies`
- `listFiles`
- `listProvidingChannels`
- `listProvidingErrata`
- `listSourcePackages`
- `removePackage`
- `removeSourcePackage`

Chapter 523. Description

Methods to retrieve information about the Packages contained within this server.

Namespace:

packages

Chapter 524. Method: findByNvrea

HTTP **GET**

Description:

Lookup the details for packages with the given name, version, release, architecture label, and (optionally) epoch.

Parameters:

- string sessionKey
- string name
- string version
- string release
- string epoch - If set to something other than empty string, strict matching will be used and the epoch string must be correct. If set to an empty string, if the epoch is null or there is only one NVRA combination, it will be returned. (Empty string is recommended.)
- string archLabel

Returns:

- array :
- struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"
 - dateTime.iso8601 "last_modified"
 - string "path" - the path on that file system that the package resides
 - boolean "part_of_retracted_patch" - true if the package is a part of a retracted patch
 - string "provider" - the provider of the package, determined by the gpg key it was signed

with.

Chapter 525. Method: getDetails

HTTP **GET**

Description:

Retrieve details for the package with the ID.

Parameters:

- string sessionKey
- int pid

Returns:

- struct package
 - int "id"
 - string "name"
 - string "epoch"
 - string "version"
 - string "release"
 - string "arch_label"
 - string array "providing_channels" - Channel label providing this package.
 - string "build_host"
 - string "description"
 - string "checksum"
 - string "checksum_type"
 - string "vendor"
 - string "summary"
 - string "cookie"
 - string "license"
 - string "file"
 - string "build_date"

-
- string "last_modified_date"
 - string "size"
 - string "path" - The path on the SUSE Manager server's file system that the package resides.
 - string "payload_size"

Chapter 526. Method: getPackage

HTTP **GET**

Description:

Retrieve the package file associated with a package. (Consider using `#getPackageUrlpackages.getPackageUrl` for larger files.)

Parameters:

- string `sessionKey`
- int `pid`

Returns:

- byte array binary object - package file

Chapter 527. Method: getPackageUrl

HTTP **GET**

Description:

Retrieve the url that can be used to download a package. This will expire after a certain time period.

Parameters:

- string sessionKey
- int pid

Returns:

- string - the download url

Chapter 528. Method: listChangelog

HTTP **GET**

Description:

List the change log for a package.

Parameters:

- string sessionKey
- int pid

Returns:

- string

Chapter 529. Method: listDependencies

HTTP **GET**

Description:

List the dependencies for a package.

Parameters:

- string sessionKey
- int pid

Returns:

- array :
 - struct dependency
 - string "dependency"
 - string "dependency_type" - One of the following:
 - requires
 - conflicts
 - obsoletes
 - provides
 - recommends
 - suggests
 - supplements
 - enhances
 - predepends
 - breaks
 - string "dependency_modifier"

Chapter 530. Method: listFiles

HTTP **GET**

Description:

List the files associated with a package.

Parameters:

- string sessionKey
- int pid

Returns:

- array :
 - struct file info
 - string "path"
 - string "type"
 - string "last_modified_date"
 - string "checksum"
 - string "checksum_type"
 - int "size"
 - string "linkto"

Chapter 531. Method: listProvidingChannels

HTTP **GET**

Description:

List the channels that provide the a package.

Parameters:

- string sessionKey
- int pid

Returns:

- array :
 - struct channel
 - string "label"
 - string "parent_label"
 - string "name"

Chapter 532. Method: listProvidingErrata

HTTP **GET**

Description:

List the errata providing the a package.

Parameters:

- string sessionKey
- int pid

Returns:

- array :
 - struct errata
 - string "advisory"
 - string "issue_date"
 - string "last_modified_date"
 - string "update_date"
 - string "synopsis"
 - string "type"

Chapter 533. Method: listSourcePackages

HTTP **GET**

Description:

List all source packages in user's organization.

Parameters:

- string sessionKey

Returns:

- array :
 - struct source_package
 - int "id"
 - string "name"

Chapter 534. Method: removePackage

HTTP **POST**

Description:

Remove a package from #product().

Parameters:

- string sessionKey
- int pid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 535. Method: removeSourcePackage

HTTP **POST**

Description:

Remove a source package.

Parameters:

- string sessionKey
- int psid - package source ID

Returns:

- int - 1 on success, exception thrown otherwise.

packages.provider

Chapter 536. Available methods

- `associateKey`
- `list`
- `listKeys`

Chapter 537. Description

Methods to retrieve information about Package Providers associated with packages.

Namespace:

packages.provider

Chapter 538. Method: associateKey

HTTP **POST**

Description:

Associate a package security key and with the package provider. If the provider or key doesn't exist, it is created. User executing the request must be a #product() administrator.

Parameters:

- string sessionKey
- string providerName - The provider name
- string key - The actual key
- string type - The type of the key. Currently, only 'gpg' is supported

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 539. Method: list

HTTP **POST**

Description:

List all Package Providers. User executing the request must be a `#product()` administrator.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct package provider
 - string "name"
 - array "keys"
- struct package security key
 - string "key"
 - string "type"

Chapter 540. Method: listKeys

HTTP **GET**

Description:

List all security keys associated with a package provider. User executing the request must be a #product() administrator.

Parameters:

- string sessionKey
- string providerName - The provider name

Returns:

- array :
- struct package security key
 - string "key"
 - string "type"

packages.search

Chapter 541. Available methods

- `advanced`
- `advancedWithActKey`
- `advancedWithChannel`
- `name`
- `nameAndDescription`
- `nameAndSummary`

Chapter 542. Description

Methods to interface to package search capabilities in search server..

Namespace:

packages.search

Chapter 543. Method: advanced

HTTP **GET**

Description:

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax. Lucene Query Parser syntax is defined at http://lucene.apache.org/java/3_5_0/queryparsersyntax.html_blank lucene.apache.org. Fields searchable for Packages: name, epoch, version, release, arch, description, summary Lucene Query Example: "name:kernel AND version:2.6.18 AND -description:devel"

Parameters:

- string sessionKey
- string luceneQuery - a query written in the form of Lucene QueryParser Syntax

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

Chapter 544. Method: advancedWithActKey

HTTP **GET**

Description:

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax, additionally this method will limit results to those which are associated with a given activation key. Lucene Query Parser syntax is defined at http://lucene.apache.org/java/3_5_0/queryparsersyntax.html_blank lucene.apache.org. Fields searchable for Packages: name, epoch, version, release, arch, description, summary Lucene Query Example: "name:kernel AND version:2.6.18 AND -description:devel"

Parameters:

- string sessionKey
- string luceneQuery - a query written in the form of Lucene QueryParser Syntax
- string activationKey - activation key to look for packages in

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

Chapter 545. Method: advancedWithChannel

HTTP **GET**

Description:

Advanced method to search lucene indexes with a passed in query written in Lucene Query Parser syntax, additionally this method will limit results to those which are in the passed in channel label. Lucene Query Parser syntax is defined at http://lucene.apache.org/java/3_5_0/queryparsersyntax.html_blank lucene.apache.org. Fields searchable for Packages: name, epoch, version, release, arch, description, summary Lucene Query Example: "name:kernel AND version:2.6.18 AND -description:devel"

Parameters:

- string sessionKey
- string luceneQuery - a query written in the form of Lucene QueryParser Syntax
- string channelLabel - the channel Label

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

Chapter 546. Method: name

HTTP **GET**

Description:

Search the lucene package indexes for all packages which match the given name.

Parameters:

- string sessionKey
- string name - package name to search for

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

Chapter 547. Method: nameAndDescription

HTTP **GET**

Description:

Search the lucene package indexes for all packages which match the given query in name or description

Parameters:

- string sessionKey
- string query - text to match in package name or description

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

Chapter 548. Method: nameAndSummary

HTTP **GET**

Description:

Search the lucene package indexes for all packages which match the given query in name or summary.

Parameters:

- string sessionKey
- string query – text to match in package name or summary

Returns:

- array :
- struct package overview
 - int "id"
 - string "name"
 - string "summary"
 - string "description"
 - string "version"
 - string "release"
 - string "arch"
 - string "epoch"
 - string "provider"

preferences.locale

Chapter 549. Available methods

- `listLocales`
- `listTimeZones`
- `setLocale`
- `setTimeZone`

Chapter 550. Description

Provides methods to access and modify user locale information

Namespace:

preferences.locale

Chapter 551. Method: listLocales

HTTP **GET**

Description:

Returns a list of all understood locales. Can be used as input to setLocale.

Parameters:

Returns:

- string array Locale code.

Chapter 552. Method: listTimeZones

HTTP **GET**

Description:

Returns a list of all understood timezones. Results can be used as input to setTimeZone.

Parameters:

Returns:

- array :
- struct timezone
 - int "time_zone_id" - unique identifier for timezone
 - string "olson_name" - name as identified by the Olson database

Chapter 553. Method: setLocale

HTTP **POST**

Description:

Set a user's locale.

Parameters:

- string sessionKey
- string login - User's login name.
- string locale - Locale to set. (from listLocales)

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 554. Method: setTimeZone

HTTP **POST**

Description:

Set a user's timezone.

Parameters:

- string sessionKey
- string login - User's login name.
- int tzid - Timezone ID. (from listTimeZones)

Returns:

- int - 1 on success, exception thrown otherwise.

proxy

Chapter 555. Available methods

- `activateProxy`
- `containerConfig`
- `containerConfig`
- `createMonitoringScout`
- `deactivateProxy`
- `isProxy`
- `listAvailableProxyChannels`
- `listProxies`
- `listProxyClients`

Chapter 556. Description

Provides methods to activate/deactivate a proxy server.

Namespace:

proxy

Chapter 557. Method: activateProxy

HTTP **POST**

Description:

Activates the proxy identified by the given client certificate i.e. systemid file.

Parameters:

- string clientcert - client certificate file
- string version - Version of proxy to be registered.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 558. Method: containerConfig

HTTP **POST**

Description:

Compute and download the configuration for proxy containers

Parameters:

- string sessionKey
- string proxyName
- int proxyPort
- string server
- int maxCache
- string email
- string rootCA
- string array intermediateCAs - intermediate CAs used to sign the SSL certificate in PEM format
- string proxyCrt
- string proxyKey

Returns:

- byte array binary object - package file

Chapter 559. Method: containerConfig

HTTP **POST**

Description:

Compute and download the configuration for proxy containers

Parameters:

- string sessionKey
- string proxyName
- int proxyPort
- string server
- int maxCache
- string email
- string caCrt
- string caKey
- string caPassword
- string array cnames - Proxy alternate cnames to set in the SSL certificate
- string country
- string state
- string city
- string org
- string orgUnit
- string sslEmail

Returns:

- byte array binary object - package file

Chapter 560. Method: createMonitoringScout

HTTP **POST**

Description:

Create Monitoring Scout for proxy.

Parameters:

- string clientcert - client certificate file

Returns:

- string

Available since API version: 10.7

Chapter 561. Method: deactivateProxy

HTTP **POST**

Description:

Deactivates the proxy identified by the given client certificate i.e. systemid file.

Parameters:

- string clientcert - client certificate file

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 562. Method: isProxy

HTTP **GET**

Description:

Test, if the system identified by the given client certificate i.e. systemid file, is proxy.

Parameters:

- string clientcert - client certificate file

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 563. Method: listAvailableProxyChannels

HTTP **GET**

Description:

List available version of proxy channel for system identified by the given client certificate i.e. systemid file.

Parameters:

- string clientcert - client certificate file

Returns:

- string array version

Available since API version: 10.5

Chapter 564. Method: listProxies

HTTP **GET**

Description:

List the proxies within the user's organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 565. Method: listProxyClients

HTTP **GET**

Description:

List the clients directly connected to a given Proxy.

Parameters:

- string sessionKey
- int proxyId

Returns:

- int array clientId

recurring

Chapter 566. Available methods

- delete
- listByEntity
- lookupById

Chapter 567. Description

Provides methods to handle recurring actions for minions, system groups and organizations.

Namespace:

recurring

Chapter 568. Method: delete

HTTP **POST**

Description:

Delete a recurring action with the given action ID.

Parameters:

- string sessionKey
- int id - the action ID

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 569. Method: listByEntity

HTTP **GET**

Description:

Return a list of recurring actions for a given entity.

Parameters:

- string `sessionKey`
- string `"type"` - the type of the target entity. One of the following:
 - `minion`
 - `group`
 - `org`
- int `id` - the ID of the target entity

Returns:

- array :
- struct recurring action information (some fields may be absent for some action types)
 - int `"id"`
 - string `"name"`
 - int `"entity_id"`
 - string `"entity_type"`
 - string `"cron_expr"`
 - `dateTime.iso8601` `"created"`
 - string `"creator"`
 - boolean `"test"`
 - string array `"states"` - the ordered list of states to be executed by a custom state action
 - boolean `"active"`

Chapter 570. Method: lookupById

HTTP **GET**

Description:

Find a recurring action with the given action ID.

Parameters:

- string sessionKey
- int id - the action ID

Returns:

- * struct recurring action information (some fields may be absent for some action types)
 - int "id"
 - string "name"
 - int "entity_id"
 - string "entity_type"
 - string "cron_expr"
 - dateTime.iso8601 "created"
 - string "creator"
 - boolean "test"
 - string array "states" - the ordered list of states to be executed by a custom state action
 - boolean "active"

recurring.custom

Chapter 571. Available methods

- create
- listAvailable
- update

Chapter 572. Description

Provides methods to handle recurring custom states for minions, system groups and organizations.

Namespace:

recurring.custom

Chapter 573. Method: create

HTTP **POST**

Description:

Create a new recurring custom state action.

Parameters:

- string `sessionKey`
- struct `actionProps`
 - string `"entity_type"` - the type of the target entity. One of the following:
 - `minion`
 - `group`
 - `org`
 - int `"entity_id"` - the ID of the target entity
 - string `"name"` - the name of the recurring action
 - string `"cron_expr"` - the execution frequency of the action as a cron expression
 - string array `"states"` - the ordered list of custom state names to be executed
 - boolean `"test"` - whether the action should be executed in test mode (optional)

Returns:

- int `id` - the ID of the newly created recurring action

Chapter 574. Method: listAvailable

HTTP **GET**

Description:

List all the custom states available to the user.

Parameters:

- string sessionKey

Returns:

- string array the list of custom channels available to the user

Chapter 575. Method: update

HTTP **POST**

Description:

Update a recurring custom state action.

Parameters:

- string `sessionKey`
- struct `actionProps`
 - int `"id"` - the ID of the action to update
 - string `"name"` - the name of the action (optional)
 - string `"cron_expr"` - the execution frequency of the action (optional)
 - string array `"states"` - the ordered list of custom state names to be executed (optional)
 - boolean `"test"` - whether the action should be executed in test mode (optional)
 - boolean `"active"` - whether the action should be active (optional)

Returns:

- int `id` - the ID of the updated recurring action

recurring.highstate

Chapter 576. Available methods

- create
- update

Chapter 577. Description

Provides methods to handle recurring highstates for minions, system groups and organizations.

Namespace:

recurring.highstate

Chapter 578. Method: create

HTTP **POST**

Description:

Create a new recurring highstate action.

Parameters:

- string `sessionKey`
- struct `actionProps`
 - string `"entity_type"` - the type of the target entity. One of the following:
 - `minion`
 - `group`
 - `org`
 - int `"entity_id"` - the ID of the target entity
 - string `"name"` - the name of the recurring action
 - string `"cron_expr"` - the execution frequency of the action as a cron expression
 - boolean `"test"` - whether the action should be executed in test mode (optional)

Returns:

- int `id` - the ID of the newly created recurring action

Chapter 579. Method: update

HTTP **POST**

Description:

Update the properties of a recurring highstate action.

Parameters:

- string `sessionKey`
- struct `actionProps`
 - int `"id"` - the ID of the action to update
 - string `"name"` - the name of the action (optional)
 - string `"cron_expr"` - the execution frequency of the action (optional)
 - boolean `"test"` - whether the action should be executed in test mode (optional)
 - boolean `"active"` - whether the action should be active (optional)

Returns:

- int `id` - the ID of the updated recurring action

recurringaction

Chapter 580. Available methods

- create
- delete
- listByEntity
- lookupById
- update

Chapter 581. Description

Provides methods to handle recurring actions for minions, system groups and organizations.

Deprecated - This namespace will be removed in a future API version. To work with recurring actions, please check out the newer 'recurring' namespace.

Namespace:

recurringaction

Chapter 582. Method: create (Deprecated)

HTTP **POST**

Description:

Create a new recurring highstate action.

Deprecated – This method will be removed in a future API version. To create recurring actions, please use either 'recurring.highstate.create' or 'recurring.custom.create' instead.

Parameters:

- string sessionKey
- struct actionProps
 - string "entity_type" - the type of the target entity. One of the following:
 - minion
 - group
 - org
 - int "entity_id" - the ID of the target entity
 - string "name" - the name of the action
 - string "cron_expr" - the execution frequency of the action
 - boolean "test" - whether the action should be executed in test mode (optional)

Returns:

- int id - the ID of the recurring action

Chapter 583. Method: delete (Deprecated)

HTTP **POST**

Description:

Delete a recurring action with the given action ID.

Deprecated – This method will be removed in a future API version. To work with recurring actions, please check out the newer 'recurring' namespace.

Parameters:

- string sessionKey
- int actionId – the action ID

Returns:

- int id – the ID of the recurring action

Chapter 584. Method: listByEntity (Deprecated)

HTTP **GET**

Description:

Return a list of recurring actions for a given entity.

Deprecated – This method will be removed in a future API version. To work with recurring actions, please check out the newer 'recurring' namespace.

Parameters:

- string sessionKey
- string "entityType" - the type of the target entity. One of the following:
 - MINION
 - GROUP
 - ORG
- int entityId - the ID of the target entity

Returns:

- array :
- struct recurring action information (some fields may be absent for some action types)
 - int "id"
 - string "name"
 - int "entity_id"
 - string "entity_type"
 - string "cron_expr"
 - dateTime.iso8601 "created"
 - string "creator"
 - boolean "test"
 - string array "states" - the ordered list of states to be executed by a custom state action

- boolean "active"

Chapter 585. Method: lookupById (Deprecated)

HTTP **GET**

Description:

Find a recurring action with the given action ID.

Deprecated – This method will be removed in a future API version. To work with recurring actions, please check out the newer 'recurring' namespace.

Parameters:

- string sessionKey
- int actionId – the action ID

Returns:

- * struct recurring action information (some fields may be absent for some action types)
 - int "id"
 - string "name"
 - int "entity_id"
 - string "entity_type"
 - string "cron_expr"
 - dateTime.iso8601 "created"
 - string "creator"
 - boolean "test"
 - string array "states" – the ordered list of states to be executed by a custom state action
 - boolean "active"

Chapter 586. Method: update (Deprecated)

HTTP **POST**

Description:

Update a recurring highstate action.

Deprecated – This method will be removed in a future API version. To update recurring actions, please use either 'recurring.highstate.update' or 'recurring.custom.update' instead.

Parameters:

- string `sessionKey`
- struct `actionProps`
 - int `"id"` - the ID of the action to update
 - string `"name"` - the name of the action (optional)
 - string `"cron_expr"` - the execution frequency of the action (optional)
 - boolean `"test"` - whether the action should be executed in test mode (optional)
 - boolean `"active"` - whether the action should be active (optional)

Returns:

- int `id` - the ID of the recurring action

saltkey

Chapter 587. Available methods

- `accept`
- `acceptedList`
- `delete`
- `deniedList`
- `pendingList`
- `reject`
- `rejectedList`

Chapter 588. Description

Provides methods to manage salt keys

Namespace:

saltkey

Chapter 589. Method: accept

HTTP **POST**

Description:

Accept a minion key

Parameters:

- string sessionKey
- string minionId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 590. Method: acceptedList

HTTP **GET**

Description:

List accepted salt keys

Parameters:

- string sessionKey

Returns:

- string array Accepted salt key list

Chapter 591. Method: delete

HTTP **POST**

Description:

Delete a minion key

Parameters:

- string sessionKey
- string minionId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 592. Method: deniedList

HTTP **GET**

Description:

List of denied salt keys

Parameters:

- string sessionKey

Returns:

- string array Denied salt key list

Chapter 593. Method: pendingList

HTTP **GET**

Description:

List pending salt keys

Parameters:

- string sessionKey

Returns:

- string array Pending salt key list

Chapter 594. Method: reject

HTTP **POST**

Description:

Reject a minion key

Parameters:

- string sessionKey
- string minionId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 595. Method: rejectedList

HTTP **GET**

Description:

List of rejected salt keys

Parameters:

- string sessionKey

Returns:

- string array Rejected salt key list

schedule

Chapter 596. Available methods

- `archiveActions`
- `cancelActions`
- `deleteActions`
- `failSystemAction`
- `failSystemAction`
- `listAllActions`
- `listAllArchivedActions`
- `listAllCompletedActions`
- `listArchivedActions`
- `listCompletedActions`
- `listCompletedSystems`
- `listFailedActions`
- `listFailedSystems`
- `listInProgressActions`
- `listInProgressSystems`
- `rescheduleActions`

Chapter 597. Description

Methods to retrieve information about scheduled actions.

Namespace:

schedule

Chapter 598. Method: archiveActions

HTTP **POST**

Description:

Archive all actions in the given list.

Parameters:

- string sessionKey
- int array actionIds

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 599. Method: cancelActions

HTTP **POST**

Description:

Cancel all actions in given list. If an invalid action is provided, none of the actions given will be canceled.

Parameters:

- string sessionKey
- int array actionIds

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 600. Method: deleteActions

HTTP **POST**

Description:

Delete all archived actions in the given list.

Parameters:

- string sessionKey
- int array actionIds

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 601. Method: failSystemAction

HTTP **POST**

Description:

Fail specific event on specified system

Parameters:

- string sessionKey
- int sid
- int actionId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 602. Method: failSystemAction

HTTP **POST**

Description:

Fail specific event on specified system

Parameters:

- string sessionKey
- int sid
- int actionId
- string message

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 603. Method: listAllActions

HTTP **GET**

Description:

Returns a list of all actions. This includes completed, in progress, failed and archived actions.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `action`
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)
 - `dateTime.iso8601` `"earliest"` - the earliest date and time the action will be performed
 - int `"prerequisite"` - ID of the prerequisite action (optional)
 - int `"completedSystems"` - number of systems that completed the action
 - int `"failedSystems"` - number of systems that failed the action
 - int `"inProgressSystems"` - number of systems that are in progress

Chapter 604. Method: listAllArchivedActions

HTTP **GET**

Description:

Returns a list of actions that have been archived.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `action`
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)
 - `dateTime.iso8601` `"earliest"` - the earliest date and time the action will be performed
 - int `"prerequisite"` - ID of the prerequisite action (optional)
 - int `"completedSystems"` - number of systems that completed the action
 - int `"failedSystems"` - number of systems that failed the action
 - int `"inProgressSystems"` - number of systems that are in progress

Chapter 605. Method: listAllCompletedActions

HTTP **GET**

Description:

Returns a list of actions that have been completed.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `action`
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)
 - `dateTime.iso8601` `"earliest"` - the earliest date and time the action will be performed
 - int `"prerequisite"` - ID of the prerequisite action (optional)
 - int `"completedSystems"` - number of systems that completed the action
 - int `"failedSystems"` - number of systems that failed the action
 - int `"inProgressSystems"` - number of systems that are in progress

Chapter 606. Method: listArchivedActions

HTTP **GET**

Description:

Returns a list of actions that have been archived.

Parameters:

- string sessionKey

Returns:

- array :
- struct action
 - int "id" - action ID
 - string "name" - action name
 - string "type" - action type
 - string "scheduler" - the user that scheduled the action (optional)
 - dateTime.iso8601 "earliest" - the earliest date and time the action will be performed
 - int "prerequisite" - ID of the prerequisite action (optional)
 - int "completedSystems" - number of systems that completed the action
 - int "failedSystems" - number of systems that failed the action
 - int "InProgressSystems" - number of systems that are in progress

Chapter 607. Method: listCompletedActions

HTTP **GET**

Description:

Returns a list of actions that have completed successfully.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `action`
 - int `"id"` - action ID
 - string `"name"` - action name
 - string `"type"` - action type
 - string `"scheduler"` - the user that scheduled the action (optional)
 - `dateTime.iso8601` `"earliest"` - the earliest date and time the action will be performed
 - int `"prerequisite"` - ID of the prerequisite action (optional)
 - int `"completedSystems"` - number of systems that completed the action
 - int `"failedSystems"` - number of systems that failed the action
 - int `"inProgressSystems"` - number of systems that are in progress

Chapter 608. Method: listCompletedSystems

HTTP **GET**

Description:

Returns a list of systems that have completed a specific action.

Parameters:

- string sessionKey
- int actionId

Returns:

- array :
- struct system
 - int "server_id"
 - string "server_name" - server name
 - string "base_channel" - base channel used by the server
 - dateTime.iso8601 "timestamp" - the time the action was completed
 - string "message" - optional message containing details on the execution of the action.
For example, if the action failed, this will contain the failure text.

Chapter 609. Method: listFailedActions

HTTP **GET**

Description:

Returns a list of actions that have failed.

Parameters:

- string sessionKey

Returns:

- array :
- struct action
 - int "id" - action ID
 - string "name" - action name
 - string "type" - action type
 - string "scheduler" - the user that scheduled the action (optional)
 - dateTime.iso8601 "earliest" - the earliest date and time the action will be performed
 - int "prerequisite" - ID of the prerequisite action (optional)
 - int "completedSystems" - number of systems that completed the action
 - int "failedSystems" - number of systems that failed the action
 - int "inProgressSystems" - number of systems that are in progress

Chapter 610. Method: listFailedSystems

HTTP **GET**

Description:

Returns a list of systems that have failed a specific action.

Parameters:

- string sessionKey
- int actionId

Returns:

- array :
- struct system
 - int "server_id"
 - string "server_name" - server name
 - string "base_channel" - base channel used by the server
 - dateTime.iso8601 "timestamp" - the time the action was completed
 - string "message" - optional message containing details on the execution of the action.
For example, if the action failed, this will contain the failure text.

Chapter 611. Method: listInProgressActions

HTTP **GET**

Description:

Returns a list of actions that are in progress.

Parameters:

- string sessionKey

Returns:

- array :
- struct action
 - int "id" - action ID
 - string "name" - action name
 - string "type" - action type
 - string "scheduler" - the user that scheduled the action (optional)
 - dateTime.iso8601 "earliest" - the earliest date and time the action will be performed
 - int "prerequisite" - ID of the prerequisite action (optional)
 - int "completedSystems" - number of systems that completed the action
 - int "failedSystems" - number of systems that failed the action
 - int "InProgressSystems" - number of systems that are in progress

Chapter 612. Method: listInProgressSystems

HTTP **GET**

Description:

Returns a list of systems that have a specific action in progress.

Parameters:

- string sessionKey
- int actionId

Returns:

- array :
- struct system
 - int "server_id"
 - string "server_name" - server name
 - string "base_channel" - base channel used by the server
 - dateTime.iso8601 "timestamp" - the time the action was completed
 - string "message" - optional message containing details on the execution of the action.
For example, if the action failed, this will contain the failure text.

Chapter 613. Method: rescheduleActions

HTTP **POST**

Description:

Reschedule all actions in the given list.

Parameters:

- string sessionKey
- int array actionIds
- boolean onlyFailed - True to only reschedule failed actions, False to reschedule all

Returns:

- int - 1 on success, exception thrown otherwise.

Sample Scripts

Chapter 614. JSON over HTTP API scripts

614.1. HTTP login with Curl

Below is an example of the login process using an authentication token with the JSON over HTTP API.

The JSON over HTTP API uses authentication tokens for access. The token is sent in a cookie called `pxt-session-cookie` in a response to a call to the `auth.login` endpoint. The `auth.login` endpoint accepts a POST request with a JSON body that has `login` and `password` properties in a top-level object.

```
$ API=https://manager.example.com/rhn/manager/api
$ curl -H "Content-Type: application/json" -d '{"login": "myusername", "password":
"mypass"}' -i $API/auth/login

HTTP/1.1 200 200
...
Set-Cookie: pxt-session-cookie=&tokenhash&; ...
...
{"success":true,"messages":[]}
```

Once the login is successful, the retrieved cookie must be added to each subsequent request for authenticated access.

614.2. HTTP GET example

Below is an example of an HTTP GET call to the `contentmanagement.lookupProject` API. In a GET request, method parameters must be sent as query string key-value pairs.

The JSON output is pretty-printed for clarity.

```

$ API=https://manager.example.com/rhn/manager/api
$ curl -H "Content-Type: application/json" --cookie "pxt-session-
cookie&tokenhash&gt;" \
> $API/contentmanagement/lookupProject?projectLabel=myproject
{
  "success": true,
  "result": {
    "name": "My Project",
    "description": "My CLM project",
    "id": 1,
    "label": "myproject",
    "orgId": 1
  }
}

```

614.3. HTTP POST example

Below is an example of an HTTP POST call to the `contentmanagement.createProject` API. In a POST request, method parameters can be sent as query string key-value pairs, as a JSON object in the request body, or a mix of both. `object` type parameters cannot be represented as a query string element and therefore must be sent in the request body. The following examples show both approaches.

The JSON output is pretty-printed for clarity.

614.3.1. Using the query string

```

$ API=https://manager.example.com/rhn/manager/api
$ curl -H "Content-Type: application/json" --cookie "pxt-session-
cookie&tokenhash&gt;" -X POST \
>
"$API/contentmanagement/createProject?projectLabel=myproject&name=My%20Project
&description="
{
  "success": true,
  "result": {
    "name": "My Project",
    "id": 1,
    "label": "myproject",
    "orgId": 1
  }
}

```

614.3.2. Using the request body

The request body must be a top-level JSON object that contains all the parameters as its properties.

```
$ API=https://manager.example.com/rhn/manager/api
$ curl -H "Content-Type: application/json" --cookie "pvt-session-
cookie&tokenhash;" \
> -d '{"projectLabel":"myproject","name":"My Project","description":""}' \
> $API/contentmanagement/createProject
{
  "success": true,
  "result": {
    "name": "My Project",
    "id": 1,
    "label": "myproject",
    "orgId": 1
  }
}
```

===Python 3 example

Below is an example of the `system.listActiveSystems` call being used.


```
#!/usr/bin/env python3
import requests
import pprint

MANAGER_URL = "https://manager.example.com/rhn/manager/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"
SSLVERIFY = "/path/to/CA" # or False to disable verify;

data = {"login": MANAGER_LOGIN, "password": MANAGER_PASSWORD}
response = requests.post(MANAGER_URL + '/auth/login', json=data, verify=SSLVERIFY)
print("LOGIN: {}".format(response.status_code, response.json()))

cookies = response.cookies
res2 = requests.get(MANAGER_URL + '/system/listActiveSystems', cookies=cookies, verify=SSLVERIFY)
print("RETCODE: {}".format(res2.status_code))
pprint.pprint(res2.json())

sysinfo = res2.json()['result'][0]

note = {"sid": sysinfo['id'], "subject": "Title", "body": "Content of the Note"}
res2 = requests.post(MANAGER_URL + '/system/addNote', json=note, cookies=cookies, verify=SSLVERIFY)
print("RETCODE: {}".format(res2.status_code))
pprint.pprint(res2.json())

res2 = requests.get(MANAGER_URL + '/system/listNotes?sid={}'.format(sysinfo['id']),
cookies=cookies, verify=SSLVERIFY)
print("RETCODE: {}".format(res2.status_code))
pprint.pprint(res2.json())

res2 = requests.post(MANAGER_URL + '/auth/logout', cookies=cookies, verify=SSLVERIFY)
print("RETCODE: {}".format(res2.status_code))
pprint.pprint(res2.json())
```

Chapter 615. XMLRPC Scripts

615.1. Perl Example

This Perl example shows the `system.listUserSystems` call being used to get a list of systems a user has access to. In the example below, the name of each system will be printed.

```
#!/usr/bin/perl
use Frontier::Client;

my $HOST = 'manager.example.com';
my $user = 'username';
my $pass = 'password';

my $client = new Frontier::Client(url => "http://$HOST/rpc/api");
my $session = $client->call('auth.login',$user, $pass);

my $systems = $client->call('system.listUserSystems', $session);
foreach my $system (@$systems) {
    print $system->{'name'}."\n";
}
$client->call('auth.logout', $session);
```

615.2. Python 2 Example

Below is an example of the `user.listUsers` call being used. Only the login of each user is printed.

```
#!/usr/bin/python
import xmlrpclib

MANAGER_URL = "http://manager.example.com/rpc/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"

client = xmlrpclib.Server(MANAGER_URL, verbose=0)

key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)
list = client.user.list_users(key)
for user in list:
    print user.get('login')

client.auth.logout(key)
```

The following code shows how to use date-time parameters. This code will schedule immediate

installation of package `rhnlb-2.5.22.9.el6.noarch` to system with id `1000000001`.

```
#!/usr/bin/python
from datetime import datetime
import time
import xmlrpclib

MANAGER_URL = "http://manager.example.com/rpc/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"

client = xmlrpclib.Server(MANAGER_URL, verbose=0)

key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)
package_list = client.packages.findByName(key, 'rhnlb', '2.5.22', '9.el6', ", ", 'noarch')
today = datetime.today()
earliest_occurrence = xmlrpclib.DateTime(today)
client.system.schedulePackageInstall(key, 1000000001, package_list[0]['id'],
earliest_occurrence)

client.auth.logout(key)
```

615.3. Python 3 with SSL Example

Below is an example of the `user.listUsers` call being called.

```
#!/usr/bin/env python3
from xmlrpc.client import ServerProxy
import ssl

MANAGER_URL = "https://manager.example.com/rpc/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"

# You might need to set to set other options depending on your
# server SSL configuration and your local SSL configuration
context = ssl.create_default_context()
client = ServerProxy(MANAGER_URL, context=context)
key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)

print(client.user.list_users(key))

client.auth.logout(key)
```

615.4. Python 3 Example

Below is an example of the `user.listUsers` call being called.

```
#!/usr/bin/env python3
from xmlrpc.client import ServerProxy

MANAGER_URL = "http://manager.example.com/rpc/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"

client = ServerProxy(MANAGER_URL)
key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)

print(client.user.list_users(key))

client.auth.logout(key)
```

The following code shows how to use date-time parameters. This code will schedule immediate installation of package `rhnlb-2.5.22.9.el6.noarch` to system with id `1000000001`.

```
#!/usr/bin/env python3
from datetime import datetime
from xmlrpc.client import ServerProxy

MANAGER_URL = "http://manager.example.com/rpc/api"
MANAGER_LOGIN = "username"
MANAGER_PASSWORD = "password"

client = ServerProxy(MANAGER_URL)

key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)
package_list = client.packages.findByName(key, 'rhnlb', '2.5.22', '9.el6', "", 'noarch')
earliest_occurrence = datetime.today()
client.system.schedulePackageInstall(key, 1000000001, [package_list[0]['id']],
earliest_occurrence)

client.auth.logout(key)
```

615.5. Ruby Example

Below is an example of the `channel.listAllChannels` API call. List of channel labels is printed.

```
#!/usr/bin/ruby
require "xmlrpc/client"

@MANAGER_URL = "http://manager.example.com/rpc/api"
@MANAGER_LOGIN = "username"
@MANAGER_PASSWORD = "password"

@client = XMLRPC::Client.new2(@MANAGER_URL)

@key = @client.call('auth.login', @MANAGER_LOGIN, @MANAGER_PASSWORD)
channels = @client.call('channel.listAllChannels', @key)
for channel in channels do
  p channel["label"]
end

@client.call('auth.logout', @key)
```

subscriptionmatching.pinnedsubscription

Chapter 616. Available methods

- create
- delete
- list

Chapter 617. Description

Provides the namespace for operations on Pinned Subscriptions

Namespace:

subscriptionmatching.pinnedsubscription

Chapter 618. Method: create

HTTP **POST**

Description:

Creates a Pinned Subscription based on given subscription and system

Parameters:

- string sessionKey
- int subscriptionId - Subscription ID
- int sid - System ID

Returns:

- * struct pinned subscription
 - int "id"
 - int "subscription_id"
 - int "system_id"

Chapter 619. Method: delete

HTTP **POST**

Description:

Deletes Pinned Subscription with given id

Parameters:

- string sessionKey
- int subscriptionId - Pinned Subscription ID

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 620. Method: list

HTTP **POST**

Description:

Lists all PinnedSubscriptions

Parameters:

- string sessionKey

Returns:

- array :
- struct pinned subscription
 - int "id"
 - int "subscription_id"
 - int "system_id"

sync.content

Chapter 621. Available methods

- `addChannel`
- `addChannels`
- `addCredentials`
- `deleteCredentials`
- `listChannels`
- `listCredentials`
- `listProducts`
- `synchronizeChannelFamilies`
- `synchronizeChannels`
- `synchronizeProducts`
- `synchronizeRepositories`
- `synchronizeSubscriptions`

Chapter 622. Description

Provides the namespace for the content synchronization methods.

Namespace:

sync.content

Chapter 623. Method: addChannel

HTTP **POST**

Description:

Add a new channel to the #product() database

Parameters:

- string sessionKey
- string channelLabel - Label of the channel to add
- string mirrorUrl - Sync from mirror temporarily

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 624. Method: addChannels

HTTP **POST**

Description:

Add a new channel to the #product() database

Parameters:

- string sessionKey
- string channelLabel - Label of the channel to add
- string mirrorUrl - Sync from mirror temporarily

Returns:

- string array enabled channel labels

Chapter 625. Method: addCredentials

HTTP **POST**

Description:

Add organization credentials (mirror credentials) to #product().

Parameters:

- string sessionKey
- string username - Organization credentials (Mirror credentials) username
- string password - Organization credentials (Mirror credentials) password
- boolean primary - Make this the primary credentials

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 626. Method: deleteCredentials

HTTP **POST**

Description:

Delete organization credentials (mirror credentials) from #product().

Parameters:

- string sessionKey
- string username - Username of credentials to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 627. Method: listChannels

HTTP **GET**

Description:

List all accessible channels.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `channel`
 - string `"arch"` - architecture of the channel
 - string `"description"` - description of the channel
 - string `"family"` - channel family label
 - boolean `"is_signed"` - channel has signed metadata
 - string `"label"` - label of the channel
 - string `"name"` - name of the channel
 - boolean `"optional"` - channel is optional
 - string `"parent"` - the label of the parent channel
 - string `"product_name"` - product name
 - string `"product_version"` - product version
 - string `"source_url"` - repository source URL
 - string `"status"` - 'available', 'unavailable' or 'installed'
 - string `"summary"` - channel summary
 - string `"update_tag"` - update tag
 - boolean `"installer_updates"` - is an installer update channel

Chapter 628. Method: listCredentials

HTTP **GET**

Description:

List organization credentials (mirror credentials) available in #product().

Parameters:

- string sessionKey

Returns:

- array :
- struct credentials
 - int "id" - ID of the credentials
 - string "user" - username
 - boolean "isPrimary" - primary

Chapter 629. Method: listProducts

HTTP **GET**

Description:

List all accessible products.

Parameters:

- string `sessionKey`

Returns:

- array :
- struct `product`
 - string `"friendly_name"` - friendly name of the product
 - string `"arch"` - architecture
 - string `"status"` - 'available', 'unavailable' or 'installed'
 - array `"channels"`
- struct `channel`
 - string `"arch"` - architecture of the channel
 - string `"description"` - description of the channel
 - string `"family"` - channel family label
 - boolean `"is_signed"` - channel has signed metadata
 - string `"label"` - label of the channel
 - string `"name"` - name of the channel
 - boolean `"optional"` - channel is optional
 - string `"parent"` - the label of the parent channel
 - string `"product_name"` - product name
 - string `"product_version"` - product version
 - string `"source_url"` - repository source URL
 - string `"status"` - 'available', 'unavailable' or 'installed'

-
- string "summary" - channel summary
 - string "update_tag" - update tag
 - boolean "installer_updates" - is an installer update channel
 - array "extensions"
 - struct extension product
 - string "friendly_name" - friendly name of extension product
 - string "arch" - architecture
 - string "status" - 'available', 'unavailable' or 'installed'
 - array "channels"
 - struct channel
 - string "arch" - architecture of the channel
 - string "description" - description of the channel
 - string "family" - channel family label
 - boolean "is_signed" - channel has signed metadata
 - string "label" - label of the channel
 - string "name" - name of the channel
 - boolean "optional" - channel is optional
 - string "parent" - the label of the parent channel
 - string "product_name" - product name
 - string "product_version" - product version
 - string "source_url" - repository source URL
 - string "status" - 'available', 'unavailable' or 'installed'
 - string "summary" - channel summary
 - string "update_tag" - update tag
 - boolean "installer_updates" - is an installer update channel
 - boolean "recommended" - recommended

Chapter 630. Method: synchronizeChannelFamilies

HTTP **POST**

Description:

Synchronize channel families between the Customer Center and the #product() database.

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 631. Method: synchronizeChannels

HTTP **POST**

Description:

(Deprecated) Synchronize channels between the Customer Center and the #product() database.

Parameters:

- string sessionKey
- string mirrorUrl - Sync from mirror temporarily

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 632. Method: synchronizeProducts

HTTP **POST**

Description:

Synchronize SUSE products between the Customer Center and the #product() database.

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 633. Method: synchronizeRepositories

HTTP **POST**

Description:

Synchronize repositories between the Customer Center and the #product() database.

Parameters:

- string sessionKey
- string mirrorUrl - Optional mirror url or null

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 634. Method: synchronizeSubscriptions

HTTP **POST**

Description:

Synchronize subscriptions between the Customer Center and the #product() database.

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

sync.master

Chapter 635. Available methods

- addToMaster
- create
- delete
- getDefaultMaster
- getMaster
- getMasterByLabel
- getMasterOrgs
- getMasters
- hasMaster
- makeDefault
- mapToLocal
- setCaCert
- setMasterOrgs
- unsetDefaultMaster
- update

Chapter 636. Description

Contains methods to set up information about known-"masters", for use on the "slave" side of ISS

Namespace:

sync.master

Chapter 637. Method: addToMaster

HTTP **POST**

Description:

Add a single organizations to the list of those the specified Master has exported to this Slave

Parameters:

- string sessionKey
- int masterId - Id of the desired Master
- struct orgMap
 - int "masterOrgId"
 - string "masterOrgName"
 - int "localOrgId"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 638. Method: create

HTTP **POST**

Description:

Create a new Master, known to this Slave.

Parameters:

- string sessionKey
- string label - Master's fully-qualified domain name

Returns:

- * struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

Chapter 639. Method: delete

HTTP **POST**

Description:

Remove the specified Master

Parameters:

- string sessionKey
- int masterId - Id of the Master to remove

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 640. Method: getDefaultMaster

HTTP **GET**

Description:

Return the current default-Master for this Slave

Parameters:

- string sessionKey

Returns:

- * struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

Chapter 641. Method: getMaster

HTTP **GET**

Description:

Find a Master by specifying its ID

Parameters:

- string sessionKey
- int masterId - ID of the desired Master

Returns:

- * struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

Chapter 642. Method: getMasterByLabel

HTTP **GET**

Description:

Find a Master by specifying its label

Parameters:

- string sessionKey
- string label - Label of the desired Master

Returns:

- * struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

Chapter 643. Method: getMasterOrgs

HTTP **GET**

Description:

List all organizations the specified Master has exported to this Slave

Parameters:

- string sessionKey
- int masterId - ID of the desired Master

Returns:

- array :
- struct IssMasterOrg info
 - int "masterOrgId"
 - string "masterOrgName"
 - int "localOrgId"

Chapter 644. Method: getMasters

HTTP **GET**

Description:

Get all the Masters this Slave knows about

Parameters:

- string sessionKey

Returns:

- array :
- struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

Chapter 645. Method: hasMaster

HTTP **POST**

Description:

Check if this host is reading configuration from an ISS master.

Parameters:

Returns:

- boolean master - True if has an ISS master, false otherwise

Chapter 646. Method: makeDefault

HTTP **POST**

Description:

Make the specified Master the default for this Slave's inter-server-sync

Parameters:

- string sessionKey
- int masterId - Id of the Master to make the default

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 647. Method: mapToLocal

HTTP **POST**

Description:

Add a single organizations to the list of those the specified Master has exported to this Slave

Parameters:

- string sessionKey
- int masterId - ID of the desired Master
- int masterOrgId - ID of the desired Master
- int localOrgId - ID of the desired Master

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 648. Method: setCaCert

HTTP **POST**

Description:

Set the CA-CERT filename for specified Master on this Slave

Parameters:

- string sessionKey
- int masterId - ID of the Master to affect
- string caCertFilename - path to specified Master's CA cert

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 649. Method: setMasterOrgs

HTTP **POST**

Description:

Reset all organizations the specified Master has exported to this Slave

Parameters:

- string sessionKey
- int masterId - Id of the desired Master
- array orgMaps
 - struct master-org details
 - int "masterOrgId"
 - string "masterOrgName"
 - int "localOrgId"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 650. Method: unsetDefaultMaster

HTTP **POST**

Description:

Make this slave have no default Master for inter-server-sync

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 651. Method: update

HTTP **POST**

Description:

Updates the label of the specified Master

Parameters:

- string sessionKey
- int masterId - ID of the Master to update
- string label - Desired new label

Returns:

- * struct IssMaster info
 - int "id"
 - string "label"
 - string "caCert"
 - boolean "isCurrentMaster"

sync.slave

Chapter 652. Available methods

- create
- delete
- getAllowedOrgs
- getSlave
- getSlaveByName
- getSlaves
- setAllowedOrgs
- update

Chapter 653. Description

Contains methods to set up information about allowed-"slaves", for use on the "master" side of ISS

Namespace:

sync.slave

Chapter 654. Method: create

HTTP **POST**

Description:

Create a new Slave, known to this Master.

Parameters:

- string sessionKey
- string slaveFqdn - Slave's fully-qualified domain name
- boolean isEnabled - Let this slave talk to us?
- boolean allowAllOrgs - Export all our orgs to this slave?

Returns:

- * struct IssSlave info
 - int "id"
 - string "slave"
 - boolean "enabled"
 - boolean "allowAllOrgs"

Chapter 655. Method: delete

HTTP **POST**

Description:

Remove the specified Slave

Parameters:

- string sessionKey
- int slaveId - ID of the Slave to remove

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 656. Method: getAllowedOrgs

HTTP **GET**

Description:

Get all orgs this Master is willing to export to the specified Slave

Parameters:

- string sessionKey
- int slaveId - Id of the desired Slave

Returns:

- int array ids of allowed organizations

Chapter 657. Method: getSlave

HTTP **GET**

Description:

Find a Slave by specifying its ID

Parameters:

- string sessionKey
- int slaveId - ID of the desired Slave

Returns:

- * struct IssSlave info
 - int "id"
 - string "slave"
 - boolean "enabled"
 - boolean "allowAllOrgs"

Chapter 658. Method: getSlaveByName

HTTP **GET**

Description:

Find a Slave by specifying its Fully-Qualified Domain Name

Parameters:

- string sessionKey
- string slaveFqdn - Domain-name of the desired Slave

Returns:

- * struct IssSlave info
 - int "id"
 - string "slave"
 - boolean "enabled"
 - boolean "allowAllOrgs"

Chapter 659. Method: getSlaves

HTTP **GET**

Description:

Get all the Slaves this Master knows about

Parameters:

- string sessionKey

Returns:

- array :
- struct IssSlave info
 - int "id"
 - string "slave"
 - boolean "enabled"
 - boolean "allowAllOrgs"

Chapter 660. Method: setAllowedOrgs

HTTP **POST**

Description:

Set the orgs this Master is willing to export to the specified Slave

Parameters:

- string sessionKey
- int slaveId - ID of the desired Slave
- int array orgIds - List of org-ids we're willing to export

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 661. Method: update

HTTP **POST**

Description:

Updates attributes of the specified Slave

Parameters:

- string sessionKey
- int slaveId - ID of the Slave to update
- string slaveFqdn - Slave's fully-qualified domain name
- boolean isEnabled - Let this slave talk to us?
- boolean allowAllOrgs - Export all our orgs to this Slave?

Returns:

- * struct IssSlave info
 - int "id"
 - string "slave"
 - boolean "enabled"
 - boolean "allowAllOrgs"

system

Chapter 662. Available methods

- addEntitlements
- addNote
- bootstrap
- bootstrap
- bootstrap
- bootstrap
- bootstrapWithPrivateSshKey
- bootstrapWithPrivateSshKey
- bootstrapWithPrivateSshKey
- bootstrapWithPrivateSshKey
- changeProxy
- comparePackageProfile
- comparePackages
- createPackageProfile
- createSystemProfile
- createSystemRecord
- createSystemRecord
- deleteCustomValues
- deleteGuestProfiles
- deleteNote
- deleteNotes
- deletePackageProfile
- deleteSystem
- deleteSystem
- deleteSystem

-
- deleteSystems
 - deleteSystems
 - deleteTagFromSnapshot
 - downloadSystemId
 - getCoCoAttestationConfig
 - getCoCoAttestationResultDetails
 - getConnectionPath
 - getCpu
 - getCustomValues
 - getDetails
 - getDevices
 - getDmi
 - getEntitlements
 - getEventDetails
 - getEventHistory
 - getEventHistory
 - getEventHistory
 - getEventHistory
 - getEventHistory
 - getId
 - getInstalledProducts
 - getKernelLivePatch
 - getLatestCoCoAttestationReport
 - getMemory
 - getMinionIdMap
 - getName
 - getNetwork
 - getNetworkDevices

-
- [getNetworkForSystems](#)
 - [getOsaPing](#)
 - [getPillar](#)
 - [getPillar](#)
 - [getRegistrationDate](#)
 - [getRelevantErrata](#)
 - [getRelevantErrata](#)
 - [getRelevantErrataByType](#)
 - [getRunningKernel](#)
 - [getScriptActionDetails](#)
 - [getScriptResults](#)
 - [getSubscribedBaseChannel](#)
 - [getSystemCurrencyMultipliers](#)
 - [getSystemCurrencyScores](#)
 - [getUnscheduledErrata](#)
 - [getUuid](#)
 - [getVariables](#)
 - [isNvreInstalled](#)
 - [isNvreInstalled](#)
 - [listActivationKeys](#)
 - [listActiveSystems](#)
 - [listActiveSystemsDetails](#)
 - [listAdministrators](#)
 - [listAllInstallablePackages](#)
 - [listCoCoAttestationReports](#)
 - [listCoCoAttestationReports](#)
 - [listCoCoAttestationReports](#)

-
- `listDuplicatesByHostname`
 - `listDuplicatesByIp`
 - `listDuplicatesByMac`
 - `listEmptySystemProfiles`
 - `listExtraPackages`
 - `listFqdns`
 - `listGroups`
 - `listInactiveSystems`
 - `listInactiveSystems`
 - `listInstalledPackages`
 - `listLatestAvailablePackage`
 - `listLatestInstallablePackages`
 - `listLatestUpgradablePackages`
 - `listMigrationTargets`
 - `listMigrationTargets`
 - `listNewerInstalledPackages`
 - `listNotes`
 - `listOlderInstalledPackages`
 - `listOutOfDateSystems`
 - `listPackageProfiles`
 - `listPackageState`
 - `listPackages`
 - `listPackagesFromChannel`
 - `listPackagesLockStatus`
 - `listPhysicalSystems`
 - `listSubscribableBaseChannels`
 - `listSubscribableChildChannels`

-
- `listSubscribedChildChannels`
 - `listSuggestedReboot`
 - `listSystemEvents`
 - `listSystemEvents`
 - `listSystemEvents`
 - `listSystemEvents`
 - `listSystemGroupsForSystemsWithEntitlement`
 - `listSystems`
 - `listSystemsWithEntitlement`
 - `listSystemsWithExtraPackages`
 - `listSystemsWithPackage`
 - `listSystemsWithPackage`
 - `listUngroupedSystems`
 - `listUserSystems`
 - `listUserSystems`
 - `listVirtualGuests`
 - `listVirtualHosts`
 - `obtainReactivationKey`
 - `obtainReactivationKey`
 - `provisionSystem`
 - `provisionSystem`
 - `provisionSystem`
 - `provisionSystem`
 - `provisionVirtualGuest`
 - `provisionVirtualGuest`
 - `provisionVirtualGuest`
 - `refreshPillar`

-
- refreshPillar
 - registerPeripheralServer
 - removeEntitlements
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyErrata
 - scheduleApplyHighstate
 - scheduleApplyHighstate
 - scheduleApplyStates
 - scheduleApplyStates
 - scheduleCertificateUpdate
 - scheduleCertificateUpdate
 - scheduleChangeChannels
 - scheduleChangeChannels
 - scheduleCoCoAttestation
 - scheduleDistUpgrade
 - scheduleDistUpgrade
 - scheduleGuestAction
 - scheduleGuestAction
 - scheduleHardwareRefresh

-
- `schedulePackageInstall`
 - `schedulePackageInstall`
 - `schedulePackageInstall`
 - `schedulePackageInstall`
 - `schedulePackageInstallByNevra`
 - `schedulePackageInstallByNevra`
 - `schedulePackageInstallByNevra`
 - `schedulePackageInstallByNevra`
 - `schedulePackageInstallByNevra`
 - `schedulePackageLockChange`
 - `schedulePackageRefresh`
 - `schedulePackageRemove`
 - `schedulePackageRemove`
 - `schedulePackageRemove`
 - `schedulePackageRemove`
 - `schedulePackageRemoveByNevra`
 - `schedulePackageRemoveByNevra`
 - `schedulePackageRemoveByNevra`
 - `schedulePackageRemoveByNevra`
 - `schedulePackageUpdate`
 - `scheduleProductMigration`
 - `scheduleProductMigration`
 - `scheduleProductMigration`
 - `scheduleProductMigration`
 - `scheduleProductMigration`
 - `scheduleReboot`
 - `scheduleSPMigration`
 - `scheduleSPMigration`

-
- `scheduleSPMigration`
 - `scheduleSPMigration`
 - `scheduleScriptRun`
 - `scheduleScriptRun`
 - `scheduleScriptRun`
 - `scheduleScriptRun`
 - `scheduleSyncPackagesWithSystem`
 - `searchByName`
 - `sendOsaPing`
 - `setBaseChannel`
 - `setBaseChannel`
 - `setChildChannels`
 - `setCoCoAttestationConfig`
 - `setCustomValues`
 - `setDetails`
 - `setGroupMembership`
 - `setGuestCpus`
 - `setGuestMemory`
 - `setLockStatus`
 - `setPillar`
 - `setPillar`
 - `setPrimaryFqdn`
 - `setPrimaryInterface`
 - `setProfileName`
 - `setVariables`
 - `tagLatestSnapshot`
 - `unentitle`

-
- `updatePackageState`
 - `updatePeripheralServerInfo`
 - `upgradeEntitlement`
 - `whoRegistered`

Chapter 663. Description

Provides methods to access and modify registered system.

Namespace:

system

Chapter 664. Method: addEntitlements

HTTP **POST**

Description:

Add entitlements to a server. Entitlements a server already has are quietly ignored.

Parameters:

- string sessionKey
- int sid
- string array entitlements

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 665. Method: addNote

HTTP **POST**

Description:

Add a new note to the given server.

Parameters:

- string sessionKey
- int sid
- string subject - What the note is about.
- string body - Content of the note.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 666. Method: bootstrap

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPassword - SSH password of given user
- string activationKey - Activation key
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 667. Method: bootstrap

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPassword - SSH password of given user
- string activationKey - Activation key
- int proxyId - System ID of proxy to use
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 668. Method: bootstrap

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPassword - SSH password of given user
- string activationKey - Activation key
- string reactivationKey - Reactivation key
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 669. Method: bootstrap

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPassword - SSH password of given user
- string activationKey - Activation key
- string reactivationKey - Reactivation key
- int proxyId - System ID of proxy to use
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 670. Method: bootstrapWithPrivateSshKey

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH. Use SSH private key for authentication.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPrivKey - SSH private key as a string in PEM format
- string sshPrivKeyPass - SSH passphrase for the key (use empty string for no passphrase)
- string activationKey - Activation key
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 671. Method: bootstrapWithPrivateSshKey

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH. Use SSH private key for authentication.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPrivKey - SSH private key as a string in PEM format
- string sshPrivKeyPass - SSH passphrase for the key (use empty string for no passphrase)
- string activationKey - Activation key
- int proxyId - System ID of proxy to use
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 672. Method: bootstrapWithPrivateSshKey

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH. Use SSH private key for authentication.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPrivKey - SSH private key as a string in PEM format
- string sshPrivKeyPass - SSH passphrase for the key (use empty string for no passphrase)
- string activationKey - Activation key
- string reactivationKey - Reactivation key
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 673. Method: bootstrapWithPrivateSshKey

HTTP **POST**

Description:

Bootstrap a system for management via either Salt or Salt SSH. Use SSH private key for authentication.

Parameters:

- string sessionKey
- string host - Hostname or IP address of target
- int sshPort - SSH port on target machine
- string sshUser - SSH user on target machine
- string sshPrivKey - SSH private key as a string in PEM format
- string sshPrivKeyPass - SSH passphrase for the key (use empty string for no passphrase)
- string activationKey - Activation key
- string reactivationKey - Reactivation key
- int proxyId - System ID of proxy to use
- boolean saltSSH - Manage system with Salt SSH

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 674. Method: changeProxy

HTTP **POST**

Description:

Connect given systems to another proxy.

Parameters:

- string sessionKey
- int array sids
- int proxyId

Returns:

- int array actionIds

Chapter 675. Method: comparePackageProfile

HTTP **POST**

Description:

Compare a system's packages against a package profile. In the result returned, 'this_system' represents the server provided as an input and 'other_system' represents the profile provided as an input.

Parameters:

- string sessionKey
- int sid
- string profileLabel

Returns:

- array :
- struct package metadata
 - int "package_name_id"
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
 - string "this_system" - version of package on this system
 - string "other_system" - version of package on the other system
 - int "comparison"
 - 0 - no difference
 - 1 - package on this system only
 - 2 - newer package version on this system
 - 3 - package on other system only

-
- 4 - newer package version on other system

Chapter 676. Method: comparePackages

HTTP **POST**

Description:

Compares the packages installed on two systems.

Parameters:

- string sessionKey
- int sid1
- int sid2

Returns:

- array :
- struct package metadata
 - int "package_name_id"
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
 - string "this_system" - version of package on this system
 - string "other_system" - version of package on the other system
 - int "comparison"
 - 0 - no difference
 - 1 - package on this system only
 - 2 - newer package version on this system
 - 3 - package on other system only
 - 4 - newer package version on other system

Chapter 677. Method: createPackageProfile

HTTP **POST**

Description:

Create a new stored Package Profile from a systems installed package list.

Parameters:

- string sessionKey
- int sid
- string profileLabel
- string description

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 678. Method: createSystemProfile

HTTP **POST**

Description:

Creates a system record in database for a system that is not registered. Either "hwAddress" or "hostname" prop must be specified in the "data" struct. If a system(s) matching given data exists, a SystemsExistFaultException is thrown which contains matching system IDs in its message.

Parameters:

- string sessionKey
- string systemName - System name
- struct data
 - string "hwAddress" - The HW address of the network interface (MAC)
 - string "hostname" - The hostname of the profile

Returns:

- int systemId - The id of the created system

Chapter 679. Method: createSystemRecord

HTTP **POST**

Description:

Creates a cobbler system record with the specified kickstart label

Parameters:

- string sessionKey
- int sid
- string ksLabel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 680. Method: createSystemRecord

HTTP **POST**

Description:

Creates a cobbler system record for a system that is not registered.

Parameters:

- string sessionKey
- string systemName
- string ksLabel
- string kOptions
- string comment
- array netDevices
 - struct network device
 - string "name"
 - string "mac"
 - string "ip"
 - string "dnsname"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 681. Method: deleteCustomValues

HTTP **POST**

Description:

Delete the custom values defined for the custom system information keys provided from the given system. (Note: Attempt to delete values of non-existing keys throws exception. Attempt to delete value of existing key which has assigned no values doesn't throw exception.)

Parameters:

- string sessionKey
- int sid
- string array keys

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 682. Method: deleteGuestProfiles

HTTP **POST**

Description:

Delete the specified list of guest profiles for a given host

Parameters:

- string sessionKey
- int hostId
- string array guestNames

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 683. Method: deleteNote

HTTP **POST**

Description:

Deletes the given note from the server.

Parameters:

- string sessionKey
- int sid
- int noteld

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 684. Method: deleteNotes

HTTP **POST**

Description:

Deletes all notes from the server.

Parameters:

- string sessionKey
- int sid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 685. Method: deletePackageProfile

HTTP **POST**

Description:

Delete a package profile

Parameters:

- string sessionKey
- int profileId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 686. Method: deleteSystem

HTTP **POST**

Description:

Delete a system given its client certificate.

Parameters:

- string clientCert - client certificate of the system

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.10

Chapter 687. Method: deleteSystem

HTTP **POST**

Description:

Delete a system given its server id synchronously without cleanup

Parameters:

- string sessionKey
- int sid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 688. Method: deleteSystem

HTTP **POST**

Description:

Delete a system given its server id synchronously

Parameters:

- string sessionKey
- int sid
- string cleanupType - Possible values: 'FAIL_ON_CLEANUP_ERR' - fail in case of cleanup error, 'NO_CLEANUP' - do not cleanup, just delete, 'FORCE_DELETE' - Try cleanup first but delete server anyway in case of error

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 689. Method: deleteSystems

HTTP **POST**

Description:

Delete systems given a list of system ids asynchronously.

Parameters:

- string sessionKey
- int array sids

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 690. Method: deleteSystems

HTTP **POST**

Description:

Delete systems given a list of system ids asynchronously.

Parameters:

- string sessionKey
- int array sids
- string cleanupType - Possible values: 'FAIL_ON_CLEANUP_ERR' - fail in case of cleanup error, 'NO_CLEANUP' - do not cleanup, just delete, 'FORCE_DELETE' - Try cleanup first but delete server anyway in case of error

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 691. Method: deleteTagFromSnapshot

HTTP **POST**

Description:

Deletes tag from system snapshot

Parameters:

- string sessionKey
- int sid
- string tagName

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 692. Method: downloadSystemId

HTTP **POST**

Description:

Get the system ID file for a given server.

Parameters:

- string sessionKey
- int sid

Returns:

- string id

Chapter 693. Method: getCoCoAttestationConfig

HTTP **GET**

Description:

Return the Confidential Compute Attestation configuration for the given system id

Parameters:

- string sessionKey
- int sid - ID of the server to get the config for.

Returns:

- * struct coco_attestation_config
 - boolean "enabled" - true if Confidential Compute Attestation is enabled for this system
 - string "environment_type" - the configured environment type
 - int "system_id" - the ID of the system

Chapter 694. Method: getCoCoAttestationResultDetails

HTTP **GET**

Description:

Return a specific results with all details

Parameters:

- string sessionKey
- int sid
- int resultId

Returns:

- * struct result
 - int "result_id"
 - string "result_type"
 - string "result_status"
 - string "result_description"
 - dateTime.iso8601 "result_attested"
 - string "result_details"

Chapter 695. Method: getConnectionPath

HTTP **GET**

Description:

Get the list of proxies that the given system connects through in order to reach the server.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct proxy connection path details
 - int "position" - position of proxy in chain. The proxy that the system connects directly to is listed in position 1.
 - int "id" - proxy system ID
 - string "hostname" - proxy host name

Chapter 696. Method: getCpu

HTTP **GET**

Description:

Gets the CPU information of a system.

Parameters:

- string sessionKey
- int sid

Returns:

- * struct CPU
 - string "cache"
 - string "family"
 - string "mhz"
 - string "flags"
 - string "model"
 - string "vendor"
 - string "arch"
 - string "stepping"
 - string "count"
 - int "socket_count (if available)"
 - int "core_count (if available) number of cores per socket"
 - int "thread_count (if available) number of threads per core"

Chapter 697. Method: getCustomValues

HTTP **GET**

Description:

Get the custom data values defined for the server.

Parameters:

- string sessionKey
- int sid

Returns:

- struct custom value
 - string "custom info label"

Chapter 698. Method: getDetails

HTTP **GET**

Description:

Get system details.

Parameters:

- string sessionKey
- int sid

Returns:

- * struct server details
 - int "id" - system ID
 - string "profile_name"
 - string "machine_id"
 - boolean "payg" - Whether the server instance is payg or not
 - string "minion_id"
 - string "base_entitlement" - system's base entitlement label
 - string array "addon_entitlements" - system's addon entitlements labels, currently only 'virtualization_host'
 - boolean "auto_update" - true if system has auto errata updates enabled
 - string "release" - the operating system release (i.e. 4AS, 5Server)
 - string "address1"
 - string "address2"
 - string "city"
 - string "state"
 - string "country"
 - string "building"
 - string "room"

-
- string "rack"
 - string "description"
 - string "hostname"
 - dateTime.iso8601 "last_boot"
 - string "osa_status" - either 'unknown', 'offline', or 'online'
 - boolean "lock_status" - True indicates that the system is locked. False indicates that the system is unlocked.
 - string "virtualization" - virtualization type - for virtual guests only (optional)
 - string "contact_method" - one of the following:
 - default
 - ssh-push
 - ssh-push-tunnel

Chapter 699. Method: getDevices

HTTP **GET**

Description:

Gets a list of devices for a system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct device
 - string "device" - optional
 - string "device_class" - Includes CDROM, FIREWIRE, HD, USB, VIDEO, OTHER, etc.
 - string "driver"
 - string "description"
 - string "bus"
 - string "pcitype"

Chapter 700. Method: getDmi

HTTP **GET**

Description:

Gets the DMI information of a system.

Parameters:

- string sessionKey
- int sid

Returns:

- * struct DMI
 - string "vendor"
 - string "system"
 - string "product"
 - string "asset"
 - string "board"
 - string "bios_release" - (optional)
 - string "bios_vendor" - (optional)
 - string "bios_version" - (optional)

Chapter 701. Method: getEntitlements

HTTP **GET**

Description:

Gets the entitlements for a given server.

Parameters:

- string sessionKey
- int sid

Returns:

- string array entitlement_label

Chapter 702. Method: `getEventDetails`

HTTP `GET`

Description:

Returns the details of the event associated with the specified server and event. The event id must be a value returned by the `system.getEventHistory` API.

Parameters:

- string `sessionKey`
- int `sid`
- int `eid` - ID of the event

Returns:

- array :
- struct system event
 - int "id" - ID of the event
 - string "history_type" - type of history event
 - string "status" - status of the event
 - string "summary" - summary of the event
 - `dateTime.iso8601` "created" - date that the event was created
 - `dateTime.iso8601` "picked_up" - date that the event was picked up
 - `dateTime.iso8601` "completed" - date that the event occurred
 - `dateTime.iso8601` "earliest_action" - earliest date this action could occur
 - string "result_msg" - the result string of the action executed on the client machine (optional)
 - int "result_code" - the result code of the action executed on the client machine (optional)
 - array "additional_info" - additional information for the event, if available
 - struct info
 - string "detail" - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be

the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.

- string "result" - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Chapter 703. Method: `getEventHistory` (Deprecated)

HTTP `GET`

Description:

Returns a list history items associated with the system, ordered from newest to oldest. Note that the details may be empty for events that were scheduled against the system (as compared to instant). For more information on such events, see the `system.listSystemEvents` operation. Note: This version of the method is deprecated and the return value will be changed in a future API version. Please use one of the other overloaded versions of `getEventHistory`.

Deprecated - This version of the method is deprecated and the return value will be changed in a future API version. Please use one of the other overloaded versions of `getEventHistory`.

Parameters:

- string `sessionKey`
- int `sid`

Returns:

- array :
- struct history event
 - `dateTime.iso8601 "completed"` - date that the event occurred (optional)
 - string `"summary"` - summary of the event
 - string `"details"` - details of the event

Chapter 704. Method: getEventHistory

HTTP **GET**

Description:

Returns a list of history items associated with the system happened after the specified date. The list is paged and ordered from newest to oldest.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestDate
- int offset - Number of results to skip
- int limit - Maximum number of results

Returns:

- array :
- struct system event
 - int "id" - ID of the event
 - string "history_type" - type of history event
 - string "status" - status of the event
 - string "summary" - summary of the event
 - dateTime.iso8601 "completed" - date that the event occurred

Chapter 705. Method: getEventHistory

HTTP **GET**

Description:

Returns a list of history items associated with the system. The list is paged and ordered from newest to oldest.

Parameters:

- string sessionKey
- int sid
- int offset - Number of results to skip
- int limit - Maximum number of results

Returns:

- array :
- struct system event
 - int "id" - ID of the event
 - string "history_type" - type of history event
 - string "status" - status of the event
 - string "summary" - summary of the event
 - dateTime.iso8601 "completed" - date that the event occurred

Chapter 706. Method: getEventHistory

HTTP **GET**

Description:

Returns a list of history items associated with the system happened after the specified date. The list is ordered from newest to oldest.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestDate

Returns:

- array :
- struct system event
 - int "id" - ID of the event
 - string "history_type" - type of history event
 - string "status" - status of the event
 - string "summary" - summary of the event
 - dateTime.iso8601 "completed" - date that the event occurred

Chapter 707. Method: getId

HTTP **GET**

Description:

Get system IDs and last check in information for the given system name.

Parameters:

- string sessionKey
- string name

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 708. Method: getInstalledProducts

HTTP **GET**

Description:

Get a list of installed products for given system

Parameters:

- User loggedInUser
- int sid

Returns:

- array :
- struct installed product
 - string "name"
 - boolean "isBaseProduct"
 - string "version" - returned only if applies
 - string "arch" - returned only if applies
 - string "release" - returned only if applies
 - string "friendlyName" - returned only if available

Chapter 709. Method: getKernelLivePatch

HTTP **GET**

Description:

Returns the currently active kernel live patching version relative to the running kernel version of the system, or empty string if live patching feature is not in use for the given system.

Parameters:

- string sessionKey
- int sid

Returns:

- string

Chapter 710. Method: getLatestCoCoAttestationReport

HTTP **GET**

Description:

Return the latest report for the given system

Parameters:

- string sessionKey
- int sid

Returns:

- * struct reportResults
 - int "report_id"
 - string "report_status"
 - dateTime.iso8601 "report_created"
 - dateTime.iso8601 "report_modified"
 - array "results"
 - struct result
 - int "result_id"
 - string "result_type"
 - string "result_status"
 - string "result_description"
 - dateTime.iso8601 "result_attested"

Chapter 711. Method: getMemory

HTTP **GET**

Description:

Gets the memory information for a system.

Parameters:

- string sessionKey
- int sid

Returns:

- struct memory
 - int "ram" - The amount of physical memory in MB.
 - int "swap" - The amount of swap space in MB.

Chapter 712. Method: getMinionIdMap

HTTP **GET**

Description:

Return a map from Salt minion IDs to System IDs. Map entries are limited to systems that are visible by the current user.

Parameters:

- string sessionKey

Returns:

- map id_map - minion IDs to system IDs

Chapter 713. Method: getName

HTTP **GET**

Description:

Get system name and last check in information for the given system ID.

Parameters:

- string sessionKey
- string sid

Returns:

- struct name info
 - int "id" - Server id
 - string "name" - Server name
 - dateTime.iso8601 "last_checkin" - Last time server successfully checked in

Chapter 714. Method: getNetwork

HTTP **GET**

Description:

Get the addresses and hostname for a given server.

Parameters:

- string sessionKey
- int sid

Returns:

- struct network info
 - string "ip" - IPv4 address of server
 - string "ip6" - IPv6 address of server
 - string "hostname" - Hostname of server

Chapter 715. Method: getNetworkDevices

HTTP **GET**

Description:

Returns the network devices for the given server.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct network device
 - string "ip" - IP address assigned to this network device
 - string "interface" - network interface assigned to device, e.g. eth0
 - string "netmask" - network mask assigned to device
 - string "hardware_address" - hardware address of device
 - string "module" - network driver used for this device
 - string "broadcast" - broadcast address for device
 - array "ipv6" - the list of IPv6 addresses
 - struct ipv6 address
 - string "address" - IPv6 address of this network device
 - string "netmask" - IPv6 netmask of this network device
 - string "scope" - IPv6 address scope
 - array "ipv4" - the list of IPv4 addresses
 - struct ipv4 address
 - string "address" - IPv4 address of this network device
 - string "netmask" - IPv4 netmask of this network device
 - string "broadcast" - IPv4 broadcast address of this network device

Chapter 716. Method: getNetworkForSystems

HTTP **GET**

Description:

Get the addresses and hostname for a given list of systems.

Parameters:

- string sessionKey
- int array sids

Returns:

- array :
 - struct network info
 - int "system_id" - ID of the system
 - string "ip" - IPv4 address of system
 - string "ip6" - IPv6 address of system
 - string "hostname" - Hostname of system
 - string "primary_fqdn" - Primary FQDN of system

Chapter 717. Method: getOsaPing

HTTP **GET**

Description:

get details about a ping sent to a system using OSA

Parameters:

- User loggedInUser
- int sid

Returns:

- struct osaPing
 - string "state" - state of the system (unknown, online, offline)
 - dateTime.iso8601 "lastMessageTime" - time of the last received response (1970/01/01 00:00:00 if never received a response)
 - dateTime.iso8601 "lastPingTime" - time of the last sent ping (1970/01/01 00:00:00 if no ping is pending)

Chapter 718. Method: getPillar

HTTP **GET**

Description:

Get pillar data of given category for given system

Parameters:

- string sessionKey
- int systemId
- string category

Returns:

- struct the pillar data

Chapter 719. Method: getPillar

HTTP **GET**

Description:

Get pillar data of given category for given system

Parameters:

- string sessionKey
- int minionId
- string category

Returns:

- struct the pillar data

Chapter 720. Method: getRegistrationDate

HTTP **GET**

Description:

Returns the date the system was registered.

Parameters:

- string sessionKey
- int sid

Returns:

- dateTime.iso8601 date - The date the system was registered, in local time

Chapter 721. Method: getRelevantErrata

HTTP **GET**

Description:

Returns a list of all errata that are relevant to the system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 722. Method: getRelevantErrata

HTTP **GET**

Description:

Returns a list of all errata that are relevant to a list of systems.

Parameters:

- string `sessionKey`
- int array `sids`

Returns:

- array :
 - struct `server_errata`
 - string `"system_id"` - The ID of the system
 - array `"errata"` - An array of available errata infos
- struct `errata`
 - int `"id"` - errata ID
 - string `"issue_date"` - the date erratum was updated (deprecated)
 - string `"date"` - the date erratum was created (deprecated)
 - string `"update_date"` - the date erratum was updated (deprecated)
 - string `"advisory_synopsis"` - summary of the erratum
 - string `"advisory_type"` - type label such as 'Security', 'Bug Fix'
 - string `"advisory_status"` - status label such as 'final', 'testing', 'retracted'
 - string `"advisory_name"` - name such as 'RHSA', etc.
 - boolean `"reboot_suggested"` - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean `"restart_suggested"` - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 723. Method: getRelevantErrataByType

HTTP **GET**

Description:

Returns a list of all errata of the specified type that are relevant to the system.

Parameters:

- string `sessionKey`
- int `sid`
- string `advisoryType` - type of advisory (one of the following: 'Security Advisory', 'Product Enhancement Advisory', 'Bug Fix Advisory')

Returns:

- array :
- struct `errata`
 - int "id" - errata ID
 - string "issue_date" - the date erratum was updated (deprecated)
 - string "date" - the date erratum was created (deprecated)
 - string "update_date" - the date erratum was updated (deprecated)
 - string "advisory_synopsis" - summary of the erratum
 - string "advisory_type" - type label such as 'Security', 'Bug Fix'
 - string "advisory_status" - status label such as 'final', 'testing', 'retracted'
 - string "advisory_name" - name such as 'RHSA', etc.
 - boolean "reboot_suggested" - A boolean flag signaling whether a system reboot is advisable following the application of the errata. Typical example is upon kernel update.
 - boolean "restart_suggested" - A boolean flag signaling a weather reboot of the package manager is advisable following the application of the errata. This is commonly used to address update stack issues before proceeding with other updates.

Chapter 724. Method: getRunningKernel

HTTP **GET**

Description:

Returns the running kernel of the given system.

Parameters:

- string sessionKey
- int sid

Returns:

- string kernel

Chapter 725. Method: getScriptActionDetails

HTTP **GET**

Description:

Returns script details for script run actions

Parameters:

- string `sessionKey`
- int `actionId` - ID of the script run action.

Returns:

- struct Script details
 - int "id" - action id
 - string "content" - script content
 - string "run_as_user" - Run as user
 - string "run_as_group" - Run as group
 - int "timeout" - Timeout in seconds
 - array :
- struct script result
 - int "serverId" - ID of the server the script runs on
 - dateTime.iso8601 "startDate" - time script began execution
 - dateTime.iso8601 "stopDate" - time script stopped execution
 - int "returnCode" - script execution return code
 - string "output" - output of the script (base64 encoded according to the `output_enc64` attribute)
 - boolean "output_enc64" - identifies base64 encoded output

Chapter 726. Method: getScriptResults

HTTP **POST**

Description:

Fetch results from a script execution. Returns an empty array if no results are yet available.

Parameters:

- string `sessionKey`
- int `actionId` - ID of the script run action.

Returns:

- array :
- struct script result
 - int "serverId" - ID of the server the script runs on
 - dateTime.iso8601 "startDate" - time script began execution
 - dateTime.iso8601 "stopDate" - time script stopped execution
 - int "returnCode" - script execution return code
 - string "output" - output of the script (base64 encoded according to the `output_enc64` attribute)
 - boolean "output_enc64" - identifies base64 encoded output

Chapter 727. Method: getSubscribedBaseChannel

HTTP **GET**

Description:

Provides the base channel of a given system

Parameters:

- string sessionKey
- int sid

Returns:

- * struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"
 - string "gpg_key_fp"

-
- `dateTime.iso8601 "yumrepo_last_sync"` - (optional)
 - `string "end_of_life"`
 - `string "parent_channel_label"`
 - `string "clone_original"`
 - `array "contentSources"`
 - `struct content source`
 - `int "id"`
 - `string "label"`
 - `string "sourceUrl"`
 - `string "type"`

Chapter 728. Method: getSystemCurrencyMultipliers

HTTP **GET**

Description:

Get the System Currency score multipliers

Parameters:

- string sessionKey

Returns:

- map multipliers - Map of score multipliers

Chapter 729. Method: `getSystemCurrencyScores`

HTTP `GET`

Description:

Get the System Currency scores for all servers the user has access to

Parameters:

- string `sessionKey`

Returns:

- array :
 - struct system currency
 - int "sid"
 - int "critical security errata count"
 - int "important security errata count"
 - int "moderate security errata count"
 - int "low security errata count"
 - int "bug fix errata count"
 - int "enhancement errata count"
 - int "system currency score"

Chapter 730. Method: getUnscheduledErrata

HTTP **GET**

Description:

Provides an array of errata that are applicable to a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct errata
 - int "id" - errata ID
 - string "date" - the date erratum was created
 - string "advisory_type" - type of the advisory
 - string "advisory_status" - status of the advisory
 - string "advisory_name" - name of the advisory
 - string "advisory_synopsis" - summary of the erratum

Chapter 731. Method: getUuid

HTTP **GET**

Description:

Get the UUID from the given system ID.

Parameters:

- string sessionKey
- int sid

Returns:

- string uuid

Chapter 732. Method: getVariables

HTTP **GET**

Description:

Lists kickstart variables set in the system record for the specified server. Note: This call assumes that a system record exists in cobbler for the given system and will raise an XMLRPC fault if that is not the case. To create a system record over xmlrpc use `system.createSystemRecord`

To create a system record in the Web UI please go to System > <Specified System> > Provisioning > Select a Kickstart profile > Create Cobbler System Record.

Parameters:

- string sessionKey
- int sid

Returns:

- struct System kickstart variables
 - boolean "netboot" - netboot enabled
 - array "kickstart variables"
 - struct kickstart variable
 - string "key"
 - string or int "value"

Chapter 733. Method: isNvreInstalled

HTTP **GET**

Description:

Check if the package with the given NVRE is installed on given system.

Parameters:

- string sessionKey
- int sid
- string name - Package name.
- string version - Package version.
- string release - Package release.

Returns:

- int status - 1 if package exists, 0 if not, exception is thrown if an error occurs

Chapter 734. Method: isNvreInstalled

HTTP **GET**

Description:

Is the package with the given NVRE installed on given system.

Parameters:

- string sessionKey
- int sid
- string name - Package name.
- string version - Package version.
- string release - Package release.
- string epoch - Package epoch.

Returns:

- int status - 1 if package exists, 0 if not, exception is thrown if an error occurs

Chapter 735. Method: listActivationKeys

HTTP **GET**

Description:

List the activation keys the system was registered with. An empty list will be returned if an activation key was not used during registration.

Parameters:

- string sessionKey
- int sid

Returns:

- string array key

Chapter 736. Method: listActiveSystems

HTTP **GET**

Description:

Returns a list of active servers visible to the user.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 737. Method: listActiveSystemsDetails

HTTP **GET**

Description:

Given a list of server ids, returns a list of active servers' details visible to the user.

Parameters:

- string sessionKey
- int array sids

Returns:

- array :
 - struct server details
 - int "id" - The server's id
 - string "name" - The server's name
 - boolean "payg" - Whether the server instance is payg or not
 - string "minion_id" - The server's minion id, in case it is a salt minion client
 - dateTime.iso8601 "last_checkin" - Last time server successfully checked in (in UTC)
 - int "ram" - The amount of physical memory in MB.
 - int "swap" - The amount of swap space in MB.
 - struct "network_devices" - The server's network devices
- struct network device
 - string "ip" - IP address assigned to this network device
 - string "interface" - network interface assigned to device, e.g. eth0
 - string "netmask" - network mask assigned to device
 - string "hardware_address" - hardware address of device
 - string "module" - network driver used for this device
 - string "broadcast" - broadcast address for device
 - array "ipv6" - the list of IPv6 addresses

-
- struct ipv6 address
 - string "address" - IPv6 address of this network device
 - string "netmask" - IPv6 netmask of this network device
 - string "scope" - IPv6 address scope
 - array "ipv4" - the list of IPv4 addresses
 - struct ipv4 address
 - string "address" - IPv4 address of this network device
 - string "netmask" - IPv4 netmask of this network device
 - string "broadcast" - IPv4 broadcast address of this network device
 - struct "dmi_info" - The server's dmi info
 - struct DMI
 - string "vendor"
 - string "system"
 - string "product"
 - string "asset"
 - string "board"
 - string "bios_release" - (optional)
 - string "bios_vendor" - (optional)
 - string "bios_version" - (optional)
 - struct "cpu_info" - The server's cpu info
 - struct CPU
 - string "cache"
 - string "family"
 - string "mhz"
 - string "flags"
 - string "model"
 - string "vendor"
 - string "arch"
 - string "stepping"

-
- string "count"
 - int "socket_count (if available)"
 - int "core_count (if available) number of cores per socket"
 - int "thread_count (if available) number of threads per core"
 - array "subscribed_channels" - List of subscribed channels
 - array :
 - struct channel
 - int "channel_id" - The channel id.
 - string "channel_label" - The channel label.
 - array "active_guest_system_ids" - List of virtual guest system ids for active guests
 - array :
 - int "guest_id" - The guest's system id.

Chapter 738. Method: listAdministrators

HTTP **GET**

Description:

Returns a list of users which can administer the system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct user
 - int "id"
 - string "login"
 - string "login_uc" - upper case version of the login
 - boolean "enabled" - true if user is enabled, false if the user is disabled

Chapter 739. Method: listAllInstallablePackages

HTTP **GET**

Description:

Get the list of all installable packages for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"

Chapter 740. Method: listCoCoAttestationReports

HTTP **GET**

Description:

Return a list of reports with its results for the given filters

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliest

Returns:

- array :
- struct reportResults
 - int "report_id"
 - string "report_status"
 - dateTime.iso8601 "report_created"
 - dateTime.iso8601 "report_modified"
 - array "results"
 - struct result
 - int "result_id"
 - string "result_type"
 - string "result_status"
 - string "result_description"
 - dateTime.iso8601 "result_attested"

Chapter 741. Method: listCoCoAttestationReports

HTTP **GET**

Description:

Return a list of reports with its results for the given filters

Parameters:

- string sessionKey
- int sid
- int offset - Number of reports to skip
- int limit - Maximum number of reports

Returns:

- array :
- struct reportResults
 - int "report_id"
 - string "report_status"
 - dateTime.iso8601 "report_created"
 - dateTime.iso8601 "report_modified"
 - array "results"
 - struct result
 - int "result_id"
 - string "result_type"
 - string "result_status"
 - string "result_description"
 - dateTime.iso8601 "result_attested"

Chapter 742. Method: listCoCoAttestationReports

HTTP **GET**

Description:

Return a list of reports with its results for the given filters

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliest
- int offset - Number of reports to skip
- int limit - Maximum number of reports

Returns:

- array :
- struct reportResults
 - int "report_id"
 - string "report_status"
 - dateTime.iso8601 "report_created"
 - dateTime.iso8601 "report_modified"
 - array "results"
 - struct result
 - int "result_id"
 - string "result_type"
 - string "result_status"
 - string "result_description"
 - dateTime.iso8601 "result_attested"

Chapter 743. Method: listDuplicatesByHostname

HTTP **GET**

Description:

List duplicate systems by Hostname.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Duplicate Group
 - string "hostname"
 - array "systems"
- struct system
 - int "systemId"
 - string "systemName"
 - dateTime.iso8601 "last_checkin" - last time the server successfully checked in

Chapter 744. Method: listDuplicatesByIp

HTTP **GET**

Description:

List duplicate systems by IP Address.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Duplicate Group
 - string "ip"
 - array "systems"
- struct system
 - int "systemId"
 - string "systemName"
 - dateTime.iso8601 "last_checkin" - last time the server successfully checked in

Chapter 745. Method: listDuplicatesByMac

HTTP **GET**

Description:

List duplicate systems by Mac Address.

Parameters:

- string sessionKey

Returns:

- array :
 - struct Duplicate Group
 - string "mac"
 - array "systems"
- struct system
 - int "systemId"
 - string "systemName"
 - dateTime.iso8601 "last_checkin" - last time the server successfully checked in

Chapter 746. Method: listEmptySystemProfiles

HTTP **GET**

Description:

Returns a list of empty system profiles visible to user (created by the createSystemProfile method).

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "created" - Server creation time
 - string array "hw_address" - HW address

Chapter 747. Method: listExtraPackages

HTTP **GET**

Description:

List extra packages for a system

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch" - returned only if non-zero
 - string "arch"
 - date "installtime" - returned only if known

Chapter 748. Method: listFqdns

HTTP **GET**

Description:

Provides a list of FQDNs associated with a system.

Parameters:

- string sessionKey
- int sid

Returns:

- string array fqdn

Chapter 749. Method: listGroups

HTTP **GET**

Description:

List the available groups for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct system group
 - int "id" - server group id
 - int "subscribed" - 1 if the given server is subscribed to this server group, 0 otherwise
 - string "system_group_name" - Name of the server group
 - string "sgid" - server group id (Deprecated)

Chapter 750. Method: listInactiveSystems

HTTP **GET**

Description:

Lists systems that have been inactive for the default period of inactivity

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 751. Method: listInactiveSystems

HTTP **GET**

Description:

Lists systems that have been inactive for the specified number of days..

Parameters:

- string sessionKey
- int days

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 752. Method: listInstalledPackages

HTTP **GET**

Description:

List the installed packages for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - int "package_id" - PackageID, -1 if package is installed but not available in subscribed channels
 - string "name"
 - string "epoch"
 - string "version"
 - string "release"
 - string "arch" - architecture label
 - date "installtime" - returned only if known
 - boolean "retracted"

Chapter 753. Method: listLatestAvailablePackage

HTTP **GET**

Description:

Get the latest available version of a package for each system

Parameters:

- string sessionKey
- int array sids
- string packageName

Returns:

- array :
 - struct system
 - int "id" - server ID
 - string "name" - server name
 - struct "package" - package structure
 - struct package
 - int "id"
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - string "arch"

Chapter 754. Method: listLatestInstallablePackages

HTTP **GET**

Description:

Get the list of latest installable packages for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"

Chapter 755. Method: listLatestUpgradablePackages

HTTP **GET**

Description:

Get the list of latest upgradable packages for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - string "name"
 - string "arch"
 - string "from_version"
 - string "from_release"
 - string "from_epoch"
 - string "to_version"
 - string "to_release"
 - string "to_epoch"
 - string "to_package_id"

Chapter 756. Method: listMigrationTargets

HTTP **GET**

Description:

List possible migration targets for a system

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct migrationtarget
 - string "ident"
 - string "friendly"

Chapter 757. Method: listMigrationTargets

HTTP **POST**

Description:

List possible migration targets for a system, if `excludeTargetWhereMissingSuccessors` is false then valid targets without some successors will also be listed.

Parameters:

- string `sessionKey`
- int `sid`
- boolean `excludeTargetWhereMissingSuccessors`

Returns:

- array :
 - struct `migrationtarget`
 - string `"ident"`
 - string `"friendly"`

Chapter 758. Method: listNewerInstalledPackages

HTTP **GET**

Description:

Given a package name, version, release, and epoch, returns the list of packages installed on the system w/ the same name that are newer.

Parameters:

- string `sessionKey`
- int `sid`
- string `name` - Package name.
- string `version` - Package version.
- string `release` - Package release.
- string `epoch` - Package epoch.

Returns:

- array :
 - struct `package`
 - string `"name"`
 - string `"version"`
 - string `"release"`
 - string `"epoch"`

Chapter 759. Method: listNotes

HTTP **GET**

Description:

Provides a list of notes associated with a system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct note details
 - int "id"
 - string "subject" - subject of the note
 - string "note" - contents of the note
 - int "system_id" - the ID of the system associated with the note
 - string "creator" - creator of the note if exists (optional)
 - date "updated" - date of the last note update

Chapter 760. Method: listOlderInstalledPackages

HTTP **GET**

Description:

Given a package name, version, release, and epoch, returns the list of packages installed on the system with the same name that are older.

Parameters:

- string sessionKey
- int sid
- string name - Package name.
- string version - Package version.
- string release - Package release.
- string epoch - Package epoch.

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"

Chapter 761. Method: listOutOfDateSystems

HTTP **GET**

Description:

Returns list of systems needing package updates.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 762. Method: listPackageProfiles

HTTP **GET**

Description:

List the package profiles in this organization

Parameters:

- string sessionKey

Returns:

- array :
- struct package profile
 - int "id"
 - string "name"
 - string "channel"

Chapter 763. Method: listPackageState

HTTP **GET**

Description:

List possible migration targets for a system

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct package state
 - int "id"
 - string "name"
 - int "state_revision_id" - state revision ID
 - string "package_state_type_id" - 'INSTALLED' or 'REMOVED'
 - string "version_constraint_id" - 'LATEST' or 'ANY'

Chapter 764. Method: listPackages (Deprecated)

HTTP **GET**

Description:

List the installed packages for a given system. Usage of listInstalledPackages is preferred, as it returns architecture label (not name).

Deprecated – This is here for backwards compatibility: The method returns architecture name, whereas the other endpoints return/accept architecture label. Instead of this method, use listInstalledPackages preferably.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - string "arch" – Architecture name
 - date "installtime" – returned only if known

Chapter 765. Method: listPackagesFromChannel

HTTP **GET**

Description:

Provides a list of packages installed on a system that are also contained in the given channel. The installed package list did not include arch information before RHEL 5, so it is arch unaware. RHEL 5 systems do upload the arch information, and thus are arch aware.

Parameters:

- string sessionKey
- int sid
- string channelLabel

Returns:

- array :
- struct package
 - string "name"
 - string "version"
 - string "release"
 - string "epoch"
 - int "id"
 - string "arch_label"
 - dateTime.iso8601 "last_modified"
 - string "path" - the path on that file system that the package resides
 - boolean "part_of_retracted_patch" - true if the package is a part of a retracted patch
 - string "provider" - the provider of the package, determined by the gpg key it was signed with.

Chapter 766. Method: listPackagesLockStatus

HTTP **GET**

Description:

List current package locks status.

Parameters:

- string sessionKey
- string sid

Returns:

- array :
 - struct package
 - int "package_id" - PackageID, -1 if package is locked but not available in subscribed channels
 - string "name"
 - string "epoch"
 - string "version"
 - string "release"
 - string "arch" - architecture label
 - string "pending status" - return only if there is a pending locking

Chapter 767. Method: listPhysicalSystems

HTTP **GET**

Description:

Returns a list of all Physical servers visible to the user.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 768. Method: listSubscribableBaseChannels

HTTP **GET**

Description:

Returns a list of subscribable base channels.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct channel
 - int "id" - Base Channel ID.
 - string "name" - Name of channel.
 - string "label" - Label of Channel
 - int "current_base" - 1 indicates it is the current base channel

Chapter 769. Method: listSubscribableChildChannels

HTTP **GET**

Description:

Returns a list of subscribable child channels. This only shows channels the system is **not** currently subscribed to.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
 - struct child channel
 - int "id"
 - string "name"
 - string "label"
 - string "summary"
 - string "has_license"
 - string "gpg_key_url"

Chapter 770. Method: listSubscribedChildChannels

HTTP **GET**

Description:

Returns a list of subscribed child channels.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct channel
 - int "id"
 - string "name"
 - string "label"
 - string "arch_name"
 - string "arch_label"
 - string "summary"
 - string "description"
 - string "checksum_label"
 - dateTime.iso8601 "last_modified"
 - string "maintainer_name"
 - string "maintainer_email"
 - string "maintainer_phone"
 - string "support_policy"
 - string "gpg_key_url"
 - string "gpg_key_id"

-
- string "gpg_key_fp"
 - dateTime.iso8601 "yumrepo_last_sync" - (optional)
 - string "end_of_life"
 - string "parent_channel_label"
 - string "clone_original"
 - array "contentSources"
 - struct content source
 - int "id"
 - string "label"
 - string "sourceUrl"
 - string "type"

Chapter 771. Method: listSuggestedReboot

HTTP **GET**

Description:

List systems that require reboot.

Parameters:

- string sessionKey

Returns:

- array :
 - struct system
 - int "id"
 - string "name"

Chapter 772. Method: listSystemEvents

HTTP **GET**

Description:

List system actions of the specified type that were **scheduled** against the given server after the specified date. "actionType" should be exactly the string returned in the action_type field from the listSystemEvents(sessionKey, serverId) method. For example, 'Package Install' or 'Initiate a kickstart for a virtual guest.' Note: see also system.getEventHistory method which returns a history of all events.

Parameters:

- string sessionKey
- int sid - ID of system.
- string actionType - Type of the action.
- dateTime.iso8601 earliestDate

Returns:

- array :
 - struct action
 - int "failed_count" - Number of times action failed.
 - string "modified" - Date modified. (Deprecated by modified_date)
 - dateTime.iso8601 "modified_date" - Date modified.
 - string "created" - Date created. (Deprecated by created_date)
 - dateTime.iso8601 "created_date" - Date created.
 - string "action_type"
 - int "successful_count" - Number of times action was successful.
 - string "earliest_action" - Earliest date this action will occur.
 - int "archived" - If this action is archived. (1 or 0)
 - string "scheduler_user" - available only if concrete user has scheduled the action
 - string "prerequisite" - Pre-requisite action. (optional)

-
- string "name" - Name of this action.
 - int "id" - Id of this action.
 - string "version" - Version of action.
 - string "completion_time" - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by completed_date)
 - dateTime.iso8601 "completed_date" - The date/time the event was completed. (optional)
 - string "pickup_time" - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by pickup_date)
 - dateTime.iso8601 "pickup_date" - The date/time the action was picked up. (optional)
 - string "result_msg" - The result string after the action executes at the client machine. (optional)
 - array "additional_info" - This array contains additional information for the event, if available.
 - struct info
 - string "detail" - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
 - string "result" - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Available since API version: 10.8

Chapter 773. Method: listSystemEvents

HTTP **GET**

Description:

List all system actions that were **scheduled** against the given server. This may require the caller to filter the result to fetch actions with a specific action type or to use the overloaded `system.listSystemEvents` method with `actionType` as a parameter. Note: see also `system.getEventHistory` method which returns a history of all events.

Parameters:

- string `sessionKey`
- int `sid` - ID of system.

Returns:

- array :
 - struct `action`
 - int `"failed_count"` - Number of times action failed.
 - string `"modified"` - Date modified. (Deprecated by `modified_date`)
 - `dateTime.iso8601` `"modified_date"` - Date modified.
 - string `"created"` - Date created. (Deprecated by `created_date`)
 - `dateTime.iso8601` `"created_date"` - Date created.
 - string `"action_type"`
 - int `"successful_count"` - Number of times action was successful.
 - string `"earliest_action"` - Earliest date this action will occur.
 - int `"archived"` - If this action is archived. (1 or 0)
 - string `"scheduler_user"` - available only if concrete user has scheduled the action
 - string `"prerequisite"` - Pre-requisite action. (optional)
 - string `"name"` - Name of this action.
 - int `"id"` - Id of this action.
 - string `"version"` - Version of action.

-
- string "completion_time" - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by completed_date)
 - dateTime.iso8601 "completed_date" - The date/time the event was completed. (optional)
 - string "pickup_time" - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by pickup_date)
 - dateTime.iso8601 "pickup_date" - The date/time the action was picked up. (optional)
 - string "result_msg" - The result string after the action executes at the client machine. (optional)
 - array "additional_info" - This array contains additional information for the event, if available.
 - struct info
 - string "detail" - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
 - string "result" - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Available since API version: 10.8

Chapter 774. Method: listSystemEvents

HTTP **GET**

Description:

List system actions of the specified type that were **scheduled** against the given server. "actionType" should be exactly the string returned in the action_type field from the listSystemEvents(sessionKey, serverId) method. For example, 'Package Install' or 'Initiate a kickstart for a virtual guest.' Note: see also system.getEventHistory method which returns a history of all events.

Parameters:

- string sessionKey
- int sid - ID of system.
- string actionType - Type of the action.

Returns:

- array :
 - struct action
 - int "failed_count" - Number of times action failed.
 - string "modified" - Date modified. (Deprecated by modified_date)
 - dateTime.iso8601 "modified_date" - Date modified.
 - string "created" - Date created. (Deprecated by created_date)
 - dateTime.iso8601 "created_date" - Date created.
 - string "action_type"
 - int "successful_count" - Number of times action was successful.
 - string "earliest_action" - Earliest date this action will occur.
 - int "archived" - If this action is archived. (1 or 0)
 - string "scheduler_user" - available only if concrete user has scheduled the action
 - string "prerequisite" - Pre-requisite action. (optional)
 - string "name" - Name of this action.

-
- int "id" - Id of this action.
 - string "version" - Version of action.
 - string "completion_time" - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by completed_date)
 - dateTime.iso8601 "completed_date" - The date/time the event was completed. (optional)
 - string "pickup_time" - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by pickup_date)
 - dateTime.iso8601 "pickup_date" - The date/time the action was picked up. (optional)
 - string "result_msg" - The result string after the action executes at the client machine. (optional)
 - array "additional_info" - This array contains additional information for the event, if available.
 - struct info
 - string "detail" - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
 - string "result" - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Available since API version: 10.8

Chapter 775. Method: listSystemEvents

HTTP **GET**

Description:

List system actions of the specified type that were **scheduled** against the given server after the specified date. This may require the caller to filter the result to fetch actions with a specific action type or to use the overloaded `system.listSystemEvents` method with `actionType` as a parameter.

Note: see also `system.getEventHistory` method which returns a history of all events.

Parameters:

- string `sessionKey`
- int `sid` - ID of system.
- `dateTime.iso8601` `earliestDate`

Returns:

- array :
 - struct `action`
 - int `"failed_count"` - Number of times action failed.
 - string `"modified"` - Date modified. (Deprecated by `modified_date`)
 - `dateTime.iso8601` `"modified_date"` - Date modified.
 - string `"created"` - Date created. (Deprecated by `created_date`)
 - `dateTime.iso8601` `"created_date"` - Date created.
 - string `"action_type"`
 - int `"successful_count"` - Number of times action was successful.
 - string `"earliest_action"` - Earliest date this action will occur.
 - int `"archived"` - If this action is archived. (1 or 0)
 - string `"scheduler_user"` - available only if concrete user has scheduled the action
 - string `"prerequisite"` - Pre-requisite action. (optional)
 - string `"name"` - Name of this action.
 - int `"id"` - Id of this action.

-
- string "version" - Version of action.
 - string "completion_time" - The date/time the event was completed. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by completed_date)
 - dateTime.iso8601 "completed_date" - The date/time the event was completed. (optional)
 - string "pickup_time" - The date/time the action was picked up. Format ->YYYY-MM-dd hh:mm:ss.ms Eg ->2007-06-04 13:58:13.0. (optional) (Deprecated by pickup_date)
 - dateTime.iso8601 "pickup_date" - The date/time the action was picked up. (optional)
 - string "result_msg" - The result string after the action executes at the client machine. (optional)
 - array "additional_info" - This array contains additional information for the event, if available.
 - struct info
 - string "detail" - The detail provided depends on the specific event. For example, for a package event, this will be the package name, for an errata event, this will be the advisory name and synopsis, for a config file event, this will be path and optional revision information...etc.
 - string "result" - The result (if included) depends on the specific event. For example, for a package or errata event, no result is included, for a config file event, the result might include an error (if one occurred, such as the file was missing) or in the case of a config file comparison it might include the differences found.

Available since API version: 10.8

Chapter 776. Method: listSystemGroupsForSystemsWithEntitlement

HTTP **GET**

Description:

Returns the groups information a system is member of, for all the systems visible to the passed user and that are entitled with the passed entitlement.

Parameters:

- string sessionKey
- string entitlement

Returns:

- array :
- struct system
 - int "id" - system ID
 - array "system_groups"
 - struct system group
 - int "id" - system group ID
 - string "name" - system group name

Chapter 777. Method: listSystems

HTTP **GET**

Description:

Returns a list of all servers visible to the user.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 778. Method: listSystemsWithEntitlement

HTTP **GET**

Description:

Lists the systems that have the given entitlement

Parameters:

- string `sessionKey`
- string `entitlementName` - the entitlement name

Returns:

- array :
- struct `system`
 - int `"id"`
 - string `"name"`
 - `dateTime.iso8601` `"last_checkin"` - last time server successfully checked in
 - `dateTime.iso8601` `"created"` - server registration time
 - `dateTime.iso8601` `"last_boot"` - last server boot time
 - int `"extra_pkg_count"` - number of packages not belonging to any assigned channel
 - int `"outdated_pkg_count"` - number of out-of-date packages

Chapter 779. Method: listSystemsWithExtraPackages

HTTP **GET**

Description:

List systems with extra packages

Parameters:

- string sessionKey

Returns:

- array :
 - struct system
 - int "id" - System ID
 - string "name" - System profile name
 - int "extra_pkg_count" - Extra packages count

Chapter 780. Method: listSystemsWithPackage

HTTP **GET**

Description:

Lists the systems that have the given installed package

Parameters:

- string sessionKey
- int pid - the package id

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 781. Method: listSystemsWithPackage

HTTP **GET**

Description:

Lists the systems that have the given installed package

Parameters:

- string `sessionKey`
- string `name` - the package name
- string `version` - the package version
- string `release` - the package release

Returns:

- array :
- struct `system`
 - int `"id"`
 - string `"name"`
 - `dateTime.iso8601` `"last_checkin"` - last time server successfully checked in
 - `dateTime.iso8601` `"created"` - server registration time
 - `dateTime.iso8601` `"last_boot"` - last server boot time
 - int `"extra_pkg_count"` - number of packages not belonging to any assigned channel
 - int `"outdated_pkg_count"` - number of out-of-date packages

Chapter 782. Method: listUngroupedSystems

HTTP **GET**

Description:

List systems that are not associated with any system groups.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 783. Method: listUserSystems

HTTP **GET**

Description:

List systems for a given user.

Parameters:

- string sessionKey
- string login - User's login name.

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 784. Method: listUserSystems

HTTP **GET**

Description:

List systems for the logged in user.

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 785. Method: listVirtualGuests

HTTP **GET**

Description:

Lists the virtual guests for a given virtual host

Parameters:

- string sessionKey
- int sid - the virtual host's id

Returns:

- array :
- struct virtual system
 - int "id"
 - string "name"
 - string "guest_name" - the virtual guest name as provided by the virtual host
 - dateTime.iso8601 "last_checkin" - last time the server successfully checked in
 - string "uuid"

Chapter 786. Method: listVirtualHosts

HTTP **GET**

Description:

Lists the virtual hosts visible to the user

Parameters:

- string sessionKey

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 787. Method: obtainReactivationKey

HTTP **POST**

Description:

Obtains a reactivation key for this server.

Parameters:

- string sessionKey
- int sid

Returns:

- string key

Chapter 788. Method: obtainReactivationKey

HTTP **POST**

Description:

Obtains a reactivation key for this server.

Parameters:

- string clientCert - client certificate of the system

Returns:

- string key

Available since API version: 10.10

Chapter 789. Method: provisionSystem

HTTP **POST**

Description:

Provision a system using the specified kickstart/autoinstallation profile.

Parameters:

- string sessionKey
- int sid - ID of the system to be provisioned.
- string profileName - Profile to use.

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Chapter 790. Method: provisionSystem

HTTP **POST**

Description:

Provision a system using the specified kickstart/autoinstallation profile.

Parameters:

- string sessionKey
- int sid - ID of the system to be provisioned.
- string profileName - Profile to use.

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Chapter 791. Method: provisionSystem

HTTP **POST**

Description:

Provision a system using the specified kickstart/autoinstallation profile.

Parameters:

- string sessionKey
- int sid - ID of the system to be provisioned.
- string profileName - Profile to use.
- dateTime.iso8601 earliestDate

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Chapter 792. Method: provisionSystem

HTTP **POST**

Description:

Provision a system using the specified kickstart/autoinstallation profile.

Parameters:

- string sessionKey
- int sid - ID of the system to be provisioned.
- string profileName - Profile to use.
- dateTime.iso8601 earliestDate

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Chapter 793. Method: provisionVirtualGuest

HTTP **POST**

Description:

Provision a guest on the host specified. Defaults to: memory=512MB, vcpu=1, storage=3GB, mac_address=random.

Parameters:

- string sessionKey
- int sid - ID of host to provision guest on.
- string guestName
- string profileName - Kickstart profile to use.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 794. Method: provisionVirtualGuest

HTTP **POST**

Description:

Provision a guest on the host specified. This schedules the guest for creation and will begin the provisioning process when the host checks in or if OSAD is enabled will begin immediately. Defaults to mac_address=random.

Parameters:

- string sessionKey
- int sid - ID of host to provision guest on.
- string guestName
- string profileName - Kickstart Profile to use.
- int memoryMb - Memory to allocate to the guest
- int vcpus - Number of virtual CPUs to allocate to the guest.
- int storageGb - Size of the guests disk image.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 795. Method: provisionVirtualGuest

HTTP **POST**

Description:

Provision a guest on the host specified. This schedules the guest for creation and will begin the provisioning process when the host checks in or if OSAD is enabled will begin immediately.

Parameters:

- string sessionKey
- int sid - ID of host to provision guest on.
- string guestName
- string profileName - Kickstart Profile to use.
- int memoryMb - Memory to allocate to the guest
- int vcpus - Number of virtual CPUs to allocate to the guest.
- int storageGb - Size of the guests disk image.
- string macAddress - macAddress to give the guest's virtual networking hardware.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 796. Method: refreshPillar

HTTP **POST**

Description:

refresh all the pillar data of a list of systems.

Parameters:

- string sessionKey
- int array sids

Returns:

- int array skippedIds

Chapter 797. Method: refreshPillar

HTTP **POST**

Description:

refresh the pillar data of a list of systems. The subset value represents the pillar to be refreshed and can be one of 'general', 'group_membership', 'virtualization' or 'custom_info'.

Parameters:

- string sessionKey
- string subset
- int array sids

Returns:

- int array skippedIds

Chapter 798. Method: registerPeripheralServer

HTTP **POST**

Description:

Register foreign peripheral server. This is used for registering containerized peripheral servers.

Parameters:

- string sessionKey
- string fqdn - FQDN of the server

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 799. Method: removeEntitlements

HTTP **POST**

Description:

Remove addon entitlements from a server. Entitlements a server does not have are quietly ignored.

Parameters:

- string sessionKey
- int sid
- string array entitlements

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 800. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to multiple systems.

Parameters:

- string sessionKey
- int array sids
- int array errataIds

Returns:

- int array actionId

Available since API version: 13.0

Chapter 801. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to multiple systems.

Parameters:

- string sessionKey
- int array sids
- int array errataIds
- boolean allowModules – Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 802. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to multiple systems at a given date/time.

Parameters:

- string sessionKey
- int array sids
- int array errataIds
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Available since API version: 13.0

Chapter 803. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to multiple systems at a given date/time.

Parameters:

- string sessionKey
- int array sids
- int array errataIds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 804. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to multiple systems at a given date/time.

Parameters:

- string sessionKey
- int array sids
- int array errataIds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present
- boolean onlyRelevant - If true not all erratas are applied to all systems. Systems get only the erratas relevant for them.

Returns:

- int array actionId

Available since API version: 24

Chapter 805. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to a system.

Parameters:

- string sessionKey
- int sid
- int array errataIds

Returns:

- int array actionId

Available since API version: 13.0

Chapter 806. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to a system.

Parameters:

- string sessionKey
- int sid
- int array errataIds
- boolean allowModules – Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 807. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to a system at a given date/time.

Parameters:

- string sessionKey
- int sid
- int array errataIds
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Available since API version: 13.0

Chapter 808. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to a system at a given date/time.

Parameters:

- string sessionKey
- int sid
- int array errataIds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 809. Method: scheduleApplyErrata

HTTP **POST**

Description:

Schedules an action to apply errata updates to a system at a given date/time.

Parameters:

- string sessionKey
- int sid
- int array errataIds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present
- boolean onlyRelevant

Returns:

- int array actionId

Available since API version: 24

Chapter 810. Method: scheduleApplyHighstate

HTTP **POST**

Description:

Schedule highstate application for a given system.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestOccurrence
- boolean test - Run states in test-only mode

Returns:

- int actionId

Chapter 811. Method: scheduleApplyHighstate

HTTP **POST**

Description:

Schedule highstate application for a given system.

Parameters:

- string sessionKey
- int array sids
- dateTime.iso8601 earliestOccurrence
- boolean test - Run states in test-only mode

Returns:

- int actionId

Chapter 812. Method: scheduleApplyStates

HTTP **POST**

Description:

Schedule highstate application for a given system.

Parameters:

- string sessionKey
- int sid
- string array stateNames
- dateTime.iso8601 earliestOccurrence
- boolean test - Run states in test-only mode

Returns:

- int actionId

Chapter 813. Method: scheduleApplyStates

HTTP **POST**

Description:

Schedule highstate application for a given system.

Parameters:

- string sessionKey
- int array sids
- string array stateNames
- dateTime.iso8601 earliestOccurrence
- boolean test - Run states in test-only mode

Returns:

- int actionId

Chapter 814. Method: scheduleCertificateUpdate

HTTP **POST**

Description:

Schedule update of client certificate

Parameters:

- string sessionKey
- int sid

Returns:

- int actionId - The action id of the scheduled action

Chapter 815. Method: scheduleCertificateUpdate

HTTP **POST**

Description:

Schedule update of client certificate at given date and time

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 816. Method: scheduleChangeChannels

HTTP **POST**

Description:

Schedule an action to change the channels of the given system. Works for both traditional and Salt systems. This method accepts labels for the base and child channels. If the user provides an empty string for the channelLabel, the current base channel and all child channels will be removed from the system.

Parameters:

- string sessionKey
- int sid
- string baseChannelLabel
- string array childLabels
- dateTime.iso8601 earliestOccurrence - the time/date to schedule the action

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Available since API version: 19.0

Chapter 817. Method: scheduleChangeChannels

HTTP **POST**

Description:

Schedule an action to change the channels of the given system. Works for both traditional and Salt systems. This method accepts labels for the base and child channels. If the user provides an empty string for the channelLabel, the current base channel and all child channels will be removed from the system.

Parameters:

- string sessionKey
- int array sids
- string baseChannelLabel
- string array childLabels
- dateTime.iso8601 earliestOccurrence - the time/date to schedule the action

Returns:

- long array actionIds

Available since API version: 19.0

Chapter 818. Method: scheduleCoCoAttestation

HTTP **POST**

Description:

Schedule Confidential Compute Attestation Action

Parameters:

- string sessionKey
- int sid - ID of the server to run the script on.
- dateTime.iso8601 earliestOccurrence - Earliest the script can run.

Returns:

- int id - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`

Chapter 819. Method: scheduleDistUpgrade

HTTP **POST**

Description:

Schedule a dist upgrade for a system. This call takes a list of channel labels that the system will be subscribed to before performing the dist upgrade. Note: You can seriously damage your system with this call, use it only if you really know what you are doing! Make sure that the list of channel labels is complete and in any case do a dry run before scheduling an actual dist upgrade.

Parameters:

- string sessionKey
- int sid
- string array channels
- boolean dryRun
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId

Chapter 820. Method: scheduleDistUpgrade

HTTP **POST**

Description:

Schedule a dist upgrade for a system. This call takes a list of channel labels that the system will be subscribed to before performing the dist upgrade. Note: You can seriously damage your system with this call, use it only if you really know what you are doing! Make sure that the list of channel labels is complete and in any case do a dry run before scheduling an actual dist upgrade.

Parameters:

- string sessionKey
- int sid
- string array channels
- boolean dryRun
- boolean allowVendorChange
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId

Chapter 821. Method: scheduleGuestAction

HTTP **POST**

Description:

Schedules a guest action for the specified virtual guest for a given date/time.

Parameters:

- string sessionKey
- int sid - the system Id of the guest
- string state - One of the following actions 'start', 'suspend', 'resume', 'restart', 'shutdown'.
- dateTime.iso8601 date - the time/date to schedule the action

Returns:

- int actionId - The action id of the scheduled action

Chapter 822. Method: scheduleGuestAction

HTTP **POST**

Description:

Schedules a guest action for the specified virtual guest for the current time.

Parameters:

- string sessionKey
- int sid - the system Id of the guest
- string state - One of the following actions 'start', 'suspend', 'resume', 'restart', 'shutdown'.

Returns:

- int actionId - The action id of the scheduled action

Chapter 823. Method: scheduleHardwareRefresh

HTTP **POST**

Description:

Schedule a hardware refresh for a system.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 13.0

Chapter 824. Method: schedulePackageInstall

HTTP **POST**

Description:

Schedule package installation for several systems.

Parameters:

- string sessionKey
- int array sids
- int array packagelds
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Chapter 825. Method: schedulePackageInstall

HTTP **POST**

Description:

Schedule package installation for several systems.

Parameters:

- string sessionKey
- int array sids
- int array packagelds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 826. Method: schedulePackageInstall

HTTP **POST**

Description:

Schedule package installation for a system.

Parameters:

- string sessionKey
- int sid
- int array packagelds
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 13.0

Chapter 827. Method: schedulePackageInstall

HTTP **POST**

Description:

Schedule package installation for a system.

Parameters:

- string sessionKey
- int sid
- int array packagelds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 21

Chapter 828. Method: schedulePackageInstallByNevra

HTTP **POST**

Description:

Schedule package installation for several systems.

Parameters:

- string sessionKey
- int array sids
- array packageNevraList
 - struct Package Nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Chapter 829. Method: schedulePackageInstallByNevra

HTTP **POST**

Description:

Schedule package installation for several systems.

Parameters:

- string sessionKey
- int array sids
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 830. Method: schedulePackageInstallByNevra

HTTP **POST**

Description:

Schedule package installation for a system.

Parameters:

- string sessionKey
- int sid
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 831. Method: schedulePackageInstallByNevra

HTTP **POST**

Description:

Schedule package installation for a system.

Parameters:

- string sessionKey
- int sid
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 21

Chapter 832. Method: schedulePackageLockChange

HTTP **POST**

Description:

Schedule package lock for a system.

Parameters:

- string sessionKey
- int sid
- int array pkgIdsToLock
- int array pkgIdsToUnlock
- dateTime.iso8601 earliestOccurrence

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 833. Method: schedulePackageRefresh

HTTP **POST**

Description:

Schedule a package list refresh for a system.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestOccurrence

Returns:

- int id - ID of the action scheduled, otherwise exception thrown on error

Chapter 834. Method: schedulePackageRemove

HTTP **POST**

Description:

Schedule package removal for several systems.

Parameters:

- string sessionKey
- int array sids
- int array packagelds
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Chapter 835. Method: schedulePackageRemove

HTTP **POST**

Description:

Schedule package removal for several systems.

Parameters:

- string sessionKey
- int array sids
- int array packagelds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 836. Method: schedulePackageRemove

HTTP **POST**

Description:

Schedule package removal for a system.

Parameters:

- string sessionKey
- int sid
- int array packagelds
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 837. Method: schedulePackageRemove

HTTP **POST**

Description:

Schedule package removal for a system.

Parameters:

- string sessionKey
- int sid
- int array packagelds
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 21

Chapter 838. Method: schedulePackageRemoveByNevra

HTTP **POST**

Description:

Schedule package removal for several systems.

Parameters:

- string sessionKey
- int array sids
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Chapter 839. Method: schedulePackageRemoveByNevra

HTTP **POST**

Description:

Schedule package removal for several systems.

Parameters:

- string sessionKey
- int array sids
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 840. Method: schedulePackageRemoveByNevra

HTTP **POST**

Description:

Schedule package removal for a system.

Parameters:

- string sessionKey
- int sid
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Chapter 841. Method: schedulePackageRemoveByNevra

HTTP **POST**

Description:

Schedule package removal for a system.

Parameters:

- string sessionKey
- int sid
- array packageNevraList
 - struct Package nevra
 - string "package_name"
 - string "package_epoch"
 - string "package_version"
 - string "package_release"
 - string "package_arch"
- dateTime.iso8601 earliestOccurrence
- boolean allowModules - Allow this API call, despite modular content being present

Returns:

- int array actionId

Available since API version: 21

Chapter 842. Method: schedulePackageUpdate

HTTP **POST**

Description:

Schedule full package update for several systems.

Parameters:

- string sessionKey
- int array sids
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId

Available since API version: 25

Chapter 843. Method: scheduleProductMigration

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Parameters:

- string sessionKey
- int sid
- string baseChannelLabel
- string array optionalChildChannels
- boolean dryRun
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 844. Method: scheduleProductMigration

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Parameters:

- string sessionKey
- int sid
- string baseChannelLabel
- string array optionalChildChannels
- boolean dryRun
- boolean allowVendorChange
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 845. Method: scheduleProductMigration

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Parameters:

- string sessionKey
- int sid
- string targetIdent - Identifier for the selected migration target. Use listMigrationTargets to list the identifiers
- string baseChannelLabel
- string array optionalChildChannels
- boolean dryRun
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 846. Method: scheduleProductMigration

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Parameters:

- string sessionKey
- int sid
- string targetIdent - Identifier for the selected migration target. Use listMigrationTargets to list the identifiers
- string baseChannelLabel
- string array optionalChildChannels
- boolean dryRun
- boolean allowVendorChange
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 847. Method: scheduleProductMigration

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Parameters:

- string `sessionKey`
- int `sid`
- string `targetIdent` - Identifier for the selected migration target - User `listMigrationTargets` to list the identifiers
- string `baseChannelLabel`
- string array `optionalChildChannels`
- boolean `dryRun`
- boolean `allowVendorChange`
- boolean `removeProductsWithNoSuccessorAfterMigration` - set to remove products which have no successors. This flag will only have effect if `targetIdent` will also be specified
- `dateTime.iso8601` `earliestOccurrence`

Returns:

- int `actionId` - The action id of the scheduled action

Chapter 848. Method: scheduleReboot

HTTP **POST**

Description:

Schedule a reboot for a system.

Parameters:

- string sessionKey
- int sid
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 13.0

Chapter 849. Method: scheduleSPMigration (Deprecated)

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Note: This method is deprecated and will be removed in a future API version. Please use `scheduleProductMigration` instead.

Deprecated - being replaced by `scheduleProductMigration(User loggedInUser, Integer sid, String baseChannelLabel, List(String) optionalChildChannels, boolean dryRun, Date earliest)`

Parameters:

- string `sessionKey`
- int `sid`
- string `baseChannelLabel`
- string array `optionalChildChannels`
- boolean `dryRun`
- `dateTime.iso8601` `earliestOccurrence`

Returns:

- int `actionId` - The action id of the scheduled action

Chapter 850. Method: scheduleSPMigration (Deprecated)

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Note: This method is deprecated and will be removed in a future API version. Please use `scheduleProductMigration` instead.

Deprecated - being replaced by `scheduleProductMigration(User loggedInUser, Integer sid, String baseChannelLabel, List(String) optionalChildChannels, boolean dryRun, boolean allowVendorChange, Date earliest)`

Parameters:

- string `sessionKey`
- int `sid`
- string `baseChannelLabel`
- string array `optionalChildChannels`
- boolean `dryRun`
- boolean `allowVendorChange`
- `dateTime.iso8601` `earliestOccurrence`

Returns:

- int `actionId` - The action id of the scheduled action

Chapter 851. Method: scheduleSPMigration (Deprecated)

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Note: This method is deprecated and will be removed in a future API version. Please use `scheduleProductMigration` instead.

Deprecated - being replaced by `scheduleProductMigration(User loggedInUser, Integer sid, String targetId, String baseChannelLabel, List(String) optionalChildChannels, boolean dryRun, Date earliest)`

Parameters:

- string `sessionKey`
- int `sid`
- string `targetId` - identifier for the selected migration target. Use `listMigrationTargets` to list the identifiers
- string `baseChannelLabel`
- string array `optionalChildChannels`
- boolean `dryRun`
- `dateTime.iso8601` `earliestOccurrence`

Returns:

- int `actionId` - The action id of the scheduled action

Chapter 852. Method: scheduleSPMigration (Deprecated)

HTTP **POST**

Description:

Schedule a Product migration for a system. This call is the recommended and supported way of migrating a system to the next Service Pack. It will automatically find all mandatory product channels below a given target base channel and subscribe the system accordingly. Any additional optional channels can be subscribed by providing their labels.

Note: This method is deprecated and will be removed in a future API version. Please use `scheduleProductMigration` instead.

Deprecated - being replaced by `scheduleProductMigration(User loggedInUser, Integer sid, String targetId, String baseChannelLabel, List(String) optionalChildChannels, boolean dryRun, boolean allowVendorChange, Date earliest)`

Parameters:

- string `sessionKey`
- int `sid`
- string `targetId` - Identifier for the selected migration target. Use `listMigrationTargets` to list the identifiers
- string `baseChannelLabel`
- string array `optionalChildChannels`
- boolean `dryRun`
- boolean `allowVendorChange`
- `dateTime.iso8601` `earliestOccurrence`

Returns:

- int `actionId` - The action id of the scheduled action

Chapter 853. Method: scheduleScriptRun

HTTP **POST**

Description:

Schedule a script to run.

Parameters:

- string sessionKey
- string label
- int array sids - System IDs of the servers to run the script on.
- string username - User to run script as.
- string groupname - Group to run script as.
- int timeout - Seconds to allow the script to run before timing out.
- string script - Contents of the script to run. Must start with a shebang (e.g. #!/bin/bash)
- dateTime.iso8601 earliestOccurrence - Earliest the script can run.

Returns:

- int id - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`

Chapter 854. Method: scheduleScriptRun

HTTP **POST**

Description:

Schedule a script to run.

Parameters:

- string `sessionKey`
- int array `sids` - System IDs of the servers to run the script on.
- string `username` - User to run script as.
- string `groupname` - Group to run script as.
- int `timeout` - Seconds to allow the script to run before timing out.
- string `script` - Contents of the script to run. Must start with a shebang (e.g. `#!/bin/bash`)
- `dateTime.iso8601 earliestOccurrence` - Earliest the script can run.

Returns:

- int `id` - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`

Chapter 855. Method: scheduleScriptRun

HTTP **POST**

Description:

Schedule a script to run.

Parameters:

- string sessionKey
- int sid - ID of the server to run the script on.
- string username - User to run script as.
- string groupname - Group to run script as.
- int timeout - Seconds to allow the script to run before timing out.
- string script - Contents of the script to run. Must start with a shebang (e.g. #!/bin/bash)
- dateTime.iso8601 earliestOccurrence - Earliest the script can run.

Returns:

- int id - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`

Chapter 856. Method: scheduleScriptRun

HTTP **POST**

Description:

Schedule a script to run.

Parameters:

- string sessionKey
- string label
- int sid - ID of the server to run the script on.
- string username - User to run script as.
- string groupname - Group to run script as.
- int timeout - Seconds to allow the script to run before timing out.
- string script - Contents of the script to run. Must start with a shebang (e.g. #!/bin/bash)
- dateTime.iso8601 earliestOccurrence - Earliest the script can run.

Returns:

- int id - ID of the script run action created. Can be used to fetch results with `system.getScriptResults`

Chapter 857. Method: scheduleSyncPackagesWithSystem

HTTP **POST**

Description:

Sync packages from a source system to a target.

Parameters:

- string sessionKey
- int targetServerId - Target system to apply package changes to.
- int sourceServerId - Source system to retrieve package state from.
- int array packageIds - Package IDs to be synced.
- dateTime.iso8601 earliestOccurrence - Date to schedule action for

Returns:

- int actionId - The action id of the scheduled action

Available since API version: 13.0

Chapter 858. Method: searchByName

HTTP **POST**

Description:

Returns a list of system IDs whose name matches the supplied regular expression (defined by http://docs.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html_blank Java representation of regular expressions)

Parameters:

- string sessionKey
- string regexp - A regular expression

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time

Chapter 859. Method: sendOsaPing

HTTP **POST**

Description:

send a ping to a system using OSA

Parameters:

- string sessionKey
- int serverId

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 860. Method: setBaseChannel (Deprecated)

HTTP **POST**

Description:

Assigns the server to a new baseChannel.

Deprecated - being replaced by `system.setBaseChannel(string sessionKey, int serverId, string channelLabel)`

Parameters:

- string sessionKey
- int sid
- int cid - channel ID

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 861. Method: setBaseChannel (Deprecated)

HTTP **POST**

Description:

Assigns the server to a new base channel. If the user provides an empty string for the channelLabel, the current base channel and all child channels will be removed from the system.

Deprecated - being replaced by `system.scheduleChangeChannels(string sessionKey, int serverId, String baseChannelLabel, array_single channelLabels, date earliestOccurrence)`.

Parameters:

- string sessionKey
- int sid
- string channelLabel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 862. Method: setChildChannels (Deprecated)

HTTP **POST**

Description:

Subscribe the given server to the child channels provided. This method will unsubscribe the server from any child channels that the server is currently subscribed to, but that are not included in the list. The user may provide either a list of channel ids (int) or a list of channel labels (string) as input. Changes to channel assignments on salt managed systems will take effect at next highstate application.

Deprecated - being replaced by `system.scheduleChangeChannels(string sessionKey, int serverId, String baseChannelLabel, array_single channelLabels, date earliestOccurrence)`. This method will schedule an action for changing the child channels immediately.

Parameters:

- string sessionKey
- int sidd
- int (deprecated) or string array channelIdsOrLabels - channelId (deprecated) or channelLabel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 863. Method: setCoCoAttestationConfig

HTTP **POST**

Description:

Configure Confidential Compute Attestation for the given system

Parameters:

- string sessionKey
- int sid - ID of the server to get the config for.
- boolean enabled - set the enabled state for Confidential Compute Attestation
- string environmentType - set the environment type of the system:
 - NONE
 - KVM_AMD_EPYC_MILAN
 - KVM_AMD_EPYC_GENOA
- boolean attestOnBoot - set if the attestation should be performed on system boot

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 864. Method: setCustomValues

HTTP **POST**

Description:

Set custom values for the specified server.

Parameters:

- string sessionKey
- int sid
- struct values
 - string "custom info label"
 - string "value"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 865. Method: setDetails

HTTP **POST**

Description:

Set server details. All arguments are optional and will only be modified if included in the struct.

Parameters:

- string `sessionKey`
- int `sid` - ID of server to lookup details for.
- struct `details`
 - string `"profile_name"` - System's profile name
 - string `"base_entitlement"` - System's base entitlement label. (enterprise_entitled or unentitled)
 - boolean `"auto_errata_update"` - True if system has auto errata updates enabled
 - string `"description"` - System description
 - string `"address1"` - System's address line 1.
 - string `"address2"` - System's address line 2.
 - string `"city"`
 - string `"state"`
 - string `"country"`
 - string `"building"`
 - string `"room"`
 - string `"rack"`
 - string `"contact_method"` - One of the following:
 - default
 - ssh-push
 - ssh-push-tunnel

Returns:

-
- int - 1 on success, exception thrown otherwise.

Chapter 866. Method: setGroupMembership

HTTP **POST**

Description:

Set a servers membership in a given group.

Parameters:

- string sessionKey
- int sid
- int sgid
- boolean member - '1' to assign the given server to the given server group, '0' to remove the given server from the given server group.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 867. Method: setGuestCpus

HTTP **POST**

Description:

Schedule an action of a guest's host, to set that guest's CPU allocation

Parameters:

- string sessionKey
- int sid - The guest's system id
- int numOfCpus - The number of virtual cpus to allocate to the guest

Returns:

- int actionID - the action Id for the schedule action on the host system

Chapter 868. Method: setGuestMemory

HTTP **POST**

Description:

Schedule an action of a guest's host, to set that guest's memory allocation

Parameters:

- string sessionKey
- int sid - The guest's system id
- int memory - The amount of memory to allocate to the guest

Returns:

- int actionID - the action Id for the schedule action on the host system

Chapter 869. Method: setLockStatus

HTTP **POST**

Description:

Set server lock status.

Parameters:

- string sessionKey
- int sid
- boolean lockStatus - true to lock the system, false to unlock the system.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 870. Method: setPillar

HTTP **POST**

Description:

Set pillar data of a system.

Parameters:

- string sessionKey
- int systemId
- struct pillarData

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 871. Method: setPillar

HTTP **POST**

Description:

Set pillar data of a system.

Parameters:

- string sessionKey
- int systemId
- struct pillarData

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 872. Method: setPrimaryFqdn

HTTP **POST**

Description:

Sets new primary FQDN

Parameters:

- string sessionKey
- int sid
- string fqdn

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 873. Method: setPrimaryInterface

HTTP **POST**

Description:

Sets new primary network interface

Parameters:

- string sessionKey
- int sid
- string interfaceName

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 874. Method: setProfileName

HTTP **POST**

Description:

Set the profile name for the server.

Parameters:

- string sessionKey
- int sid
- string name - Name of the profile.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 875. Method: setVariables

HTTP **POST**

Description:

Sets a list of kickstart variables in the cobbler system record for the specified server. Note: This call assumes that a system record exists in cobbler for the given system and will raise an XMLRPC fault if that is not the case. To create a system record over xmlrpc use `system.createSystemRecord`

To create a system record in the Web UI please go to System > Specified System > Provisioning > Select a Kickstart profile > Create Cobbler System Record.

Parameters:

- string `sessionKey`
- int `sid`
- boolean `netboot`
- struct `variables`
 - string `"key"`
 - string or int `"value"`

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 876. Method: tagLatestSnapshot

HTTP **POST**

Description:

Tags latest system snapshot

Parameters:

- string sessionKey
- int sid
- string tagName

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 877. Method: unentitle

HTTP **POST**

Description:

Unentitle the system completely

Parameters:

- string clientCert - client system id file

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 878. Method: updatePackageState

HTTP **POST**

Description:

Update the package state of a given system (High state would be needed to actually install/remove the package)

Parameters:

- string sessionKey
- int sid
- string packageName - Name of the package
- int state - 0 = installed, 1 = removed, 2 = unmanaged
- int versionConstraint - 0 = latest, 1 = any

Returns:

- 1 on success, exception on failure

Chapter 879. Method: updatePeripheralServerInfo

HTTP **POST**

Description:

Update foreign peripheral server info.

Parameters:

- string sessionKey
- string reportDbName - ReportDB name
- string reportDbHost - ReportDB host
- int reportDbPort - ReportDB port
- string reportDbUser - ReportDB user
- string reportDbPassword - ReportDB password

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 880. Method: upgradeEntitlement

HTTP **POST**

Description:

Adds an entitlement to a given server.

Parameters:

- string sessionKey
- int sid
- string entitlementLevel - One of: 'enterprise_entitled' or 'virtualization_host'.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 881. Method: whoRegistered

HTTP **POST**

Description:

Returns information about the user who registered the system

Parameters:

- string sessionKey
- int sid - Id of the system in question

Returns:

- * struct user
 - int "id"
 - string "login"
 - string "login_uc" - upper case version of the login
 - boolean "enabled" - true if user is enabled, false if the user is disabled

system.appstreams

Chapter 882. Available methods

- `disable`
- `enable`
- `listModuleStreams`

Chapter 883. Description

Provides methods to handle appstreams for systems.

Namespace:

system.appstreams

Chapter 884. Method: disable

HTTP **POST**

Description:

Schedule disabling of module streams. Invalid modules will be filtered out. If all provided modules are invalid the request will fail.

Parameters:

- string sessionKey
- int sid
- array moduleStreams
 - struct Module Stream
 - string "module"
 - string "stream"
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 885. Method: enable

HTTP **POST**

Description:

Schedule enabling of module streams. Invalid modules will be filtered out. If all provided modules are invalid the request will fail.

Parameters:

- string sessionKey
- int sid
- array moduleStreams
 - struct Module Stream
 - string "module"
 - string "stream"
- dateTime.iso8601 earliestOccurrence

Returns:

- int actionId - The action id of the scheduled action

Chapter 886. Method: listModuleStreams

HTTP **GET**

Description:

List available module streams for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct ChannelAppStreams
 - string "channel_name"
 - array "AppStreams"
- struct AppStream
 - boolean "is_enabled"
 - string "stream"
 - string "module"
 - string "arch"

system.config

Chapter 887. Available methods

- `addChannels`
- `createOrUpdatePath`
- `createOrUpdateSymlink`
- `deleteFiles`
- `deployAll`
- `listChannels`
- `listFiles`
- `lookupFileInfo`
- `removeChannels`
- `scheduleApplyConfigChannel`
- `setChannels`

Chapter 888. Description

Provides methods to access and modify many aspects of configuration channels and server association. basically system.config name space

Namespace:

system.config

Chapter 889. Method: addChannels

HTTP **POST**

Description:

Given a list of servers and configuration channels, this method appends the configuration channels to either the top or the bottom (whichever you specify) of a system's subscribed configuration channels list. The ordering of the configuration channels provided in the add list is maintained while adding. If one of the configuration channels in the 'add' list has been previously subscribed by a server, the subscribed channel will be re-ranked to the appropriate place.

Parameters:

- string sessionKey
- int array sids - IDs of the systems to add the channels to.
- string array configChannelLabels - List of configuration channel labels in the ranked order.
- boolean addToTop
 - true - to prepend the given channels list to the top of the configuration channels list of a server
 - false - to append the given channels list to the bottom of the configuration channels list of a server

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 890. Method: createOrUpdatePath

HTTP **POST**

Description:

Create a new file (text or binary) or directory with the given path, or update an existing path on a server.

Parameters:

- string `sessionKey`
- int `sid`
- string `path` - the configuration file/directory path
- boolean `isDir`
 - True - if the path is a directory
 - False - if the path is a file
- struct data
 - string `"contents"` - Contents of the file (text or base64 encoded if binary) ((only for non-directories)
 - boolean `"contents_enc64"` - Identifies base64 encoded content (default: disabled, only for non-directories).
 - string `"owner"` - Owner of the file/directory.
 - string `"group"` - Group name of the file/directory.
 - string `"permissions"` - Octal file/directory permissions (eg: 644)
 - string `"macro-start-delimiter"` - Config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
 - string `"macro-end-delimiter"` - Config file macro end delimiter. Use null or empty string to accept the default. (only for non-directories)
 - string `"selinux_ctx"` - SELinux context (optional)
 - int `"revision"` - next revision number, auto increment for null
 - boolean `"binary"` - mark the binary content, if True, base64 encoded content is expected (only for non-directories)

-
- boolean `commitToLocal`
 - 1 - to commit configuration files to the system's local override configuration channel
 - 0 - to commit configuration files to the system's sandbox configuration channel

Returns:

- * struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the `contents_enc64` attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - `dateTime.iso8601` "creation" - Creation Date
 - `dateTime.iso8601` "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - `true/false` , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 891. Method: createOrUpdateSymlink

HTTP **POST**

Description:

Create a new symbolic link with the given path, or update an existing path.

Parameters:

- string `sessionKey`
- int `sid`
- string `path` - the configuration file/directory path
- struct `data`
 - string `"target_path"` - The target path for the symbolic link
 - string `"selinux_ctx"` - SELinux Security context (optional)
 - int `"revision"` - next revision number, auto increment for null
- boolean `commitToLocal`
 - 1 - to commit configuration files to the system's local override configuration channel
 - 0 - to commit configuration files to the system's sandbox configuration channel

Returns:

- * struct configuration revision information
 - string `"type"`
 - file
 - directory
 - symlink
 - string `"path"` - File Path
 - string `"target_path"` - Symbolic link Target File Path. Present for Symbolic links only.
 - string `"channel"` - Channel Name
 - string `"contents"` - File contents (base64 encoded according to the `contents_enc64` attribute)

-
- boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.
 - string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 892. Method: deleteFiles

HTTP **POST**

Description:

Removes file paths from a local or sandbox channel of a server.

Parameters:

- string sessionKey
- int sid
- string array paths
- boolean deleteFromLocal
 - True - to delete configuration file paths from the system's local override configuration channel
 - False - to delete configuration file paths from the system's sandbox configuration channel

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 893. Method: deployAll

HTTP **POST**

Description:

Schedules a deploy action for all the configuration files on the given list of systems.

Parameters:

- string sessionKey
- int array sids - IDs of the systems to schedule configuration files deployment
- dateTime.iso8601 date - Earliest date for the deploy action.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 894. Method: listChannels

HTTP **GET**

Description:

List all global('Normal', 'State') configuration channels associated to a system in the order of their ranking.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Chapter 895. Method: listFiles

HTTP **GET**

Description:

Return the list of files in a given channel.

Parameters:

- string `sessionKey`
- int `sid`
- int `listLocal`
 - 1 - to return configuration files in the system's local override configuration channel
 - 0 - to return configuration files in the system's sandbox configuration channel

Returns:

- array :
- struct configuration file information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "channel_label" - the label of the central configuration channel that has this file.
Note this entry only shows up if the file has not been overridden by a central channel.
 - struct "channel_type"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

-
- `dateTime.iso8601 "last_modified"` - Last Modified Date

Chapter 896. Method: lookupFileInfo

HTTP **GET**

Description:

Given a list of paths and a server, returns details about the latest revisions of the paths.

Parameters:

- string `sessionKey`
- int `sid`
- string array `paths` - paths to lookup on.
- boolean `searchLocal`
 - 1 - to search configuration file paths in the system's local override configuration or systems subscribed central channels
 - 0 - to search configuration file paths in the system's sandbox configuration channel

Returns:

- array :
- struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the `contents_enc64` attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision

-
- `dateTime.iso8601 "creation"` - Creation Date
 - `dateTime.iso8601 "modified"` - Last Modified Date
 - `string "owner"` - File Owner. Present for files or directories only.
 - `string "group"` - File Group. Present for files or directories only.
 - `int "permissions"` - File Permissions (Deprecated). Present for files or directories only.
 - `string "permissions_mode"` - File Permissions. Present for files or directories only.
 - `string "selinux_ctx"` - SELinux Context (optional).
 - `boolean "binary"` - `true/false` , Present for files only.
 - `string "sha256"` - File's sha256 signature. Present for files only.
 - `string "macro-start-delimiter"` - Macro start delimiter for a config file. Present for text files only.
 - `string "macro-end-delimiter"` - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 897. Method: removeChannels

HTTP **POST**

Description:

Remove config channels from the given servers.

Parameters:

- string sessionKey
- int array sids - the IDs of the systems from which you would like to remove configuration channels..
- string array configChannelLabels - List of configuration channel labels to remove.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 898. Method: scheduleApplyConfigChannel

HTTP **POST**

Description:

Schedule highstate application for a given system.

Parameters:

- string sessionKey
- int array sids
- dateTime.iso8601 earliestOccurrence
- boolean test - Run states in test-only mode

Returns:

- int actionId

Chapter 899. Method: setChannels

HTTP **POST**

Description:

Replace the existing set of config channels on the given servers. Channels are ranked according to their order in the configChannellLabels array.

Parameters:

- string sessionKey
- int array sids - IDs of the systems to set the channels on.
- string array configChannellLabels - List of configuration channel labels in the ranked order.

Returns:

- int - 1 on success, exception thrown otherwise.

system.custominfo

Chapter 900. Available methods

- `createKey`
- `deleteKey`
- `listAllKeys`
- `updateKey`

Chapter 901. Description

Provides methods to access and modify custom system information.

Namespace:

system.custominfo

Chapter 902. Method: createKey

HTTP **POST**

Description:

Create a new custom key

Parameters:

- string sessionKey
- string keyLabel - new key's label
- string keyDescription - new key's description

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 903. Method: deleteKey

HTTP **POST**

Description:

Delete an existing custom key and all systems' values for the key.

Parameters:

- string sessionKey
- string keyLabel - new key's label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 904. Method: listAllKeys

HTTP **GET**

Description:

List the custom information keys defined for the user's organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct custom info
 - int "id"
 - string "label"
 - string "description"
 - int "system_count"
 - dateTime.iso8601 "last_modified"

Chapter 905. Method: updateKey

HTTP **POST**

Description:

Update description of a custom key

Parameters:

- string sessionKey
- string keyLabel - key to change
- string keyDescription - new key's description

Returns:

- int - 1 on success, exception thrown otherwise.

system.monitoring

Chapter 906. Available methods

- `listEndpoints`

Chapter 907. Description

Provides methods to access information about managed systems, applications and formulas which can be relevant for Prometheus monitoring

Namespace:

system.monitoring

Chapter 908. Method: listEndpoints

HTTP **GET**

Description:

Get the list of monitoring endpoint details.

Parameters:

- string sessionKey
- int array sids

Returns:

- array :
- struct endpoint info
 - int "system_id"
 - string "endpoint_name"
 - string "exporter_name"
 - string "module"
 - string "path"
 - int "port"
 - bool "tls_enabled"

system.provisioning.powermanagement

Chapter 909. Available methods

- `getDetails`
- `getDetails`
- `getStatus`
- `getStatus`
- `listTypes`
- `powerOff`
- `powerOff`
- `powerOn`
- `powerOn`
- `reboot`
- `reboot`
- `setDetails`
- `setDetails`

Chapter 910. Description

Provides methods to access and modify power management for systems. Some functions exist in 2 variants. Either with server id or with a name. The function with server id is useful when a system exists with a full profile. Everybody allowed to manage that system can execute these functions. The variant with name expects a cobbler system name prefix. These functions enhance the name by adding the org id of the user to limit access to systems from the own organization. Additionally Org Admin permissions are required to call these functions.

Namespace:

system.provisioning.powermanagement

Chapter 911. Method: getDetails

HTTP **GET**

Description:

Get current power management settings of the given system

Parameters:

- string sessionKey
- int sid

Returns:

- struct powerManagementParameters
 - string "powerType" - Power management type
 - string "powerAddress" - IP address for power management
 - string "powerUsername" - The Username
 - string "powerPassword" - The Password
 - string "powerId" - Identifier

Chapter 912. Method: getDetails

HTTP **GET**

Description:

Get current power management settings of the given system

Parameters:

- string sessionKey
- string name

Returns:

- struct powerManagementParameters
 - string "powerType" - Power management type
 - string "powerAddress" - IP address for power management
 - string "powerUsername" - The Username
 - string "powerPassword" - The Password
 - string "powerId" - Identifier

Chapter 913. Method: getStatus

HTTP **GET**

Description:

Execute powermanagement actions

Parameters:

- string sessionKey
- int sid

Returns:

- boolean status - True when power is on, otherwise False

Chapter 914. Method: getStatus

HTTP **GET**

Description:

Execute powermanagement actions

Parameters:

- string sessionKey
- string name

Returns:

- boolean status - True when power is on, otherwise False

Chapter 915. Method: listTypes

HTTP **GET**

Description:

Return a list of available power management types

Parameters:

- string sessionKey

Returns:

- string array power management types

Chapter 916. Method: powerOff

HTTP **POST**

Description:

Execute power management action 'powerOff'

Parameters:

- string sessionKey
- int sid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 917. Method: powerOff

HTTP **POST**

Description:

Execute power management action 'powerOff'

Parameters:

- string sessionKey
- string name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 918. Method: powerOn

HTTP **POST**

Description:

Execute power management action 'powerOn'

Parameters:

- string sessionKey
- int sid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 919. Method: powerOn

HTTP **POST**

Description:

Execute power management action 'powerOn'

Parameters:

- string sessionKey
- string name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 920. Method: reboot

HTTP **POST**

Description:

Execute power management action 'Reboot'

Parameters:

- string sessionKey
- int sid

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 921. Method: reboot

HTTP **POST**

Description:

Execute power management action 'Reboot'

Parameters:

- string sessionKey
- string name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 922. Method: setDetails

HTTP **POST**

Description:

Get current power management settings of the given system

Parameters:

- string sessionKey
- int sid
- struct data
 - string "powerType" - Power management type
 - string "powerAddress" - IP address for power management
 - string "powerUsername" - The Username
 - string "powerPassword" - The Password
 - string "powerId" - Identifier

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 923. Method: setDetails

HTTP **POST**

Description:

Get current power management settings of the given system

Parameters:

- string sessionKey
- string name
- struct data
 - string "powerType" - Power management type
 - string "powerAddress" - IP address for power management
 - string "powerUsername" - The Username
 - string "powerPassword" - The Password
 - string "powerId" - Identifier

Returns:

- int - 1 on success, exception thrown otherwise.

system.provisioning.snapshot

Chapter 924. Available methods

- `addTagToSnapshot`
- `deleteSnapshot`
- `deleteSnapshots`
- `deleteSnapshots`
- `deleteSnapshots`
- `deleteSnapshots`
- `listSnapshotConfigFiles`
- `listSnapshotPackages`
- `listSnapshots`
- `listSnapshots`
- `rollbackToSnapshot`
- `rollbackToTag`
- `rollbackToTag`

Chapter 925. Description

Provides methods to access and delete system snapshots.

Namespace:

`system.provisioning.snapshot`

Chapter 926. Method: addTagToSnapshot

HTTP **POST**

Description:

Adds tag to snapshot

Parameters:

- string sessionKey
- int snapId - ID of the snapshot
- string tagName - Name of the snapshot tag

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 927. Method: deleteSnapshot

HTTP **POST**

Description:

Deletes a snapshot with the given snapshot id

Parameters:

- string sessionKey
- int snapId - ID of snapshot to delete

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.1

Chapter 928. Method: deleteSnapshots

HTTP **POST**

Description:

Deletes all snapshots across multiple systems based on the given date criteria. For example,

If the user provides `startDate` only, all snapshots created either on or after the date provided will be removed.

If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be removed.

If the user doesn't provide a `startDate` and `endDate`, all snapshots will be removed.

Parameters:

- string `sessionKey`
- `dateTime.iso8601` `startDate`
- `dateTime.iso8601` `endDate`

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.1

Chapter 929. Method: deleteSnapshots

HTTP **POST**

Description:

Deletes all snapshots for a given system based on the date criteria. For example,

If the user provides startDate only, all snapshots created either on or after the date provided will be removed.

If user provides startDate and endDate, all snapshots created on or between the dates provided will be removed.

If the user doesn't provide a startDate and endDate, all snapshots associated with the server will be removed.

Parameters:

- string sessionKey
- int sid - ID of system to delete snapshots for
- dateTime.iso8601 startDate
- dateTime.iso8601 endDate

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.1

Chapter 930. Method: deleteSnapshots

HTTP **POST**

Description:

Deletes all snapshots across multiple systems based on the given date criteria. For example,

If the user provides `startDate` only, all snapshots created either on or after the date provided will be removed.

If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be removed.

If the user doesn't provide a `startDate` and `endDate`, all snapshots will be removed.

Parameters:

- string `sessionKey`
- struct `dateDetails`
 - `dateTime.iso8601 "startDate"` - Optional, unless `endDate` is provided.
 - `dateTime.iso8601 "endDate"` - Optional.

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.1

Chapter 931. Method: deleteSnapshots

HTTP **POST**

Description:

Deletes all snapshots for a given system based on the date criteria. For example,

If the user provides `startDate` only, all snapshots created either on or after the date provided will be removed.

If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be removed.

If the user doesn't provide a `startDate` and `endDate`, all snapshots associated with the server will be removed.

Parameters:

- string `sessionKey`
- int `sid` - ID of system to delete snapshots for
- struct `dateDetails`
 - `dateTime.iso8601 "startDate"` - Optional, unless `endDate` is provided.
 - `dateTime.iso8601 "endDate"` - Optional.

Returns:

- int - 1 on success, exception thrown otherwise.

Available since API version: 10.1

Chapter 932. Method: listSnapshotConfigFiles

HTTP **GET**

Description:

List the config files associated with a snapshot.

Parameters:

- string sessionKey
- int snapId

Returns:

- array :
- struct configuration revision information
 - string "type"
 - file
 - directory
 - symlink
 - string "path" - File Path
 - string "target_path" - Symbolic link Target File Path. Present for Symbolic links only.
 - string "channel" - Channel Name
 - string "contents" - File contents (base64 encoded according to the contents_enc64 attribute)
 - boolean "contents_enc64" - Identifies base64 encoded content
 - int "revision" - File Revision
 - dateTime.iso8601 "creation" - Creation Date
 - dateTime.iso8601 "modified" - Last Modified Date
 - string "owner" - File Owner. Present for files or directories only.
 - string "group" - File Group. Present for files or directories only.
 - int "permissions" - File Permissions (Deprecated). Present for files or directories only.

-
- string "permissions_mode" - File Permissions. Present for files or directories only.
 - string "selinux_ctx" - SELinux Context (optional).
 - boolean "binary" - true/false , Present for files only.
 - string "sha256" - File's sha256 signature. Present for files only.
 - string "macro-start-delimiter" - Macro start delimiter for a config file. Present for text files only.
 - string "macro-end-delimiter" - Macro end delimiter for a config file. Present for text files only.

Available since API version: 10.2

Chapter 933. Method: listSnapshotPackages

HTTP **GET**

Description:

List the packages associated with a snapshot.

Parameters:

- string sessionKey
- int snapId

Returns:

- array :
- struct package nevra
 - string "name"
 - string "epoch"
 - string "version"
 - string "release"
 - string "arch"

Available since API version: 10.1

Chapter 934. Method: listSnapshots

HTTP **GET**

Description:

List snapshots for a given system. A user may optionally provide a start and end date to narrow the snapshots that will be listed. For example,

If the user provides `startDate` only, all snapshots created either on or after the date provided will be returned.

If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be returned.

If the user doesn't provide a `startDate` and `endDate`, all snapshots associated with the server will be returned.

Parameters:

- string `sessionKey`
- int `sid`
- `dateTime.iso8601` `startDate`
- `dateTime.iso8601` `endDate`

Returns:

- array :
- struct server snapshot
 - int "id"
 - string "reason" - the reason for the snapshot's existence
 - `dateTime.iso8601` "created"
 - string array "channels" - labels of channels associated with the snapshot
 - string array "groups" - names of server groups associated with the snapshot
 - string array "entitlements" - names of system entitlements associated with the snapshot
 - string array "config_channels" - labels of config channels the snapshot is associated with
 - string array "tags" - tag names associated with this snapshot

-
- string "Invalid_reason" - if the snapshot is invalid, this is the reason (optional)

Available since API version: 10.1

Chapter 935. Method: listSnapshots

HTTP **GET**

Description:

List snapshots for a given system. A user may optionally provide a start and end date to narrow the snapshots that will be listed. For example,

If the user provides `startDate` only, all snapshots created either on or after the date provided will be returned.

If user provides `startDate` and `endDate`, all snapshots created on or between the dates provided will be returned.

If the user doesn't provide a `startDate` and `endDate`, all snapshots associated with the server will be returned.

Parameters:

- string `sessionKey`
- int `sid`
- struct `dateDetails`
 - `dateTime.iso8601 "startDate"` - Optional, unless `endDate` is provided.
 - `dateTime.iso8601 "endDate"` - Optional.

Returns:

- array :
- struct `server snapshot`
 - int `"id"`
 - string `"reason"` - the reason for the snapshot's existence
 - `dateTime.iso8601 "created"`
 - string array `"channels"` - labels of channels associated with the snapshot
 - string array `"groups"` - names of server groups associated with the snapshot
 - string array `"entitlements"` - names of system entitlements associated with the snapshot
 - string array `"config_channels"` - labels of config channels the snapshot is associated with

-
- string array "tags" - tag names associated with this snapshot
 - string "Invalid_reason" - if the snapshot is invalid, this is the reason (optional)

Available since API version: 10.1

Chapter 936. Method: rollbackToSnapshot

HTTP **POST**

Description:

Rollbacks server to snapshot

Parameters:

- string sessionKey
- int sid
- int snapId - ID of the snapshot

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 937. Method: rollbackToTag

HTTP **POST**

Description:

Rollbacks server to snapshot

Parameters:

- string sessionKey
- int sid
- string tagName - Name of the snapshot tag

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 938. Method: rollbackToTag

HTTP **POST**

Description:

Rollbacks server to snapshot

Parameters:

- string sessionKey
- string tagName - Name of the snapshot tag

Returns:

- int - 1 on success, exception thrown otherwise.

system.scap

Chapter 939. Available methods

- `deleteXccdfScan`
- `getXccdfScanDetails`
- `getXccdfScanRuleResults`
- `listXccdfScans`
- `scheduleXccdfScan`
- `scheduleXccdfScan`
- `scheduleXccdfScan`
- `scheduleXccdfScan`
- `scheduleXccdfScan`
- `scheduleXccdfScan`

Chapter 940. Description

Provides methods to schedule SCAP scans and access the results.

Namespace:

system.scap

Chapter 941. Method: deleteXccdfScan

HTTP **POST**

Description:

Delete OpenSCAP XCCDF Scan from the #product() database. Note that only those SCAP Scans can be deleted which have passed their retention period.

Parameters:

- string sessionKey
- int xid - ID of XCCDF scan.

Returns:

- boolean status - indicates success of the operation

Chapter 942. Method: getXccdfScanDetails

HTTP **GET**

Description:

Get details of given OpenSCAP XCCDF scan.

Parameters:

- string sessionKey
- int xid – ID of XCCDF scan.

Returns:

- * struct OpenSCAP XCCDF Scan
 - int "xid" – XCCDF TestResult ID
 - int "sid" – serverId
 - int "action_id" – ID of the parent action
 - string "path" – path to XCCDF document
 - string "ovalfiles" – optional OVAL files
 - string "oscap_parameters" – oscap command-line arguments
 - string "test_result" – identifier of XCCDF TestResult
 - string "benchmark" – identifier of XCCDF Benchmark
 - string "benchmark_version" – version of the Benchmark
 - string "profile" – identifier of XCCDF Profile
 - string "profile_title" – title of XCCDF Profile
 - dateTime.iso8601 "start_time" – client machine time of scan start
 - dateTime.iso8601 "end_time" – client machine time of scan completion
 - string "errors" – stderr output of scan
 - boolean "deletable" – indicates whether the scan can be deleted

Chapter 943. Method: getXccdfScanRuleResults

HTTP **GET**

Description:

Return a full list of RuleResults for given OpenSCAP XCCDF scan.

Parameters:

- string sessionKey
- int xid – ID of XCCDF scan.

Returns:

- array :
- struct OpenSCAP XCCDF RuleResult
 - string "idref" – idref from XCCDF document
 - string "result" – result of evaluation
 - string "idents" – comma separated list of XCCDF idents

Chapter 944. Method: listXccdfScans

HTTP **GET**

Description:

Return a list of finished OpenSCAP scans for a given system.

Parameters:

- string sessionKey
- int sid

Returns:

- array :
- struct OpenSCAP XCCDF Scan
 - int "xid" - XCCDF TestResult ID
 - string "profile" - XCCDF Profile
 - string "path" - path to XCCDF document
 - string "ovalfiles" - optional OVAL files
 - dateTime.iso8601 "completed" - scan completion time

Chapter 945. Method: scheduleXccdfScan

HTTP **POST**

Description:

Schedule OpenSCAP scan.

Parameters:

- string sessionKey
- int array sids
- string xccdfPath - path to xccdf content on targeted systems.
- string oscapParams - additional parameters for oscap tool.

Returns:

- int id - ID if SCAP action created

Chapter 946. Method: scheduleXccdfScan

HTTP **POST**

Description:

Schedule OpenSCAP scan.

Parameters:

- string sessionKey
- int array sids
- string xccdfPath - path to xccdf content on targeted systems.
- string oscapParams - additional parameters for oscap tool.
- dateTime.iso8601 date - The date to schedule the action

Returns:

- int id - ID if SCAP action created

Chapter 947. Method: scheduleXccdfScan

HTTP **POST**

Description:

Schedule OpenSCAP scan.

Parameters:

- string sessionKey
- int array sids
- string xccdfPath - Path to xccdf content on targeted systems.
- string oscapPrms - Additional parameters for oscap tool.
- string ovalFiles - Additional OVAL files for oscap tool.
- dateTime.iso8601 date - The date to schedule the action

Returns:

- int id - ID if SCAP action created

Chapter 948. Method: scheduleXccdfScan

HTTP **POST**

Description:

Schedule Scap XCCDF scan.

Parameters:

- string sessionKey
- int sid
- string xccdfPath - Path to xccdf content on targeted systems.
- string oscapPrms - Additional parameters for oscap tool.

Returns:

- int id - ID of the scap action created

Chapter 949. Method: scheduleXccdfScan

HTTP **POST**

Description:

Schedule Scap XCCDF scan.

Parameters:

- string sessionKey
- int sid
- string xccdfPath - Path to xccdf content on targeted systems.
- string oscapPrms - Additional parameters for oscap tool.
- dateTime.iso8601 date - The date to schedule the action

Returns:

- int id - ID of the scap action created

system.search

Chapter 950. Available methods

- `deviceDescription`
- `deviceDriver`
- `deviceId`
- `deviceVendorId`
- `hostname`
- `ip`
- `nameAndDescription`
- `uuid`

Chapter 951. Description

Provides methods to perform system search requests using the search server.

Namespace:

system.search

Chapter 952. Method: deviceDescription

HTTP **GET**

Description:

List the systems which match the device description.

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 953. Method: deviceDriver

HTTP **GET**

Description:

List the systems which match this device driver.

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 954. Method: deviceId

HTTP **GET**

Description:

List the systems which match this device id

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 955. Method: deviceVendorId

HTTP **GET**

Description:

List the systems which match this device vendor_id

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 956. Method: hostname

HTTP **GET**

Description:

List the systems which match this hostname

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 957. Method: ip

HTTP **GET**

Description:

List the systems which match this ip.

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 958. Method: nameAndDescription

HTTP **GET**

Description:

List the systems which match this name or description

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

Chapter 959. Method: uuid

HTTP **GET**

Description:

List the systems which match this UUID

Parameters:

- string sessionKey
- string searchTerm

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - string "hostname"
 - string "uuid"
 - string "ip"
 - string "hw_description" - HW description if not null
 - string "hw_device_id" - HW device id if not null
 - string "hw_vendor_id" - HW vendor id if not null
 - string "hw_driver" - HW driver if not null

systemgroup

Chapter 960. Available methods

- `addOrRemoveAdmins`
- `addOrRemoveSystems`
- `create`
- `delete`
- `getDetails`
- `getDetails`
- `listActiveSystemsInGroup`
- `listAdministrators`
- `listAllGroups`
- `listAssignedConfigChannels`
- `listAssignedFormuals`
- `listGroupsWithNoAssociatedAdmins`
- `listInactiveSystemsInGroup`
- `listInactiveSystemsInGroup`
- `listSystems`
- `listSystemsMinimal`
- `scheduleApplyErrataToActive`
- `scheduleApplyErrataToActive`
- `scheduleApplyErrataToActive`
- `subscribeConfigChannel`
- `unsubscribeConfigChannel`
- `update`

Chapter 961. Description

Provides methods to access and modify system groups.

Namespace:

systemgroup

Chapter 962. Method: addOrRemoveAdmins

HTTP **POST**

Description:

Add or remove administrators to/from the given group. #product() and Organization administrators are granted access to groups within their organization by default; therefore, users with those roles should not be included in the array provided. Caller must be an organization administrator.

Parameters:

- string sessionKey
- string systemGroupName
- string array loginName - User's loginName
- int add - 1 to add administrators, 0 to remove.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 963. Method: addOrRemoveSystems

HTTP **POST**

Description:

Add/remove the given servers to a system group.

Parameters:

- string sessionKey
- string systemGroupName
- int array serverIds
- boolean add - True to add to the group, False to remove.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 964. Method: create

HTTP **POST**

Description:

Create a new system group.

Parameters:

- string sessionKey
- string name - Name of the system group.
- string description - Description of the system group.

Returns:

- * struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 965. Method: delete

HTTP **POST**

Description:

Delete a system group.

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 966. Method: getDetails

HTTP **GET**

Description:

Retrieve details of a ServerGroup based on it's id

Parameters:

- string sessionKey
- int systemGroupId

Returns:

- * struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 967. Method: getDetails

HTTP **GET**

Description:

Retrieve details of a ServerGroup based on it's name

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- * struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 968. Method: listActiveSystemsInGroup

HTTP **GET**

Description:

Lists active systems within a server group

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- int array server_id

Chapter 969. Method: listAdministrators

HTTP **GET**

Description:

Returns the list of users who can administer the given group. Caller must be a system group admin or an organization administrator.

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- array :
- struct user
 - int "id"
 - string "login"
 - string "login_uc" - upper case version of the login
 - boolean "enabled" - true if user is enabled, false if the user is disabled

Chapter 970. Method: listAllGroups

HTTP **GET**

Description:

Retrieve a list of system groups that are accessible by the logged in user.

Parameters:

- string sessionKey

Returns:

- array :
- struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 971. Method: listAssignedConfigChannels

HTTP **GET**

Description:

List all Configuration Channels assigned to a system group

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- array :
- struct configuration channel information
 - int "id"
 - int "orgId"
 - string "label"
 - string "name"
 - string "description"
 - struct "configChannelType"
- struct configuration channel type information
 - int "id"
 - string "label"
 - string "name"
 - int "priority"

Available since API version: 25

Chapter 972. Method: listAssignedFormuals

HTTP **GET**

Description:

List all Configuration Channels assigned to a system group

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- array :
- struct formula
 - string "name"
 - string "description"
 - string "formula_group"

Available since API version: 25

Chapter 973. Method: listGroupsWithNoAssociatedAdmins

HTTP **GET**

Description:

Returns a list of system groups that do not have an administrator. (who is not an organization administrator, as they have implicit access to system groups) Caller must be an organization administrator.

Parameters:

- string sessionKey

Returns:

- array :
- struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

Chapter 974. Method: listInactiveSystemsInGroup

HTTP **GET**

Description:

Lists inactive systems within a server group using a specified inactivity time.

Parameters:

- string sessionKey
- string systemGroupName
- int daysInactive - Number of days a system must not check in to be considered inactive.

Returns:

- int array server_id

Chapter 975. Method: listInactiveSystemsInGroup

HTTP **GET**

Description:

Lists inactive systems within a server group using the default 1 day threshold.

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- int array server_id

Chapter 976. Method: listSystems

HTTP **GET**

Description:

Return a list of systems associated with this system group. User must have access to this system group.

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- array :
- struct server details
 - int "id" - system ID
 - string "profile_name"
 - string "machine_id"
 - boolean "payg" - Whether the server instance is payg or not
 - string "minion_id"
 - string "base_entitlement" - system's base entitlement label
 - string array "addon_entitlements" - system's addon entitlements labels, currently only 'virtualization_host'
 - boolean "auto_update" - true if system has auto errata updates enabled
 - string "release" - the operating system release (i.e. 4AS, 5Server)
 - string "address1"
 - string "address2"
 - string "city"
 - string "state"
 - string "country"

-
- string "building"
 - string "room"
 - string "rack"
 - string "description"
 - string "hostname"
 - dateTime.iso8601 "last_boot"
 - string "osa_status" - either 'unknown', 'offline', or 'online'
 - boolean "lock_status" - True indicates that the system is locked. False indicates that the system is unlocked.
 - string "virtualization" - virtualization type - for virtual guests only (optional)
 - string "contact_method" - one of the following:
 - default
 - ssh-push
 - ssh-push-tunnel

Chapter 977. Method: listSystemsMinimal

HTTP **GET**

Description:

Return a list of systems associated with this system group. User must have access to this system group.

Parameters:

- string sessionKey
- string systemGroupName

Returns:

- array :
- struct system
 - int "id"
 - string "name"
 - dateTime.iso8601 "last_checkin" - last time server successfully checked in
 - dateTime.iso8601 "created" - server registration time
 - dateTime.iso8601 "last_boot" - last server boot time
 - int "extra_pkg_count" - number of packages not belonging to any assigned channel
 - int "outdated_pkg_count" - number of out-of-date packages

Chapter 978. Method: scheduleApplyErrataToActive

HTTP **POST**

Description:

Schedules an action to apply errata updates to active systems from a group.

Parameters:

- string sessionKey
- string systemGroupName
- int array errataIds

Returns:

- int array actionId

Available since API version: 13.0

Chapter 979. Method: scheduleApplyErrataToActive

HTTP **POST**

Description:

Schedules an action to apply errata updates to active systems from a group at a given date/time.

Parameters:

- string sessionKey
- string systemGroupName
- int array errataIds
- dateTime.iso8601 earliestOccurrence

Returns:

- int array actionId

Available since API version: 13.0

Chapter 980. Method: scheduleApplyErrataToActive

HTTP **POST**

Description:

Schedules an action to apply errata updates to active systems from a group at a given date/time.

Parameters:

- string sessionKey
- string systemGroupName
- int array errataIds
- dateTime.iso8601 earliestOccurrence
- boolean onlyRelevant

Returns:

- int array actionId

Available since API version: 24

Chapter 981. Method: subscribeConfigChannel

HTTP **POST**

Description:

Subscribe given config channels to a system group

Parameters:

- string sessionKey
- string systemGroupName
- string array configChannelLabels

Returns:

- 1 on success, exception on failure

Chapter 982. Method: unsubscribeConfigChannel

HTTP **POST**

Description:

Unsubscribe given config channels to a system group

Parameters:

- string sessionKey
- string systemGroupName
- string array configChannelLabels

Returns:

- 1 on success, exception on failure

Chapter 983. Method: update

HTTP **POST**

Description:

Update an existing system group.

Parameters:

- string sessionKey
- string systemGroupName
- string description

Returns:

- * struct server group
 - int "id"
 - string "name"
 - string "description"
 - int "org_id"
 - int "system_count"

user

Chapter 984. Available methods

- `addAssignedSystemGroup`
- `addAssignedSystemGroups`
- `addDefaultSystemGroup`
- `addDefaultSystemGroups`
- `addRole`
- `create`
- `create`
- `delete`
- `disable`
- `enable`
- `getCreateDefaultSystemGroup`
- `getDetails`
- `listAssignableRoles`
- `listAssignedSystemGroups`
- `listDefaultSystemGroups`
- `listRoles`
- `listUsers`
- `removeAssignedSystemGroup`
- `removeAssignedSystemGroups`
- `removeDefaultSystemGroup`
- `removeDefaultSystemGroups`
- `removeRole`
- `setCreateDefaultSystemGroup`
- `setDetails`
- `setErrataNotifications`

-
- `setReadOnly`
 - `usePamAuthentication`

Chapter 985. Description

User namespace contains methods to access common user functions available from the web user interface.

Namespace:

user

Chapter 986. Method: addAssignedSystemGroup

HTTP **POST**

Description:

Add system group to user's list of assigned system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string sgName
- boolean setDefault - Should system group also be added to user's list of default system groups.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 987. Method: addAssignedSystemGroups

HTTP **POST**

Description:

Add system groups to user's list of assigned system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string array sgNames - server group names
- boolean setDefault - Should system groups also be added to user's list of default system groups.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 988. Method: addDefaultSystemGroup

HTTP **POST**

Description:

Add system group to user's list of default system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string name - server group name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 989. Method: addDefaultSystemGroups

HTTP **POST**

Description:

Add system groups to user's list of default system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string array sgNames - server group names

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 990. Method: addRole

HTTP **POST**

Description:

Adds a role to a user.

Parameters:

- string sessionKey
- string login - User login name to update.
- string role - Role label to add. Can be any of: satellite_admin, org_admin, channel_admin, config_admin, system_group_admin, or activation_key_admin.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 991. Method: create

HTTP **POST**

Description:

Create a new user.

Parameters:

- string sessionKey
- string login - desired login name, will fail if already in use.
- string password
- string firstName
- string lastName
- string email - User's e-mail address.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 992. Method: create

HTTP **POST**

Description:

Create a new user.

Parameters:

- string sessionKey
- string login - desired login name, will fail if already in use.
- string password
- string firstName
- string lastName
- string email - User's e-mail address.
- int usePamAuth - 1 if you wish to use PAM authentication for this user, 0 otherwise.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 993. Method: delete

HTTP **POST**

Description:

Delete a user.

Parameters:

- string sessionKey
- string login - User login name to delete.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 994. Method: disable

HTTP **POST**

Description:

Disable a user.

Parameters:

- string sessionKey
- string login - User login name to disable.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 995. Method: enable

HTTP **POST**

Description:

Enable a user.

Parameters:

- string sessionKey
- string login - User login name to enable.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 996. Method: getCreateDefaultSystemGroup

HTTP **GET**

Description:

Returns the current value of the CreateDefaultSystemGroup setting. If True this will cause there to be a system group created (with the same name as the user) every time a new user is created, with the user automatically given permission to that system group and the system group being set as the default group for the user (so every time the user registers a system it will be placed in that system group by default). This can be useful if different users will administer different groups of servers in the same organization. Can only be called by an org_admin.

Parameters:

- string sessionKey

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 997. Method: getDetails

HTTP **GET**

Description:

Returns the details about a given user.

Parameters:

- string sessionKey
- string login - User's login name.

Returns:

- struct user details
 - string "first_names" - deprecated, use first_name
 - string "first_name"
 - string "last_name"
 - string "email"
 - int "org_id"
 - string "org_name"
 - string "prefix"
 - string "last_login_date"
 - string "created_date"
 - boolean "enabled" - true if user is enabled, false if the user is disabled
 - boolean "use_pam" - true if user is configured to use PAM authentication
 - boolean "read_only" - true if user is readonly
 - boolean "errata_notification" - true if errata e-mail notification is enabled for the user

Chapter 998. Method: listAssignableRoles

HTTP **GET**

Description:

Returns a list of user roles that this user can assign to others.

Parameters:

- string sessionKey

Returns:

- string array (role label)

Chapter 999. Method: listAssignedSystemGroups

HTTP **GET**

Description:

Returns the system groups that a user can administer.

Parameters:

- string sessionKey
- string login - User's login name.

Returns:

- array :
 - struct system group
 - int "id"
 - string "name"
 - string "description"
 - int "system_count"
 - int "org_id" - Organization ID for this system group.

Chapter 1000. Method: listDefaultSystemGroups

HTTP **GET**

Description:

Returns a user's list of default system groups.

Parameters:

- string sessionKey
- string login - User's login name.

Returns:

- array :
 - struct system group
 - int "id"
 - string "name"
 - string "description"
 - int "system_count"
 - int "org_id" - Organization ID for this system group.

Chapter 1001. Method: listRoles

HTTP **GET**

Description:

Returns a list of the user's roles.

Parameters:

- string sessionKey
- string login - User's login name.

Returns:

- string array (role label)

Chapter 1002. Method: listUsers

HTTP **GET**

Description:

Returns a list of users in your organization.

Parameters:

- string sessionKey

Returns:

- array :
- struct user
 - int "id"
 - string "login"
 - string "login_uc" - upper case version of the login
 - boolean "enabled" - true if user is enabled, false if the user is disabled

Chapter 1003. Method: removeAssignedSystemGroup

HTTP **POST**

Description:

Remove system group from the user's list of assigned system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string sgName - server group name
- boolean setDefault - Should system group also be removed from the user's list of default system groups.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1004. Method: removeAssignedSystemGroups

HTTP **POST**

Description:

Remove system groups from a user's list of assigned system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string array sgNames - server group names
- boolean setDefault - Should system groups also be removed from the user's list of default system groups.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1005. Method: removeDefaultSystemGroup

HTTP **POST**

Description:

Remove a system group from user's list of default system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string sgName - server group name

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1006. Method: removeDefaultSystemGroups

HTTP **POST**

Description:

Remove system groups from a user's list of default system groups.

Parameters:

- string sessionKey
- string login - User's login name.
- string array sgNames - server group names

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1007. Method: removeRole

HTTP **POST**

Description:

Remove a role from a user.

Parameters:

- string sessionKey
- string login - User login name to update.
- string role - Role label to remove. Can be any of: satellite_admin, org_admin, channel_admin, config_admin, system_group_admin, or activation_key_admin.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1008. Method: setCreateDefaultSystemGroup

HTTP **POST**

Description:

Sets the value of the createDefaultSystemGroup setting. If True this will cause there to be a system group created (with the same name as the user) every time a new user is created, with the user automatically given permission to that system group and the system group being set as the default group for the user (so every time the user registers a system it will be placed in that system group by default). This can be useful if different users will administer different groups of servers in the same organization. Can only be called by an org_admin.

Parameters:

- string sessionKey
- boolean createDefaultSystemGroup - true if we should automatically create system groups, false otherwise.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1009. Method: setDetails

HTTP **POST**

Description:

Updates the details of a user.

Parameters:

- string sessionKey
- string login - User's login name.
- struct details
 - string "first_names" - deprecated, use first_name
 - string "first_name"
 - string "last_name"
 - string "email"
 - string "prefix"
 - string "password"

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1010. Method: setErrataNotifications

HTTP **POST**

Description:

Enables/disables errata mail notifications for a specific user.

Parameters:

- string sessionKey
- string login - User's login name.
- boolean value - True for enabling errata notifications, False for disabling

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1011. Method: setReadOnly

HTTP **POST**

Description:

Sets whether the target user should have only read-only API access or standard full scale access.

Parameters:

- string sessionKey
- string login - User's login name.
- boolean readOnly - Sets whether the target user should have only read-only API access or standard full scale access.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1012. Method: usePamAuthentication

HTTP **POST**

Description:

Toggles whether or not a user uses PAM authentication or basic #product() authentication.

Parameters:

- string sessionKey
- string login - User's login name.
- int val
 - 1 to enable PAM authentication
 - 0 to disable.

Returns:

- int - 1 on success, exception thrown otherwise.

user.external

Chapter 1013. Available methods

- `createExternalGroupToRoleMap`
- `createExternalGroupToSystemGroupMap`
- `deleteExternalGroupToRoleMap`
- `deleteExternalGroupToSystemGroupMap`
- `getDefaultOrg`
- `getExternalGroupToRoleMap`
- `getExternalGroupToSystemGroupMap`
- `getKeepTemporaryRoles`
- `getUseOrgUnit`
- `listExternalGroupToRoleMaps`
- `listExternalGroupToSystemGroupMaps`
- `setDefaultOrg`
- `setExternalGroupRoles`
- `setExternalGroupSystemGroups`
- `setKeepTemporaryRoles`
- `setUseOrgUnit`

Chapter 1014. Description

If you are using IPA integration to allow authentication of users from an external IPA server (rare) the users will still need to be created in the `#product()` database. Methods in this namespace allow you to configure some specifics of how this happens, like what organization they are created in or what roles they will have. These options can also be set in the web admin interface.

Namespace:

`user.external`

Chapter 1015. Method: createExternalGroupToRoleMap

HTTP **POST**

Description:

Externally authenticated users may be members of external groups. You can use these groups to assign additional roles to the users when they log in. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- string name - Name of the external group. Must be unique.
- string array roles - role - Can be any of: satellite_admin, org_admin (implies all other roles except for satellite_admin), channel_admin, config_admin, system_group_admin, or activation_key_admin.

Returns:

- * struct external group
 - string "name"
 - string array "roles" - role

Chapter 1016. Method: createExternalGroupToSystemGroupMap

HTTP **POST**

Description:

Externally authenticated users may be members of external groups. You can use these groups to give access to server groups to the users when they log in. Can only be called by an org_admin.

Parameters:

- string sessionKey
- string name - Name of the external group. Must be unique.
- string array groupNames - the names of the server groups to grant access to.

Returns:

- * struct external group
 - string "name"
 - string array "groups" - roles

Chapter 1017. Method: deleteExternalGroupToRoleMap

HTTP **POST**

Description:

Delete the role map for an external group. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- string name - Name of the external group.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1018. Method: deleteExternalGroupToSystemGroupMap

HTTP **POST**

Description:

Delete the server group map for an external group. Can only be called by an org_admin.

Parameters:

- string sessionKey
- string name - Name of the external group.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1019. Method: getDefaultOrg

HTTP **GET**

Description:

Get the default org that users should be added in if orgunit from IPA server isn't found or is disabled. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey

Returns:

- int id - ID of the default organization. 0 if there is no default

Chapter 1020. Method: getExternalGroupToRoleMap

HTTP **GET**

Description:

Get a representation of the role mapping for an external group. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- string name - Name of the external group.

Returns:

- * struct external group
 - string "name"
 - string array "roles" - role

Chapter 1021. Method: getExternalGroupToSystemGroupMap

HTTP **GET**

Description:

Get a representation of the server group mapping for an external group. Can only be called by an org_admin.

Parameters:

- string sessionKey
- string name - Name of the external group.

Returns:

- * struct external group
 - string "name"
 - string array "groups" - roles

Chapter 1022. Method: getKeepTemporaryRoles

HTTP **GET**

Description:

Get whether we should keep roles assigned to users because of their IPA groups even after they log in through a non-IPA method. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey

Returns:

- boolean keep - True if we should keep roles after users log in through non-IPA method, false otherwise

Chapter 1023. Method: `getUseOrgUnit`

HTTP `GET`

Description:

Get whether we place users into the organization that corresponds to the "orgunit" set on the IPA server. The orgunit name must match exactly the `#product()` organization name. Can only be called by a `#product()` Administrator.

Parameters:

- string `sessionKey`

Returns:

- boolean `use` - True if we should use the IPA orgunit to determine which organization to create the user in, false otherwise

Chapter 1024. Method: listExternalGroupToRoleMaps

HTTP **GET**

Description:

List role mappings for all known external groups. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey

Returns:

- array :
- struct external group
 - string "name"
 - string array "roles" - role

Chapter 1025. Method: listExternalGroupToSystemGroupMaps

HTTP **GET**

Description:

List server group mappings for all known external groups. Can only be called by an org_admin.

Parameters:

- string sessionKey

Returns:

- array :
- struct external group
 - string "name"
 - string array "groups" - roles

Chapter 1026. Method: setDefaultOrg

HTTP **POST**

Description:

Set the default org that users should be added in if orgunit from IPA server isn't found or is disabled. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- int orgId - ID of the organization to set as the default org. 0 if there should not be a default organization.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1027. Method: setExternalGroupRoles

HTTP **POST**

Description:

Update the roles for an external group. Replace previously set roles with the ones passed in here. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- string name - Name of the external group.
- string array roles - role - Can be any of: satellite_admin, org_admin (implies all other roles except for satellite_admin), channel_admin, config_admin, system_group_admin, or activation_key_admin.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1028. Method: setExternalGroupSystemGroups

HTTP **POST**

Description:

Update the server groups for an external group. Replace previously set server groups with the ones passed in here. Can only be called by an org_admin.

Parameters:

- string sessionKey
- string name - Name of the external group.
- string array groupNames - the names of the server groups to grant access to.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1029. Method: setKeepTemporaryRoles

HTTP **POST**

Description:

Set whether we should keep roles assigned to users because of their IPA groups even after they log in through a non-IPA method. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- boolean keepRoles - True if we should keep roles after users log in through non-IPA method, false otherwise.

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1030. Method: setUseOrgUnit

HTTP **POST**

Description:

Set whether we place users into the organization that corresponds to the "orgunit" set on the IPA server. The orgunit name must match exactly the #product() organization name. Can only be called by a #product() Administrator.

Parameters:

- string sessionKey
- boolean useOrgUnit - true if we should use the IPA orgunit to determine which organization to create the user in, false otherwise.

Returns:

- int - 1 on success, exception thrown otherwise.

virtualhostmanager

Chapter 1031. Available methods

- create
- delete
- getDetail
- getModuleParameters
- listAvailableVirtualHostGathererModules
- listVirtualHostManagers

Chapter 1032. Description

Provides the namespace for the Virtual Host Manager methods.

Namespace:

virtualhostmanager

Chapter 1033. Method: create

HTTP **POST**

Description:

Creates a Virtual Host Manager from given arguments

Parameters:

- string sessionKey
- string label - Virtual Host Manager label
- string moduleName - the name of the Gatherer module
- parameters parameters - additional parameters (credentials, parameters for virtual-host-gatherer)

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1034. Method: delete

HTTP **POST**

Description:

Deletes a Virtual Host Manager with a given label

Parameters:

- string sessionKey
- string label - Virtual Host Manager label

Returns:

- int - 1 on success, exception thrown otherwise.

Chapter 1035. Method: getDetail

HTTP **GET**

Description:

Gets details of a Virtual Host Manager with a given label

Parameters:

- string sessionKey
- string label - Virtual Host Manager label

Returns:

- * struct virtual host manager
 - string "label"
 - int "org_id"
 - string "gatherer_module"
 - struct "configs"

Chapter 1036. Method: getModuleParameters

HTTP **GET**

Description:

Get a list of parameters for a virtual-host-gatherer module. It returns a map of parameters with their typical default values.

Parameters:

- string sessionKey
- string moduleName - The name of the module

Returns:

- map module_params - module parameters

Chapter 1037. Method: listAvailableVirtualHostGathererModules

HTTP **GET**

Description:

List all available modules from virtual-host-gatherer

Parameters:

- string sessionKey

Returns:

- string array moduleName

Chapter 1038. Method: listVirtualHostManagers

HTTP **GET**

Description:

Lists Virtual Host Managers visible to a user

Parameters:

- string sessionKey

Returns:

- array :
- struct virtual host manager
 - string "label"
 - int "org_id"
 - string "gatherer_module"
 - struct "configs"