



Red Hat Quay 3.6

Red Hat Quay API 指南

Red Hat Quay API 指南

Red Hat Quay 3.6 Red Hat Quay API 指南

Red Hat Quay API 指南

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Red_Hat_Quay_API_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

使用 Red Hat Quay API

目录

前言	14
第 1 章 使用 RED HAT QUAY API	15
1.1. 从 QUAY.IO 访问 QUAY API	15
1.2. 创建 OAUTH 访问令牌	15
1.3. 从网页浏览器访问 QUAY API	16
1.4. 从命令行访问 RED HAT QUAY API	16
1.4.1. 获取超级用户信息	16
1.4.2. 使用 API 创建超级用户	17
1.4.3. 列出用量日志	18
1.4.3.1. 分页示例	18
1.4.4. 目录同步	21
1.4.5. 通过 API 创建存储库构建	21
1.4.6. 创建机构机器人	22
1.4.7. 触发构建	22
1.4.8. 创建私有存储库	22
第 2 章 RED HAT QUAY APPLICATION PROGRAMMING INTERFACE(API)	23
2.1. 授权	23
范围	23
2.2. APPSPECIFICTOKENS	23
2.2.1. listAppTokens	23
GET /api/v1/user/apptoken	23
查询参数	23
响应	24
2.2.2. createAppToken	24
POST /api/v1/user/apptoken	24
请求正文模式(application/json)	24
响应	24
2.2.3. revokeAppToken	25
DELETE /api/v1/user/apptoken/{token_uuid}	25
路径参数	25
响应	25
2.2.4. getAppToken	25
GET /api/v1/user/apptoken/{token_uuid}	25
路径参数	25
响应	25
2.3. BUILD	26
2.3.1. getRepoBuildStatus	26
GET /api/v1/repository/{repository}/build/{build_uuid}/status	26
路径参数	26
响应	26
2.3.2. getRepoBuildLogs	27
GET /api/v1/repository/{repository}/build/{build_uuid}/logs	27
路径参数	27
响应	27
2.3.3. cancelRepoBuild	27
DELETE /api/v1/repository/{repository}/build/{build_uuid}	27
路径参数	27
响应	27
2.3.4. getRepoBuild	28

GET /api/v1/repository/{repository}/build/{build_uuid}	28
路径参数	28
响应	28
2.3.5. getRepoBuilds	28
GET /api/v1/repository/{repository}/build/	28
路径参数	29
查询参数	29
响应	29
2.3.6. requestRepoBuild	29
POST /api/v1/repository/{repository}/build/	29
路径参数	29
请求正文模式(application/json)	29
响应	30
2.4. DISCOVERY	30
2.4.1. discovery	30
GET /api/v1/discovery	30
查询参数	31
响应	31
2.5. 错误	31
2.5.1. getErrorDescription	31
GET /api/v1/error/{error_type}	31
路径参数	31
响应	31
2.6. GLOBALMESSAGES	32
2.6.1. getGlobalMessages	32
GET /api/v1/messages	32
响应	32
2.6.2. createGlobalMessage	32
POST /api/v1/messages	32
请求正文模式(application/json)	32
响应	33
2.6.3. deleteGlobalMessage	33
DELETE /api/v1/message/{uuid}	33
路径参数	33
响应	33
2.7. IMAGE	33
2.7.1. getImage	33
GET /api/v1/repository/{repository}/image/{image_id}	34
路径参数	34
响应	34
2.7.2. listRepositoryImages	34
GET /api/v1/repository/{repository}/image/	34
路径参数	34
响应	34
2.8. LOGS	35
2.8.1. getAggregateUserLogs	35
GET /api/v1/user/aggregatelogs	35
查询参数	35
响应	35
2.8.2. exportUserLogs	35
POST /api/v1/user/exportlogs	36
查询参数	36
请求正文模式(application/json)	36

响应	36
2.8.3. listUserLogs	36
GET /api/v1/user/logs	36
查询参数	36
响应	37
2.8.4. getAggregateOrgLogs	37
GET /api/v1/organization/{orgname}/aggregatelogs	37
路径参数	37
查询参数	37
响应	38
2.8.5. exportOrgLogs	38
POST /api/v1/organization/{orgname}/exportlogs	38
路径参数	38
查询参数	38
请求正文模式(application/json)	38
响应	39
2.8.6. listOrgLogs	39
GET /api/v1/organization/{orgname}/logs	39
路径参数	39
查询参数	39
响应	40
2.8.7. getAggregateRepoLogs	40
GET /api/v1/repository/{repository}/aggregatelogs	40
路径参数	40
查询参数	40
响应	40
2.8.8. exportRepoLogs	41
POST /api/v1/repository/{repository}/exportlogs	41
路径参数	41
查询参数	41
请求正文模式(application/json)	41
响应	41
2.8.9. listRepoLogs	42
GET /api/v1/repository/{repository}/logs	42
路径参数	42
查询参数	42
响应	42
2.9. 清单	43
2.9.1. deleteManifestLabel	43
DELETE /api/v1/repository/{repository}/manifest/{manifestref}/labels/{labelid}	43
路径参数	43
响应	43
2.9.2. getManifestLabel	43
GET /api/v1/repository/{repository}/manifest/{manifestref}/labels/{labelid}	44
路径参数	44
响应	44
2.9.3. listManifestLabels	44
GET /api/v1/repository/{repository}/manifest/{manifestref}/labels	44
路径参数	44
查询参数	44
响应	45
2.9.4. addManifestLabel	45
POST /api/v1/repository/{repository}/manifest/{manifestref}/labels	45

路径参数	45
请求正文模式(application/json)	45
响应	45
2.9.5. getRepoManifest	46
GET /api/v1/repository/{repository}/manifest/{manifestref}	46
路径参数	46
响应	46
2.10. MIRROR	46
2.10.1. syncCancel	46
POST /api/v1/repository/{repository}/mirror/sync-cancel	47
路径参数	47
响应	47
2.10.2. syncNow	47
POST /api/v1/repository/{repository}/mirror/sync-now	47
路径参数	47
响应	47
2.10.3. getRepoMirrorConfig	48
GET /api/v1/repository/{repository}/mirror	48
路径参数	48
响应	48
2.10.4. createRepoMirrorConfig	48
POST /api/v1/repository/{repository}/mirror	48
路径参数	48
请求正文模式(application/json)	48
响应	49
2.10.5. changeRepoMirrorConfig	49
PUT /api/v1/repository/{repository}/mirror	50
路径参数	50
请求正文模式(application/json)	50
响应	50
2.11. 机构	51
2.11.1. createOrganization	51
POST /api/v1/organization/	51
请求正文模式(application/json)	51
响应	51
2.11.2. getOrganizationCollaborators	52
GET /api/v1/organization/{orgname}/collaborators	52
路径参数	52
响应	52
2.11.3. deleteOrganizationApplication	52
DELETE /api/v1/organization/{orgname}/applications/{client_id}	52
路径参数	52
响应	53
2.11.4. getOrganizationApplication	53
GET /api/v1/organization/{orgname}/applications/{client_id}	53
路径参数	53
响应	53
2.11.5. updateOrganizationApplication	54
PUT /api/v1/organization/{orgname}/applications/{client_id}	54
路径参数	54
请求正文模式(application/json)	54
响应	54
2.11.6. getOrganizationApplications	55

GET /api/v1/organization/{orgname}/applications	55
路径参数	55
响应	55
2.11.7. createOrganizationApplication	55
POST /api/v1/organization/{orgname}/applications	55
路径参数	55
请求正文模式(application/json)	56
响应	56
2.11.8. removeOrganizationMember	56
DELETE /api/v1/organization/{orgname}/members/{membername}	56
路径参数	56
响应	57
2.11.9. getOrganizationMember	57
GET /api/v1/organization/{orgname}/members/{membername}	57
路径参数	57
响应	57
2.11.10. getOrganizationMembers	58
GET /api/v1/organization/{orgname}/members	58
路径参数	58
响应	58
2.11.11. deleteAdmindedOrganization	58
DELETE /api/v1/organization/{orgname}	58
路径参数	58
响应	58
2.11.12. getOrganization	59
GET /api/v1/organization/{orgname}	59
路径参数	59
响应	59
2.11.13. changeOrganizationDetails	59
PUT /api/v1/organization/{orgname}	59
路径参数	59
请求正文模式(application/json)	60
响应	60
2.11.14. getApplicationInformation	60
GET /api/v1/app/{client_id}	60
路径参数	60
响应	61
2.12. 权限	61
2.12.1. getUserTransitivePermission	61
GET /api/v1/repository/{repository}/permissions/user/{username}/transitive	61
路径参数	61
响应	61
2.12.2. deleteUserPermissions	62
DELETE /api/v1/repository/{repository}/permissions/user/{username}	62
路径参数	62
响应	62
2.12.3. getUserPermissions	62
GET /api/v1/repository/{repository}/permissions/user/{username}	62
路径参数	62
响应	63
2.12.4. changeUserPermissions	63
PUT /api/v1/repository/{repository}/permissions/user/{username}	63
路径参数	63

请求正文模式(application/json)	63
响应	63
2.12.5. deleteTeamPermissions	64
DELETE /api/v1/repository/{repository}/permissions/team/{teamname}	64
路径参数	64
响应	64
2.12.6. getTeamPermissions	64
GET /api/v1/repository/{repository}/permissions/team/{teamname}	64
路径参数	65
响应	65
2.12.7. changeTeamPermissions	65
PUT /api/v1/repository/{repository}/permissions/team/{teamname}	65
路径参数	65
请求正文模式(application/json)	65
响应	66
2.12.8. listRepoTeamPermissions	66
GET /api/v1/repository/{repository}/permissions/team/	66
路径参数	66
响应	66
2.12.9. listRepoUserPermissions	66
GET /api/v1/repository/{repository}/permissions/user/	66
路径参数	67
响应	67
2.13. PROTOTYPE	67
2.13.1. deleteOrganizationPrototypePermission	67
DELETE /api/v1/organization/{orgname}/prototypes/{prototypeid}	67
路径参数	67
响应	67
2.13.2. updateOrganizationPrototypePermission	68
PUT /api/v1/organization/{orgname}/prototypes/{prototypeid}	68
路径参数	68
请求正文模式(application/json)	68
响应	68
2.13.3. getOrganizationPrototypePermissions	69
GET /api/v1/organization/{orgname}/prototypes	69
路径参数	69
响应	69
2.13.4. createOrganizationPrototypePermission	69
POST /api/v1/organization/{orgname}/prototypes	69
路径参数	69
请求正文模式(application/json)	69
响应	70
2.14. 软件仓库	70
2.14.1. listRepos	70
GET /api/v1/repository	70
查询参数	70
响应	71
2.14.2. createRepo	71
POST /api/v1/repository	71
请求正文模式(application/json)	71
响应	72
2.14.3. changeRepoVisibility	72
POST /api/v1/repository/{repository}/changevisibility	72

路径参数	72
请求正文模式(application/json)	72
响应	72
2.14.4. changeRepoState	73
PUT /api/v1/repository/{repository}/changestate	73
路径参数	73
请求正文模式(application/json)	73
响应	73
2.14.5. deleteRepository	74
DELETE /api/v1/repository/{repository}	74
路径参数	74
响应	74
2.14.6. getRepo	74
GET /api/v1/repository/{repository}	74
路径参数	74
查询参数	74
响应	75
2.14.7. updateRepo	75
PUT /api/v1/repository/{repository}	75
路径参数	75
请求正文模式(application/json)	75
响应	75
2.15. REPOSITORYNOTIFICATION	76
2.15.1. testRepoNotification	76
POST /api/v1/repository/{repository}/notification/{uuid}/test	76
路径参数	76
响应	76
2.15.2. deleteRepoNotification	76
DELETE /api/v1/repository/{repository}/notification/{uuid}	76
路径参数	76
响应	77
2.15.3. getRepoNotification	77
GET /api/v1/repository/{repository}/notification/{uuid}	77
路径参数	77
响应	77
2.15.4. resetRepositoryNotificationFailures	78
POST /api/v1/repository/{repository}/notification/{uuid}	78
路径参数	78
响应	78
2.15.5. listRepoNotifications	78
GET /api/v1/repository/{repository}/notification/	78
路径参数	78
响应	78
2.15.6. createRepoNotification	79
POST /api/v1/repository/{repository}/notification/	79
路径参数	79
请求正文模式(application/json)	79
响应	79
2.16. REPTOKEN	80
2.16.1. deleteToken	80
DELETE /api/v1/repository/{repository}/tokens/{code}	80
路径参数	80
响应	80

2.16.2. getTokens	80
GET /api/v1/repository/{repository}/tokens/{code}	81
路径参数	81
响应	81
2.16.3. changeToken	81
PUT /api/v1/repository/{repository}/tokens/{code}	81
路径参数	81
请求正文模式(application/json)	81
响应	82
2.16.4. listRepoTokens	82
GET /api/v1/repository/{repository}/tokens/	82
路径参数	82
响应	82
2.16.5. createToken	83
POST /api/v1/repository/{repository}/tokens/	83
路径参数	83
请求正文模式(application/json)	83
响应	83
2.17. ROBOT	83
2.17.1. getUserRobots	83
GET /api/v1/user/robots	83
查询参数	83
响应	84
2.17.2. getOrgRobotPermissions	84
GET /api/v1/organization/{orgname}/robots/{robot_shortname}/permissions	84
路径参数	84
响应	84
2.17.3. regenerateOrgRobotToken	85
POST /api/v1/organization/{orgname}/robots/{robot_shortname}/regenerate	85
路径参数	85
响应	85
2.17.4. getUserRobotPermissions	85
GET /api/v1/user/robots/{robot_shortname}/permissions	86
路径参数	86
响应	86
2.17.5. regenerateUserRobotToken	86
POST /api/v1/user/robots/{robot_shortname}/regenerate	86
路径参数	86
响应	86
2.17.6. deleteOrgRobot	87
DELETE /api/v1/organization/{orgname}/robots/{robot_shortname}	87
路径参数	87
响应	87
2.17.7. getOrgRobot	87
GET /api/v1/organization/{orgname}/robots/{robot_shortname}	87
路径参数	87
响应	88
2.17.8. createOrgRobot	88
PUT /api/v1/organization/{orgname}/robots/{robot_shortname}	88
路径参数	88
请求正文模式(application/json)	88
响应	89
2.17.9. getOrgRobots	89

GET /api/v1/organization/{orgname}/robots	89
路径参数	89
查询参数	89
响应	89
2.17.10. deleteUserRobot	90
DELETE /api/v1/user/robots/{robot_shortcode}	90
路径参数	90
响应	90
2.17.11. getUserRobot	90
GET /api/v1/user/robots/{robot_shortcode}	90
路径参数	91
响应	91
2.17.12. createUserRobot	91
PUT /api/v1/user/robots/{robot_shortcode}	91
路径参数	91
请求正文模式(application/json)	91
响应	92
2.18. SEARCH	92
2.18.1. conductRepoSearch	92
GET /api/v1/find/repositories	92
查询参数	92
响应	92
2.18.2. conductSearch	93
GET /api/v1/find/all	93
查询参数	93
响应	93
2.18.3. getMatchingEntities	93
GET /api/v1/entities/{prefix}	93
路径参数	93
查询参数	94
响应	94
2.19. SECSCAN	94
2.19.1. getRepoManifestSecurity	94
GET /api/v1/repository/{repository}/manifest/{manifestref}/security	94
路径参数	94
查询参数	94
响应	95
2.19.2. getRepolImageSecurity	95
GET /api/v1/repository/{repository}/image/{imageid}/security	95
路径参数	95
查询参数	95
响应	95
2.20. 超级用户	96
2.20.1. listAllUsers	96
GET /api/v1/superuser/users/	96
查询参数	96
响应	96
2.20.2. createInstallUser	96
POST /api/v1/superuser/users/	97
请求正文模式(application/json)	97
响应	97
2.20.3. listAllLogs	97
GET /api/v1/superuser/logs	97

查询参数	97
响应	98
2.20.4. listServiceKeys	98
GET /api/v1/superuser/keys	98
响应	98
2.20.5. createServiceKey	98
POST /api/v1/superuser/keys	98
请求正文模式(application/json)	98
响应	99
2.20.6. deleteOrganization	99
DELETE /api/v1/superuser/organizations/{name}	99
路径参数	99
响应	99
2.20.7. changeOrganization	100
PUT /api/v1/superuser/organizations/{name}	100
路径参数	100
请求正文模式(application/json)	100
响应	100
2.20.8. approveServiceKey	101
POST /api/v1/superuser/approvedkeys/{kid}	101
路径参数	101
请求正文模式(application/json)	101
响应	101
2.20.9. deleteServiceKey	102
DELETE /api/v1/superuser/keys/{kid}	102
路径参数	102
响应	102
2.20.10. getServiceKey	102
GET /api/v1/superuser/keys/{kid}	102
路径参数	102
响应	102
2.20.11. updateServiceKey	103
PUT /api/v1/superuser/keys/{kid}	103
路径参数	103
请求正文模式(application/json)	103
响应	103
2.20.12. getRepoBuildStatusSuperUser	104
GET /api/v1/superuser/{build_uuid}/status	104
路径参数	104
响应	104
2.20.13. getRepoBuildSuperUser	104
GET /api/v1/superuser/{build_uuid}/build	104
路径参数	104
响应	104
2.20.14. getRepoBuildLogsSuperUser	105
GET /api/v1/superuser/{build_uuid}/logs	105
路径参数	105
响应	105
2.21. TAG	105
2.21.1. restoreTag	105
POST /api/v1/repository/{repository}/tag/{tag}/restore	105
路径参数	106
请求正文模式(application/json)	106

响应	106
2.21.2. listTagImages	106
GET /api/v1/repository/{repository}/tag/{tag}/images	106
路径参数	106
查询参数	107
响应	107
2.21.3. deleteFullTag	107
DELETE /api/v1/repository/{repository}/tag/{tag}	107
路径参数	107
响应	107
2.21.4. changeTag	108
PUT /api/v1/repository/{repository}/tag/{tag}	108
路径参数	108
请求正文模式(application/json)	108
响应	108
2.21.5. listRepoTags	109
GET /api/v1/repository/{repository}/tag/	109
路径参数	109
查询参数	109
响应	109
2.22. TEAM	110
2.22.1. getOrganizationTeamPermissions	110
GET /api/v1/organization/{orgname}/team/{teamname}/permissions	110
路径参数	110
响应	110
2.22.2. deleteOrganizationTeamMember	110
DELETE /api/v1/organization/{orgname}/team/{teamname}/members/{membername}	110
路径参数	111
响应	111
2.22.3. updateOrganizationTeamMember	111
PUT /api/v1/organization/{orgname}/team/{teamname}/members/{membername}	111
路径参数	111
响应	111
2.22.4. getOrganizationTeamMembers	112
GET /api/v1/organization/{orgname}/team/{teamname}/members	112
路径参数	112
查询参数	112
响应	112
2.22.5. deleteOrganizationTeam	113
DELETE /api/v1/organization/{orgname}/team/{teamname}	113
路径参数	113
响应	113
2.22.6. updateOrganizationTeam	113
PUT /api/v1/organization/{orgname}/team/{teamname}	113
路径参数	113
请求正文模式(application/json)	114
响应	114
2.23. TRIGGER	114
2.23.1. activateBuildTrigger	114
POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/activate	114
路径参数	114
请求正文模式(application/json)	115
响应	115

2.23.2. listTriggerRecentBuilds	115
GET /api/v1/repository/{repository}/trigger/{trigger_uuid}/builds	115
路径参数	115
查询参数	115
响应	116
2.23.3. manuallyStartBuildTrigger	116
POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/start	116
路径参数	116
请求正文模式(application/json)	116
响应	117
2.23.4. deleteBuildTrigger	117
DELETE /api/v1/repository/{repository}/trigger/{trigger_uuid}	117
路径参数	117
响应	117
2.23.5. getBuildTrigger	118
GET /api/v1/repository/{repository}/trigger/{trigger_uuid}	118
路径参数	118
响应	118
2.23.6. updateBuildTrigger	118
PUT /api/v1/repository/{repository}/trigger/{trigger_uuid}	118
路径参数	118
请求正文模式(application/json)	119
响应	119
2.23.7. listBuildTriggers	119
GET /api/v1/repository/{repository}/trigger/	119
路径参数	119
响应	119
2.24. USER	120
2.24.1. listStarredRepos	120
GET /api/v1/user/starred	120
查询参数	120
响应	120
2.24.2. createStar	120
POST /api/v1/user/starred	120
请求正文模式(application/json)	121
响应	121
2.24.3. getLoggedInUser	121
GET /api/v1/user/	121
响应	121
2.24.4. deleteStar	121
DELETE /api/v1/user/starred/{repository}	122
路径参数	122
响应	122
2.24.5. getUserInformation	122
GET /api/v1/users/{username}	122
路径参数	122
响应	122
2.25. 定义	123
2.25.1. ApiError	123
2.25.2. UserView	123
2.25.3. ViewMirrorConfig	124
2.25.4. ApiErrorDescription	125

前言

Red Hat Quay 应用程序编程接口(API)是一个 OAuth 2 RESTful API, 它由一组用于添加、显示、修改和删除 Red Hat Quay 功能的端点组成。本指南描述了这些端点, 并显示了用于访问它们的命令和基于浏览器的示例。

第1章 使用 RED HAT QUAY API

Red Hat Quay 提供了一个完整的 OAuth 2、RESTful API，它：

- 可以从 URL <https://<yourquayhost>/api/v1>的端点获取每个 Red Hat Quay 实例的端点
- 通过启用 Swagger UI，通过浏览器连接到端点，以获取、删除、发布和放置 Red Hat Quay 设置
- 可以被使用 API 调用和使用 OAuth 令牌的应用程序访问
- 以 JSON 格式发送和接收数据

下面的文本描述了如何访问 Red Hat Quay API，并使用它来查看和修改 Red Hat Quay 集群中的设置。下一节列出了和描述 API 端点。

1.1. 从 QUAY.IO 访问 QUAY API

如果您还没有运行自己的 Red Hat Quay 集群，您可以从 Web 浏览器浏览 Quay.io 提供的 Red Hat Quay API：

```
https://docs.quay.io/api/swagger/
```

显示 Quay.io API 端点的 API Explorer。您将无法看到在 Quay.io 上未启用的 Red Hat Quay 功能的超级用户 API 端点或端点（如存储库镜像）。

从 API Explorer，您可以获得并有时会更改以下信息：

- 账单、订阅和计划
- 存储库构建和构建触发器
- 错误消息和全局信息
- 仓库镜像、清单、权限、通知、漏洞和镜像签名
- 使用日志
- 机构、成员和 OAuth 应用程序
- 用户和机器人帐户
- and more...

选择打开端点来查看端点的每一个部分的 Model Schema。打开端点，输入所有必要的参数（如存储库名称或镜像），然后选择 尝试退出！按钮来查询或更改与 Quay.io 端点关联的设置。

1.2. 创建 OAUTH 访问令牌

要创建 OAuth 访问令牌，以便您可以访问机构的 API：

1. 登录 Red Hat Quay 并选择您的机构（或创建一个新的机构）。
2. 点击左侧导航栏中的 Applications 图标。
3. 选择 Create New Application 并在出现提示时为新应用程序提供名称。

4. 选择新应用。
5. 从左侧导航中选择 Generate Token。
6. 选择设置令牌范围的复选框，然后选择 Generate Access Token。
7. 查看允许并选择授权应用程序的权限来批准它。
8. 复制新生成的令牌，以用于访问 API。

1.3. 从网页浏览器访问 QUAY API

通过启用 Swagger，您可以通过网页浏览器访问您自己的 Red Hat Quay 实例的 API。这个 URL 通过 Swagger UI 和这个 URL 公开 Red Hat Quay API explorer：

```
https://<yourquayhost>/api/v1/discovery.
```

这样，访问 API 的方法不包括在 Red Hat Quay 安装中可用的超级用户端点。以下是通过运行 swagger-ui 容器镜像访问本地系统上运行的 Red Hat Quay API 接口的示例：

```
# export SERVER_HOSTNAME=<yourhostname>
# sudo podman run -p 8888:8080 -e API_URL=https://$SERVER_HOSTNAME:8443/api/v1/discovery
docker.io/swaggerapi/swagger-ui
```

在运行 swagger-ui 容器时，打开 Web 浏览器到 localhost 端口 8888，以通过 swagger-ui 容器查看 API 端点。

为避免在日志中出现错误，如 "API 调用，如果使用浏览器调用 X-Requested-With 标头来调用，"如果从浏览器中调用，请将下面这一行添加到集群中的所有节点上的 **config.yaml** 中，再重启 Red Hat Quay：

```
BROWSER_API_CALLS_XHR_ONLY: false
```

1.4. 从命令行访问 RED HAT QUAY API

您可以通过 Red Hat Quay 集群的 API，使用 **curl** 命令 GET、PUT、POST 或 DELETE 设置。将 **<token>** 替换为您之前创建的 OAuth 访问令牌，以便在以下示例中获取或更改设置。

1.4.1. 获取超级用户信息

```
$ curl -X GET -H "Authorization: Bearer <token_here>" \
  "https://<yourquayhost>/api/v1/superuser/users/"
```

例如：

```
$ curl -X GET -H "Authorization: Bearer mFCdgS7SAIoMcnTsHCGx23vcNsTgziAa4CmmHlsg"
http://quay-server:8080/api/v1/superuser/users/ | jq

{
  "users": [
    {
      "kind": "user",
      "name": "quayadmin",
      "username": "quayadmin",
```

```

"email": "quayadmin@example.com",
"verified": true,
"avatar": {
  "name": "quayadmin",
  "hash": "357a20e8c56e69d6f9734d23ef9517e8",
  "color": "#5254a3",
  "kind": "user"
},
"super_user": true,
"enabled": true
}
]
}

```

1.4.2. 使用 API 创建超级用户

- 配置超级用户名称，如 Deploy Quay book 所述：
 - 使用配置编辑器 UI 或者
 - 直接使用配置 API 来验证（并下载）更新的配置捆绑包，直接编辑 **config.yaml** 文件
- 为超级用户名称创建用户帐户：
 - 获取授权令牌（如上所述），并使用 **curl** 来创建用户：

```

$ curl -H "Content-Type: application/json" -H "Authorization: Bearer
Fava2kV9C92p1eXnMawBZx9vTqVnksvwNm0ckFKZ" -X POST --data '{
  "username": "quaysuper",
  "email": "quaysuper@example.com"
}' http://quay-server:8080/api/v1/superuser/users/ | jq

```

- 返回的内容包括新用户帐户的生成密码：

```

{
  "username": "quaysuper",
  "email": "quaysuper@example.com",
  "password": "EH67NB3Y6PTBED8H0HC6UVHGGGA3ODSE",
  "encrypted_password":
  "fn37AZAUQH0PTsU+vlO9IS0QxPW9A/boXL4ovZjIFtUPrBz9i4j9UDOqMjuxQ/0HTfy38go
KEpG8zYXVeQh3IOFzuOjSvKic2Vq7xdtQsU="
}

```

现在，当您请求用户列表时，它将以超级用户身份显示 **quaysuper**：

```

$ curl -X GET -H "Authorization: Bearer mFCdgS7SAIoMcnTsHCGx23vcNsTgziAa4CmmHlsg"
http://quay-server:8080/api/v1/superuser/users/ | jq

{
  "users": [
    {
      "kind": "user",
      "name": "quayadmin",
      "username": "quayadmin",
      "email": "quayadmin@example.com",

```

```

    "verified": true,
    "avatar": {
      "name": "quayadmin",
      "hash": "357a20e8c56e69d6f9734d23ef9517e8",
      "color": "#5254a3",
      "kind": "user"
    },
    "super_user": true,
    "enabled": true
  },
  {
    "kind": "user",
    "name": "quaysuper",
    "username": "quaysuper",
    "email": "quaysuper@example.com",
    "verified": true,
    "avatar": {
      "name": "quaysuper",
      "hash": "c0e0f155afcef68e58a42243b153df08",
      "color": "#969696",
      "kind": "user"
    },
    "super_user": true,
    "enabled": true
  }
]
}

```

1.4.3. 列出用量日志

一个内部 API `/api/v1/superuser/logs` 可用于列出当前系统的用量日志。这将分页结果，因此在以下示例中创建了超过 20 个仓库，以介绍如何使用多个调用来访问整个结果集。

1.4.3.1. 分页示例

首次调用

```

$ curl -X GET -k -H "Authorization: Bearer qz9NZ2Np1f55CSZ3RVOvxjeUdkzYuCp0pKggABCD"
https://example-registry-quay-quay-enterprise.apps.example.com/api/v1/superuser/logs | jq

```

初始输出

```

{
  "start_time": "Sun, 12 Dec 2021 11:41:55 -0000",
  "end_time": "Tue, 14 Dec 2021 11:41:55 -0000",
  "logs": [
    {
      "kind": "create_repo",
      "metadata": {
        "repo": "t21",
        "namespace": "namespace1"
      },
      "ip": "10.131.0.13",
      "datetime": "Mon, 13 Dec 2021 11:41:16 -0000",

```

```

"performer": {
  "kind": "user",
  "name": "user1",
  "is_robot": false,
  "avatar": {
    "name": "user1",
    "hash": "5d40b245471708144de9760f2f18113d75aa2488ec82e12435b9de34a6565f73",
    "color": "#ad494a",
    "kind": "user"
  }
},
"namespace": {
  "kind": "org",
  "name": "namespace1",
  "avatar": {
    "name": "namespace1",
    "hash": "6cf18b5c19217bfc6df0e7d788746ff7e8201a68cba333fca0437e42379b984f",
    "color": "#e377c2",
    "kind": "org"
  }
}
},
{
  "kind": "create_repo",
  "metadata": {
    "repo": "t20",
    "namespace": "namespace1"
  },
  "ip": "10.131.0.13",
  "datetime": "Mon, 13 Dec 2021 11:41:05 -0000",
  "performer": {
    "kind": "user",
    "name": "user1",
    "is_robot": false,
    "avatar": {
      "name": "user1",
      "hash": "5d40b245471708144de9760f2f18113d75aa2488ec82e12435b9de34a6565f73",
      "color": "#ad494a",
      "kind": "user"
    }
  },
  "namespace": {
    "kind": "org",
    "name": "namespace1",
    "avatar": {
      "name": "namespace1",
      "hash": "6cf18b5c19217bfc6df0e7d788746ff7e8201a68cba333fca0437e42379b984f",
      "color": "#e377c2",
      "kind": "org"
    }
  }
},
...
{
  "kind": "create_repo",

```

```

"metadata": {
  "repo": "t2",
  "namespace": "namespace1"
},
"ip": "10.131.0.13",
"datetime": "Mon, 13 Dec 2021 11:25:17 -0000",
"performer": {
  "kind": "user",
  "name": "user1",
  "is_robot": false,
  "avatar": {
    "name": "user1",
    "hash": "5d40b245471708144de9760f2f18113d75aa2488ec82e12435b9de34a6565f73",
    "color": "#ad494a",
    "kind": "user"
  }
},
"namespace": {
  "kind": "org",
  "name": "namespace1",
  "avatar": {
    "name": "namespace1",
    "hash": "6cf18b5c19217bfc6df0e7d788746ff7e8201a68cba333fca0437e42379b984f",
    "color": "#e377c2",
    "kind": "org"
  }
}
}
],
"next_page":
"gAAAAABhtzGDsH38x7pjWhD8MJq1_2FAgqUw2X9S2LoCLNPH65QJqB4XAU2qAxYb6QqtlcWj9eI6
DUiMN_q3e3I0agCvB2VPQ8rY75WeaiUzM3rQIMc4i6EIR78t8oUxVfNp1RMPiRQYYZyXP9h6E8LZZhq
TMs0S-SedaQJ3kVFtkxZqJwHVjgt23Ts2DonVoYwtKgl3bCC5"
}

```

使用 `next_page` 进行第二个调用

```

$ curl -X GET -k -H "Authorization: Bearer qz9NZ2Np1f55CSZ3RVOvxjeUdkzYuCp0pKggABCD"
https://example-registry-quay-quay-enterprise.apps.example.com/api/v1/superuser/logs?
next_page=gAAAAABhtzGDsH38x7pjWhD8MJq1_2FAgqUw2X9S2LoCLNPH65QJqB4XAU2qAxYb6Q
qtlcWj9eI6DUiMN_q3e3I0agCvB2VPQ8rY75WeaiUzM3rQIMc4i6EIR78t8oUxVfNp1RMPiRQYYZyXP9h
6E8LZZhqTMs0S-SedaQJ3kVFtkxZqJwHVjgt23Ts2DonVoYwtKgl3bCC5 | jq

```

第二个调用的输出

```

{
  "start_time": "Sun, 12 Dec 2021 11:42:46 -0000",
  "end_time": "Tue, 14 Dec 2021 11:42:46 -0000",
  "logs": [
    {
      "kind": "create_repo",
      "metadata": {
        "repo": "t1",
        "namespace": "namespace1"
      },

```

```

    "ip": "10.131.0.13",
    "datetime": "Mon, 13 Dec 2021 11:25:07 -0000",
    "performer": {
      "kind": "user",
      "name": "user1",
      "is_robot": false,
      "avatar": {
        "name": "user1",
        "hash": "5d40b245471708144de9760f2f18113d75aa2488ec82e12435b9de34a6565f73",
        "color": "#ad494a",
        "kind": "user"
      }
    },
    "namespace": {
      "kind": "org",
      "name": "namespace1",
      "avatar": {
        "name": "namespace1",
        "hash": "6cf18b5c19217bfc6df0e7d788746ff7e8201a68cba333fca0437e42379b984f",
        "color": "#e377c2",
        "kind": "org"
      }
    }
  },
  ...
]
}

```

1.4.4. 目录同步

要在机构 **testadminorg** 中为团队 **newteam** 启用目录同步，其中 LDAP 中对应的组名称是 **ldapgroup**：

```

$ curl -X POST -H "Authorization: Bearer 9rJYBR3v3pXcj5XqlA2XX6Thk4gld4TCYLLWDF" \
-H "Content-type: application/json" \
-d '{"group_dn": "cn=ldapgroup,ou=Users"}' \
http://quay1-server:8080/api/v1/organization/testadminorg/team/newteam/syncing

```

为同一团队禁用同步：

```

$ curl -X DELETE -H "Authorization: Bearer 9rJYBR3v3pXcj5XqlA2XX6Thk4gld4TCYLLWDF" \
http://quay1-server:8080/api/v1/organization/testadminorg/team/newteam/syncing

```

1.4.5. 通过 API 创建存储库构建

要从指定的输入构建存储库，并使用自定义标签标记构建，用户可以使用 `requestRepoBuild` 端点。它采用以下数据：

```

{
  "docker_tags": [
    "string"
  ],
  "pull_robot": "string",

```

```
"subdirectory": "string",
"archive_url": "string"
}
```

archive_url 参数应指向包含 Dockerfile 和其他构建所需文件的 **tar** 或 **zip** 存档。**file_id** 参数是我们较旧的构建系统外。它无法再使用。如果 Dockerfile 位于子目录中，也需要指定它。

该存档应公开访问。OAuth 应用应具有 "Administer Organization" 范围，因为只有机构管理员可以访问机器人的帐户令牌。否则，某人可以通过简单地授予构建对机器人（没有访问权限本身）来获取机器人权限，并使用它来获取镜像内容。如果出现错误，请检查 json 块返回的，并确保归档位置、拉取机器人和其他参数正确传递。点单个构建页面右上角的"下载日志"检查日志以了解更详细的消息。

1.4.6. 创建机构机器人

```
$ curl -X PUT https://quay.io/api/v1/organization/{orgname}/robots/{robot shortname} \
-H 'Authorization: Bearer <token>'
```

1.4.7. 触发构建

```
$ curl -X POST https://quay.io/api/v1/repository/YOURORNAME/YOURREPONAME/build/ \
-H 'Authorization: Bearer <token>'
```

带有请求的 Python

```
import requests
r = requests.post('https://quay.io/api/v1/repository/example/example/image', headers={'content-type':
'application/json', 'Authorization': 'Bearer <redacted>'}, data=[[<request-body-contents>])
print(r.text)
```

1.4.8. 创建私有存储库

```
$ curl -X POST https://quay.io/api/v1/repository \
-H 'Authorization: Bearer {token}' \
-H 'Content-Type: application/json' \
-d '{"namespace": "yournamespace", "repository": "yourreponame",
"description": "descriptionofyourrepo", "visibility": "private"}' | jq
```

第 2 章 RED HAT QUAY APPLICATION PROGRAMMING INTERFACE(API)

此 API 允许您执行处理 Red Hat Quay 存储库、用户和组织所需的许多操作。

2.1. 授权

oauth2_implicit

范围

以下范围用于控制对 API 端点的访问：

影响范围	描述
repo:read	此应用程序可以查看并拉取授予用户或机器人帐户可见的所有存储库
repo:write	此应用程序将可以查看、推送和拉取授予用户或机器人帐户具有写入权限的所有存储库
repo:admin	此应用程序具有管理员访问权限，可以访问授予用户或机器人帐户有权访问的所有存储库
repo:create	此应用将能够创建存储库到允许授予用户或机器人帐户的任何命名空间创建存储库
user:read	此应用将能够读取用户信息，如用户名和电子邮件地址。
org:admin	此应用程序将可以管理您的组织，包括创建机器人、创建团队、调整团队成员资格和更改账单设置。在授予此权限前，在请求的应用程序中应有绝对信任。
super:user	此应用程序将可以管理您的安装，包括管理用户、管理超级用户面板中找到的机构和其他功能。在授予此权限前，在请求的应用程序中应有绝对信任。
user:admin	此应用程序可以管理您的帐户，包括创建机器人并授予它们对存储库的权限。在授予此权限前，在请求的应用程序中应有绝对信任。

2.2. APPSPECIFICTOKENS

为当前用户管理应用特定令牌。

2.2.1. listAppTokens

列出用户的应用特定令牌。

GET /api/v1/user/apptoken

Authorizations: oauth2_implicit(user:admin)

查询参数

类型	Name	描述	模式
----	------	----	----

类型	Name	描述	模式
query	过期 可选	如果为 true, 则只会很快返回这些令牌	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.2.2. createAppToken

为用户创建新的应用特定令牌。

POST /api/v1/user/apptoken

Authorizations: `oauth2_implicit(user:admin)`

请求正文模式(`application/json`)

新令牌的描述。

名称	描述	模式
<code>friendlyName</code> 可选	帮助识别令牌的友好名称	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.2.3. revokeAppToken

为用户撤销特定的应用令牌。

DELETE /api/v1/user/apptoken/{token_uuid}
Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	token_uuid 是必需的	应用程序特定令牌的 uuid	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.2.4. getAppToken

为该用户返回特定的应用令牌。

GET /api/v1/user/apptoken/{token_uuid}
Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	token_uuid 是必需的	应用程序特定令牌的 uuid	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3. BUILD

创建、列出、取消并获取存储库构建的状态/日志。

2.3.1. getRepoBuildStatus

返回构建 uuids 指定的构建的状态。

GET /api/v1/repository/{repository}/build/{build_uuid}/status

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	build_uuid <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.2. getRepoBuildLogs

返回构建 uuid 指定构建的构建日志。

GET /api/v1/repository/{repository}/build/{build_uuid}/logs

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	build_uuid <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.3. cancelRepoBuild

取消存储库构建。

DELETE /api/v1/repository/{repository}/build/{build_uuid}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	build_uuid <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.4. getRepoBuild

返回有关构建的信息。

GET /api/v1/repository/{repository}/build/{build_uuid}

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	build_uuid <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.5. getRepoBuilds

获取存储库构建列表。

GET /api/v1/repository/{repository}/build/

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	因为 <i>可选</i>	返回自给定 unix 时间码以来的所有构建	整数
query	限制 <i>可选</i>	要返回的最大构建数	整数

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.6. requestRepoBuild

请求从指定的输入来构建和推送存储库。

POST /api/v1/repository/{repository}/build/
Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)
 新存储库构建的描述。

名称	描述	模式
file_id <i>可选</i>	构建 spec 上传时生成的文件 id	字符串
archive_url <i>可选</i>	要构建的 .tar.gz 的 URL。必须以 "http" 或 "https" 开始。	字符串
子目录 <i>可选</i>	Dockerfile 可在其中找到 Dockerfile 的子目录。您只能指定这个或 dockerfile_path	字符串
dockerfile_path <i>optional</i>	到 dockerfile 的路径。您只能指定这个或子目录。	字符串
上下文 <i>可选</i>	传递 dockerfile 的上下文。这是可选的。	字符串
pull_robot <i>optional</i>	用作拉取凭证的 Quay 机器人帐户的用户名	字符串
docker_tags <i>optional</i>	构建镜像要推送到的标签。如果没有指定，则使用"latest"。	字符串数组 非空 唯一

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.4. DISCOVERY

API 发现信息。

2.4.1. discovery

列出以 swagger API 格式提供的所有 API 端点。

GET /api/v1/discovery

授权：

查询参数

类型	Name	描述	模式
query	内部 可选	是否包含内部 API。	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.5. 错误

错误详情 API。

2.5.1. getErrorDescription

获取有关错误的详细描述。

GET /api/v1/error/{error_type}

授权：

路径参数

类型	Name	描述	模式
path	error_type required	识别错误类型的错误代码。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	ApiErrorDescription
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.6. GLOBALMESSAGES

消息 API.

2.6.1. getGlobalMessages

返回超级用户信息。

GET /api/v1/messages

授权：

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.6.2. createGlobalMessage

创建一个消息。

POST /api/v1/messages

Authorizations: oauth2_implicit(super:user)

请求正文模式(application/json)

创建新信息

名称	描述	模式
message 可选	单个消息	对象

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.6.3. deleteGlobalMessage

删除消息。

DELETE /api/v1/message/{uuid}

Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	<i>必需的</i> UUID		字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.7. IMAGE

列出和查找存储库镜像。

2.7.1. getImage

获取可供指定镜像使用的信息。

GET /api/v1/repository/{repository}/image/{image_id}**Authorizations:** oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	image_id <i>required</i>	Docker 镜像 ID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.7.2. listRepositoryImages

列出指定存储库的镜像。

GET /api/v1/repository/{repository}/image/**Authorizations:** oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.8. LOGS

访问机构或存储库的使用情况日志。

2.8.1. getAggregateUserLogs

返回当前用户的聚合日志。

GET /api/v1/user/aggregatelogs

Authorizations: oauth2_implicit(user:admin)

查询参数

类型	Name	描述	模式
query	performer <i>可选</i>	过滤日志的用户名。	字符串
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的"%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.2. exportUserLogs

返回当前用户的聚合日志。

POST /api/v1/user/exportlogs**Authorizations:** oauth2_implicit(user:admin)

查询参数

类型	Name	描述	模式
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的"%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

请求正文模式(application/json)

导出日志操作配置

名称	描述	模式
callback_url <i>可选</i>	使用到导出日志的链接调用的回调 URL	字符串
callback_email <i>可选</i>	电子邮件地址，指向导出的日志的链接	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.3. listUserLogs

列出当前用户的日志。

GET /api/v1/user/logs**Authorizations:** oauth2_implicit(user:admin)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页页面的令牌	字符串
query	performer <i>可选</i>	过滤日志的用户名。	字符串
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的 "%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的 "%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.4. getAggregateOrgLogs

获取指定机构的聚合日志。

GET /api/v1/organization/{orgname}/aggregatelogs

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

查询参数

类型	Name	描述	模式
query	performer <i>可选</i>	过滤日志的用户名。	字符串

类型	Name	描述	模式
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的"%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.5. exportOrgLogs

导出指定组织的日志。

POST /api/v1/organization/{orgname}/exportlogs

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

查询参数

类型	Name	描述	模式
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的"%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

请求正文模式(application/json)

导出日志操作配置

名称	描述	模式
<code>callback_url</code> <i>可选</i>	使用到导出日志的链接调用的回调 URL	字符串
<code>callback_email</code> <i>可选</i>	电子邮件地址，指向导出的日志的链接	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.6. listOrgLogs

列出指定机构的日志。

GET /api/v1/organization/{orgname}/logs
Authorizations: `oauth2_implicit(org:admin)`

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

查询参数

类型	Name	描述	模式
query	<code>next_page</code> <i>optional</i>	下一页页面的页面令牌	字符串
query	<code>performer</code> <i>可选</i>	过滤日志的用户名。	字符串
query	<code>ENDTIME</code> <i>可选</i>	日志的最新时间。格式：UTC 中的 "%m/%d/%Y"	字符串

类型	Name	描述	模式
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.7. getAggregateRepoLogs

返回指定存储库的聚合日志。

GET /api/v1/repository/{repository}/aggregatelogs
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的"%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的"%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.8. exportRepoLogs

为指定存储库对日志进行队列导出。

POST /api/v1/repository/{repository}/exportlogs

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	ENDTIME <i>可选</i>	日志的最新时间。格式：UTC 中的 "%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的 "%m/%d/%Y"	字符串

请求正文模式(application/json)

导出日志操作配置

名称	描述	模式
callback_url <i>可选</i>	使用到导出日志的链接调用的回调 URL	字符串
callback_email <i>可选</i>	电子邮件地址，指向导出的日志的链接	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.9. listRepoLogs

列出指定存储库的日志。

GET /api/v1/repository/{repository}/logs

Authorizations: `oauth2_implicit(repo:admin)`

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页页面的页面令牌	字符串
query	ENDTIME <i>可选</i>	日志的更新时间。格式：UTC 中的 "%m/%d/%Y"	字符串
query	startTime <i>可选</i>	日志的早期时间。格式：UTC 中的 "%m/%d/%Y"	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.9. 清单

管理存储库的清单。

2.9.1. deleteManifestLabel

从清单中删除现有标签。

DELETE /api/v1/repository/{repository}/manifest/{manifestref}/labels/{labelid}

Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	labelid 必需	标签的 ID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref 需要	清单摘要	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.2. getManifestLabel

检索带有清单下特定 ID 的标签。

GET /api/v1/repository/{repository}/manifest/{manifestref}/labels/{labelid}

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	labelid 必需	标签的 ID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref 需要	清单摘要	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.3. listManifestLabels

GET /api/v1/repository/{repository}/manifest/{manifestref}/labels

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref 需要	清单摘要	字符串

查询参数

类型	Name	描述	模式
query	过滤 可选	如果指定，则仅返回与给定前缀匹配的标签	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.4. addManifestLabel

将新标签添加到标签清单中。

POST /api/v1/repository/{repository}/manifest/{manifestref}/labels

Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref 需要	清单摘要	字符串

请求正文模式(application/json)

为清单添加标签

名称	描述	模式
key 可选	标签的密钥	字符串
value 可选	标签的值	字符串
media_type 可选	此标签的介质类型	

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.5. getRepoManifest

GET /api/v1/repository/{repository}/manifest/{manifestref}

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref 需要	清单摘要	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10. MIRROR

2.10.1. syncCancel

为给定存储库的镜像配置更新 sync_status。

POST /api/v1/repository/{repository}/mirror/sync-cancel**Authorizations:** oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10.2. syncNow

为给定存储库的镜像配置更新 sync_status。

POST /api/v1/repository/{repository}/mirror/sync-now**Authorizations:** oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.10.3. getRepoMirrorConfig

返回给定存储库的 Mirror 配置。

GET /api/v1/repository/{repository}/mirror
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	ViewMirrorConfig
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10.4. createRepoMirrorConfig

为给定的存储库创建一个 RepoMirrorConfig。

POST /api/v1/repository/{repository}/mirror
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式([application/json](#))
 创建存储库镜像配置。

名称	描述	模式
is_enabled <i>可选</i>	用于启用或禁用同步。	布尔值
external_reference <i>可选</i>	外部存储库的位置。	字符串
external_registry_username <i>可选</i>	用于与外部 registry 进行身份验证的用户名。	
external_registry_password <i>可选</i>	用于与外部 registry 进行身份验证的密码。	
sync_start_date <i>optional</i>	下次确定此存储库准备好同步的时间。	字符串
sync_interval <i>可选</i>	next_start_date 开始同步后的秒数。	整数
robot_username <i>optional</i>	用于镜像推送的机器人的用户名。	字符串
root_rule <i>optional</i>	用于决定应同步哪些标签的 glob-patterns 列表。	对象
external_registry_config <i>optional</i>		对象

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10.5. changeRepoMirrorConfig

允许用户修改存储库的镜像配置。

PUT /api/v1/repository/{repository}/mirror

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)

更新存储库镜像配置。

名称	描述	模式
is_enabled <i>可选</i>	用于启用或禁用同步。	布尔值
external_reference <i>可选</i>	外部存储库的位置。	字符串
external_registry_username <i>可选</i>	用于与外部 registry 进行身份验证的用户名。	
external_registry_password <i>可选</i>	用于与外部 registry 进行身份验证的密码。	
sync_start_date <i>optional</i>	下次确定此存储库准备好同步的时间。	字符串
sync_interval <i>可选</i>	next_start_date 开始同步后的秒数。	整数
robot_username <i>optional</i>	用于镜像推送的机器人的用户名。	字符串
root_rule <i>optional</i>	用于决定应同步哪些标签的 glob-patterns 列表。	对象
external_registry_config <i>optional</i>		对象

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11. 机构

管理组织、成员和 OAuth 应用程序。

2.11.1. createOrganization

创建新机构。

**POST /api/v1/organization/
Authorizations:** `oauth2_implicit(user:admin)`

请求正文模式(`application/json`)
新组织的描述。

名称	描述	模式
name <i>可选</i>	机构用户名	字符串
email <i>可选</i>	机构联系电子邮件	字符串
recaptcha_response <i>可选</i>	(may be disabled)recaptcha 响应代码进行验证	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.2. getOrganizationCollaborators

列出指定机构外协作器。

GET /api/v1/organization/{orgname}/collaborators
 Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.3. deleteOrganizationApplication

删除此机构下的应用程序。

DELETE /api/v1/organization/{orgname}/applications/{client_id}
 Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

类型	Name	描述	模式
path	client_id <i>required</i>	OAuth 客户端 ID	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.4. getOrganizationApplication

使用指定的 client_id 在指定的机构下检索应用程序。

GET /api/v1/organization/{orgname}/applications/{client_id}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串
path	client_id <i>required</i>	OAuth 客户端 ID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.11.5. updateOrganizationApplication

更新此机构下的应用程序。

PUT /api/v1/organization/{orgname}/applications/{client_id}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	client_id <i>required</i>	OAuth 客户端 ID	字符串

请求正文模式(application/json)

更新应用程序的描述。

名称	描述	模式
name <i>可选</i>	应用程序的名称	字符串
redirect_uri <i>optional</i>	应用程序的 OAuth 重定向的 URI	字符串
application_uri <i>可选</i>	应用程序主页的 URI	字符串
description <i>可选</i>	应用程序的人类可读描述	字符串
avatar_email <i>可选</i>	用于应用程序的 avatar 的电子邮件地址	字符串

响应

HTTP 代码	描述	模式
200	成功调用	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.6. getOrganizationApplications

列出指定机构的应用程序。

GET /api/v1/organization/{orgname}/applications

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.7. createOrganizationApplication

在此组织下创建新应用。

POST /api/v1/organization/{orgname}/applications

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

请求正文模式(application/json)
新机构应用程序的描述。

名称	描述	模式
name <i>可选</i>	应用程序的名称	字符串
redirect_uri <i>optional</i>	应用程序的 OAuth 重定向的 URI	字符串
application_uri <i>可选</i>	应用程序主页的 URI	字符串
description <i>可选</i>	应用程序的人类可读描述	字符串
avatar_email <i>可选</i>	用于应用程序的 avatar 的电子邮件地址	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.8. removeOrganizationMember

从组织中删除成员，撤销其所有存储库功能，并从机构中的所有团队中删除它。

DELETE /api/v1/organization/{orgname}/members/{membername}
Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串
path	membername 必需	机构成员的用户名	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.9. getOrganizationMember

检索组织成员的详细信息。

GET /api/v1/organization/{orgname}/members/{membername}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串
path	membername 必需	机构成员的用户名	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.11.10. getOrganizationMembers

列出指定机构的人类。

GET /api/v1/organization/{orgname}/members

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.11. deleteAdmindedOrganization

删除指定的机构。

DELETE /api/v1/organization/{orgname}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.12. getOrganization

获取指定机构的详情。

GET /api/v1/organization/{orgname}

授权：

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.13. changeOrganizationDetails

更改指定机构的详情。

PUT /api/v1/organization/{orgname}

Authorizations: `oauth2_implicit(org:admin)`

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

请求正文模式(application/json)
现有机构的更新描述

名称	描述	模式
email <i>可选</i>	机构联系电子邮件	字符串
invoice_email <i>optional</i>	无论组织是否需要接收发票的电子邮件	布尔值
invoice_email_address <i>optional</i>	接收发票的电子邮件地址	
tag_expiration_s <i>optional</i>	标签过期的秒数	整数

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.14. getApplicationInformation

获取指定应用程序的信息。

GET /api/v1/app/{client_id}

授权：

路径参数

类型	Name	描述	模式
path	client_id <i>required</i>	OAuth 客户端 ID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12. 权限

管理存储库权限。

2.12.1. getUserTransitivePermission

获取指定用户的权限。

GET /api/v1/repository/{repository}/permissions/user/{username}/transitive
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	需要用户名	权限应用到的用户的用户名	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.2. deleteUserPermissions

删除用户的权限。

DELETE /api/v1/repository/{repository}/permissions/user/{username}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	需要用户名	权限应用到的用户的用户名	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.3. getUserPermissions

获取指定用户的权限。

GET /api/v1/repository/{repository}/permissions/user/{username}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	需要用户名	权限应用到的用户的用户名	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.4. changeUserPermissions

更新现有存储库的估算。

PUT /api/v1/repository/{repository}/permissions/user/{username}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	需要用户名	权限应用到的用户的用户名	字符串

请求正文模式(application/json)

用户权限的描述。

名称	描述	模式
角色 <i>可选</i>	用户使用的角色	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.5. deleteTeamPermissions

删除指定团队的权限。

DELETE /api/v1/repository/{repository}/permissions/team/{teamname}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	<i>需要</i> teamname	权限应用到的团队的名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.6. getTeamPermissions

获取指定团队的权限。

GET /api/v1/repository/{repository}/permissions/team/{teamname}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	<i>需要</i> teamname	权限应用到的团队名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.7. changeTeamPermissions

更新现有团队权限。

PUT /api/v1/repository/{repository}/permissions/team/{teamname}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	<i>需要</i> teamname	权限应用到的团队名称	字符串

请求正文模式(application/json)

团队权限的描述。

名称	描述	模式
角色 <i>可选</i>	团队使用的角色	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.8. listRepoTeamPermissions

列出所有团队权限。

GET /api/v1/repository/{repository}/permissions/team/
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.9. listRepoUserPermissions

列出所有用户权限。

GET /api/v1/repository/{repository}/permissions/user/
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13. PROTOTYPE

管理添加到存储库的默认权限。

2.13.1. deleteOrganizationPrototypePermission

删除现有的权限原型。

DELETE /api/v1/organization/{orgname}/prototypes/{prototypeid}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	<i>需要工具</i>	原型的 ID	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.2. updateOrganizationPrototypePermission

更新现有权限原型的角色。

PUT /api/v1/organization/{orgname}/prototypes/{prototypeid}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串
path	需要工具	原型的 ID	字符串

请求正文模式(application/json)

新原型角色的描述

名称	描述	模式
角色 可选	应该应用到该权限的角色	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.3. getOrganizationPrototypePermissions

列出此机构的现有原型。

GET /api/v1/organization/{orgname}/prototypes

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.4. createOrganizationPrototypePermission

创建新权限原型。

POST /api/v1/organization/{orgname}/prototypes

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

请求正文模式(application/json)

新原型描述

名称	描述	模式
角色 <i>可选</i>	应该应用到委托的角色	字符串

名称	描述	模式
activate_user <i>可选</i>	创建用户的存储库，应应用该规则	对象
委托 <i>可选</i>	有关规则授予访问权限的用户或团队的信息	对象

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14. 软件仓库

列出、创建和管理存储库。

2.14.1. listRepos

在各种情况下，获取当前用户可见的存储库列表。

GET /api/v1/repository

Authorizations: oauth2_implicit (存储库 : read)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页页面的页面令牌	字符串
query	repo_kind <i>可选</i>	要返回的仓库种类	字符串
query	popularity <i>可选</i>	是否包含存储库的流行度指标。	布尔值
query	last_modified <i>optional</i>	是否包含存储库最后一次修改的时间。	布尔值

类型	Name	描述	模式
query	公共 可选	添加 virtue 是公开的用户可见的所有软件仓库	布尔值
query	星级 可选	过滤返回到用户星级的存储库	布尔值
query	namespace 可选	过滤返回到此命名空间的存储库	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.2. createRepo

创建新存储库。

POST /api/v1/repository

Authorizations: `oauth2_implicit(repo:create)`

请求正文模式(`application/json`)

新仓库的描述

名称	描述	模式
仓库 可选	仓库名称	字符串
可见性 可选	存储库以什么方式启动	字符串
namespace 可选	创建存储库的命名空间。如果省略，则使用调用者的用户名	字符串

名称	描述	模式
description <i>可选</i>	对存储库进行标记编码的描述	字符串
repo_kind <i>可选</i>	仓库的类型	

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.3. changeRepoVisibility

更改存储库的可见性。

POST /api/v1/repository/{repository}/changevisibility
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)
 更改存储库的可见性。

名称	描述	模式
可见性 <i>可选</i>	存储库以什么方式启动	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.4. changeRepoState

更改存储库的状态。

PUT /api/v1/repository/{repository}/changestate

Authorizations: `oauth2_implicit(repo:admin)`

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(`application/json`)

更改存储库的状态。

名称	描述	模式
State <i>可选</i>	确定是否允许推送。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.5. deleteRepository

删除存储库。

DELETE /api/v1/repository/{repository}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.6. getRepo

获取指定的存储库。

GET /api/v1/repository/{repository}

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	includeTags <i>可选</i>	是否包含存储库标签	布尔值
query	includeStats <i>可选</i>	是否包含操作统计	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.7. updateRepo

更新指定仓库中的描述。

PUT /api/v1/repository/{repository}

Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)

可以在存储库中更新的字段。

名称	描述	模式
description <i>可选</i>	对存储库进行标记编码的描述	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.15. REPOSITORYNOTIFICATION

列出、创建和管理存储库事件/通知。

2.15.1. testRepoNotification

为这个存储库队列测试通知。

POST /api/v1/repository/{repository}/notification/{uuid}/test

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	<i>必需的</i> UUID	通知的 UUID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.2. deleteRepoNotification

删除指定的通知。

DELETE /api/v1/repository/{repository}/notification/{uuid}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	必需的 UUID	通知的 UUID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.3. getRepoNotification

获取指定通知的信息。

GET /api/v1/repository/{repository}/notification/{uuid}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	必需的 UUID	通知的 UUID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.15.4. resetRepositoryNotificationFailures

将存储库通知重置为 0 个失败。

POST /api/v1/repository/{repository}/notification/{uuid}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	必需的 UUID	通知的 UUID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.5. listRepoNotifications

列出指定存储库的通知。

GET /api/v1/repository/{repository}/notification/
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.6. createRepoNotification

POST /api/v1/repository/{repository}/notification/
 Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)
 在存储库中创建通知的信息

名称	描述	模式
事件 <i>可选</i>	通知将响应的事件	字符串
方法 <i>可选</i>	通知方法（如电子邮件或 Web 回调）	字符串
配置 <i>可选</i>	通知方法的 JSON 配置信息	对象
eventConfig <i>可选</i>	通知特定事件的 JSON 配置信息	对象
标题 <i>可选</i>	通知的人类可读标题	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16. REPOTOKEN

管理存储库访问令牌(DEPRECATED)。

2.16.1. deleteToken

删除存储库令牌。

DELETE /api/v1/repository/{repository}/tokens/{code}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	代码	令牌代码	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.2. getTokens

获取指定的存储库令牌信息。

GET /api/v1/repository/{repository}/tokens/{code}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	代码	令牌代码	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.3. changeToken

更新指定存储库令牌的权限。

PUT /api/v1/repository/{repository}/tokens/{code}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	代码	令牌代码	字符串

请求正文模式(application/json)

令牌权限的描述

名称	描述	模式
角色 可选	用于令牌的角色	字符串

名称	描述	模式
----	----	----

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.4. listRepoTokens

列出指定存储库的令牌。

GET /api/v1/repository/{repository}/tokens/
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.5. createToken

创建新的存储库令牌。

**POST /api/v1/repository/{repository}/tokens/
Authorizations:** oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

请求正文模式(application/json)
新令牌的描述。

名称	描述	模式
friendlyName <i>可选</i>	帮助识别令牌的友好名称	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17. ROBOT

管理用户和机构机器人帐户。

2.17.1. getUserRobots

列出用户的可用机器人。

GET /api/v1/user/robots
Authorizations: oauth2_implicit(user:admin)

查询参数

类型	Name	描述	模式
query	限制 <i>可选</i>	如果指定，则要返回的机器数量。	整数
query	令牌 <i>可选</i>	如果为 false，则机器人的令牌不会返回。	布尔值
query	权限 <i>可选</i>	是否要包含机器有权限的存储库和团队。	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.2. getOrgRobotPermissions

返回机构机器人的存储库权限列表。

GET /api/v1/organization/{orgname}/robots/{robot_shortname}/permissions

Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串
path	robot_shortname <i>required</i>	机器器的短名称，没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
---------	----	----

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.3. regenerateOrgRobotToken

为机构机器人重新生成令牌。

POST /api/v1/organization/{orgname}/robots/{robot_shortname}/regenerate

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	robot_shortname <i>required</i>	机器机器人的短名称，没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.4. getUserRobotPermissions

返回用户机器人的存储库权限列表。

GET /api/v1/user/robots/{robot_shortname}/permissions**Authorizations:** oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器机器的短名称, 没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.5. regenerateUserRobotToken

重新生成用户机器人的令牌。

POST /api/v1/user/robots/{robot_shortname}/regenerate**Authorizations:** oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器机器的短名称, 没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.6. deleteOrgRobot

删除现有的机构机器人。

DELETE /api/v1/organization/{orgname}/robots/{robot_shortname}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	robot_shortname <i>required</i>	机器机器人的短名称，没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.7. getOrgRobot

返回具有指定名称的组织机器人。

GET /api/v1/organization/{orgname}/robots/{robot_shortname}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	robot_shortname <i>required</i>	机器机器的短名称，没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.8. createOrgRobot

在 组织中创建新机器人。

PUT /api/v1/organization/{orgname}/robots/{robot_shortname}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	robot_shortname <i>required</i>	机器机器的短名称，没有任何用户或机构前缀	字符串

请求正文模式(application/json)

创建机器机器的可选数据

名称	描述	模式
description <i>可选</i>	机器人的可选文本描述	字符串

名称	描述	模式
<code>unstructured_metadata</code> <i>optional</i>	机器人的可选非结构化元数据	对象

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.9. getOrgRobots

列出机构的机器人。

GET `/api/v1/organization/{orgname}/robots`

Authorizations: `oauth2_implicit(org:admin)`

路径参数

类型	Name	描述	模式
path	需要机构名称	机构名称	字符串

查询参数

类型	Name	描述	模式
query	限制 <i>可选</i>	如果指定, 则要返回的机器人数量。	整数
query	令牌 <i>可选</i>	如果为 <code>false</code> , 则机器人的令牌不会返回。	布尔值
query	权限 <i>可选</i>	是否包括机器人有权限的仓库和团队。	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.10. deleteUserRobot

删除现有的机器人。

DELETE /api/v1/user/robots/{robot_shortname}
Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器机器人的短名称，没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.11. getUserRobot

返回用户机器人具有指定名称。

GET /api/v1/user/robots/{robot_shortname}
Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器机器的短名称, 没有任何用户或机构前缀	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.12. createUserRobot

使用指定名称创建一个新用户机器机器。

PUT /api/v1/user/robots/{robot_shortname}

Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器机器的短名称, 没有任何用户或机构前缀	字符串

请求正文模式(application/json)

创建机器机器的可选数据

名称	描述	模式
description <i>可选</i>	机器人的可选文本描述	字符串
unstructured_metadata <i>optional</i>	机器人的可选非结构化元数据	对象

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18. SEARCH

对所有注册表上下文执行搜索。

2.18.1. conductRepoSearch

获取与指定查询匹配的应用程序和存储库列表。

GET /api/v1/find/repositories

授权：

查询参数

类型	Name	描述	模式
query	includeUsage <i>可选</i>	是否包括使用元数据	布尔值
query	<i>页 可选</i>	该页面。	整数
query	query <i>可选</i>	搜索查询。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.18.2. conductSearch

获取与指定查询匹配的实体和资源列表。

GET /api/v1/find/all

Authorizations: oauth2_implicit (存储库 : read)

查询参数

类型	Name	描述	模式
query	query <i>可选</i>	搜索查询。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.3. getMatchingEntities

获取与指定前缀匹配的实体列表。

GET /api/v1/entities/{prefix}

授权：

路径参数

类型	Name	描述	模式
path	<i>所需的前缀</i>		字符串

查询参数

类型	Name	描述	模式
query	includeOrgs <i>可选</i>	是否包括机构名称。	布尔值
query	includeTeams <i>可选</i>	是否包括团队名称。	布尔值
query	namespace <i>可选</i>	查询组织实体时使用的命名空间。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.19. SECSCAN

列出和管理存储库漏洞和其他安全信息。

2.19.1. getRepoManifestSecurity

GET /api/v1/repository/{repository}/manifest/{manifestref}/security

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	manifestref <i>需要</i>	清单摘要	字符串

查询参数

类型	Name	描述	模式
query	漏洞 可选	包括漏洞信息	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.19.2. getRepolImageSecurity

获取存储库镜像的功能和漏洞（若有）。

GET /api/v1/repository/{repository}/image/{imageid}/security

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	imageID 必需	镜像 ID	字符串
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	漏洞 可选	包括漏洞信息	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20. 超级用户

超级用户 API.

2.20.1. listAllUsers

返回系统中所有用户的列表。

GET /api/v1/superuser/users/
Authorizations: oauth2_implicit(super:user)

查询参数

类型	Name	描述	模式
query	禁用 可选	如果为 false, 则只返回启用的用户。	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.2. createInstallUser

创建新用户。

POST /api/v1/superuser/users/

Authorizations: oauth2_implicit(super:user)

请求正文模式(application/json)

创建用户的数据

名称	描述	模式
username <i>optional</i>	创建用户的用户名	字符串
email <i>可选</i>	创建用户的电子邮件地址	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.3. listAllLogs

列出当前系统的使用日志。

GET /api/v1/superuser/logs

Authorizations: oauth2_implicit(super:user)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页页面的页面令牌	字符串
query	页 <i>可选</i>	日志的页面号	整数
query	ENDTIME <i>可选</i>	获取日志的最新时间(%m/%d/%Y %Z)	字符串

类型	Name	描述	模式
query	startTime 可选	从中获取日志的早期时间(%m/%d/%Y %Z)	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.4. listServiceKeys

GET /api/v1/superuser/keys

Authorizations: oauth2_implicit(super:user)

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.5. createServiceKey

POST /api/v1/superuser/keys

Authorizations: oauth2_implicit(super:user)

请求正文模式(application/json)

创建服务密钥的描述

名称	描述	模式
服务 <i>可选</i>	使用这个密钥进行身份验证的服务	字符串
name <i>可选</i>	服务密钥的友好名称	字符串
元数据 <i>可选</i>	此键元数据的键/值对	对象
notes <i>可选</i>	如果指定, 则密钥的额外备注	字符串
过期 <i>可选</i>	作为 unix 时间戳的过期日期	

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.6. deleteOrganization

删除指定的机构。

DELETE /api/v1/superuser/organizations/{name}
Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	name <i>必需</i>	管理有 organization 的名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.7. changeOrganization

更新指定用户的信息。

PUT /api/v1/superuser/organizations/{name}

Authorizations: `oauth2_implicit(super:user)`

路径参数

类型	Name	描述	模式
path	name <i>必需</i>	管理有 organization 的名称	字符串

请求正文模式(`application/json`)

现有机构的更新描述

名称	描述	模式
email <i>可选</i>	机构联系电子邮件	字符串
invoice_email <i>optional</i>	无论组织是否需要接收发票的电子邮件	布尔值
invoice_email_address <i>optional</i>	接收发票的电子邮件地址	
tag_expiration_s <i>optional</i>	标签过期的秒数	整数

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.8. approveServiceKey

POST /api/v1/superuser/approvedkeys/{kid}

Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	kid <i>必需</i>	服务密钥的唯一标识符	字符串

请求正文模式(application/json)

批准服务密钥的信息

名称	描述	模式
notes <i>可选</i>	可选批准备注	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.9. deleteServiceKey

DELETE /api/v1/superuser/keys/{kid}
Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	kid 必需	服务密钥的唯一标识符	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.10. getServiceKey

GET /api/v1/superuser/keys/{kid}
Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	kid 必需	服务密钥的唯一标识符	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.20.11. updateServiceKey

PUT /api/v1/superuser/keys/{kid}

Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	kid 必需	服务密钥的唯一标识符	字符串

请求正文模式(application/json)

服务密钥更新的描述

名称	描述	模式
name 可选	服务密钥的友好名称	字符串
元数据 可选	此键元数据的键/值对	对象
过期 可选	作为 unix 时间戳的过期日期	

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.12. getRepoBuildStatusSuperUser

返回构建 `uuids` 指定的构建的状态。

GET `/api/v1/superuser/{build_uuid}/status`

Authorizations: `oauth2_implicit(super:user)`

路径参数

类型	Name	描述	模式
path	<code>build_uuid</code> <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.13. getRepoBuildSuperUser

返回有关构建的信息。

GET `/api/v1/superuser/{build_uuid}/build`

Authorizations: `oauth2_implicit(super:user)`

路径参数

类型	Name	描述	模式
path	<code>build_uuid</code> <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError

HTTP 代码	描述	模式
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.20.14. getRepoBuildLogsSuperUser

返回构建 uuid 指定构建的构建日志。

GET /api/v1/superuser/{build_uuid}/logs
Authorizations: oauth2_implicit(super:user)

路径参数

类型	Name	描述	模式
path	build_uuid <i>required</i>	构建的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21. TAG

管理存储库的标签。

2.21.1. restoreTag

将存储库标签还原回存储库中的以前的镜像。

POST /api/v1/repository/{repository}/tag/{tag}/restore
Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	标签 <i>是必需的</i>	标签的名称	字符串

请求正文模式(application/json)

将标签恢复到特定镜像

名称	描述	模式
镜像 <i>可选</i>	(已弃用：使用 inventory_digest) 镜像，这些镜像应指向该标签	字符串
manifest_digest <i>optional</i>	如果指定，应使用的清单摘要	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.2. listTagImages

列出指定存储库标签的镜像。

GET /api/v1/repository/{repository}/tag/{tag}/images

Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

类型	Name	描述	模式
path	标签 是必需的	标签的名称	字符串

查询参数

类型	Name	描述	模式
query	拥有的 可选	如果指定，则只返回由此标签拥有的镜像。	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.3. deleteFullTag

删除指定的存储库标签。

DELETE /api/v1/repository/{repository}/tag/{tag}

Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	标签 是必需的	标签的名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.4. changeTag

更改标签指向或创建新标签的镜像。

PUT /api/v1/repository/{repository}/tag/{tag}

Authorizations: oauth2_implicit (存储库 : write)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	标签 <i>是必需的</i>	标签的名称	字符串

请求正文模式(application/json)

更改特定标签

名称	描述	模式
镜像 <i>可选</i>	(已弃用 : 使用 inventory_digest) 镜像标签应指向的镜像。	
manifest_digest <i>optional</i>	(如果指定) 标签应指向的清单摘要	
过期 <i>可选</i>	(如果指定) 镜像的过期时间	

响应

HTTP 代码	描述	模式
200	成功调用	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.5. listRepoTags

GET /api/v1/repository/{repository}/tag/
 Authorizations: oauth2_implicit (存储库 : read)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	onlyActiveTags 可选	将 过滤为仅活跃的标签。	布尔值
query	页 可选	结果页面索引。默认 1。	整数
query	限制 可选	限制为每个页面返回的结果数。最大 100 个。	整数
query	specificTag 可选	过滤标签到特定标签。	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.22. TEAM

创建、列出和管理组织团队。

2.22.1. getOrganizationTeamPermissions

返回机构团队的存储库权限列表。

GET /api/v1/organization/{orgname}/team/{teamname}/permissions

授权：

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	<i>需要 teamname</i>	团队名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22.2. deleteOrganizationTeamMember

删除团队成员。

If the user is merely invited to join the team, then the invite is removed instead.

DELETE /api/v1/organization/{orgname}/team/{teamname}/members/{membername}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	membername <i>必需</i>	团队成员的用户名	字符串
path	<i>需要</i> teamname	团队名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22.3. updateOrganizationTeamMember

为现有团队添加或邀请一个成员。

PUT /api/v1/organization/{orgname}/team/{teamname}/members/{membername}

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	membername <i>必需</i>	团队成员的用户名	字符串
path	<i>需要</i> teamname	团队名称	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22.4. getOrganizationTeamMembers

检索指定团队的成员列表。

GET /api/v1/organization/{orgname}/team/{teamname}/members

Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	<i>需要 teamname</i>	团队名称	字符串

查询参数

类型	Name	描述	模式
query	包含Pending <i>可选</i>	是否包含待处理成员	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.22.5. deleteOrganizationTeam

删除指定的团队。

DELETE /api/v1/organization/{orgname}/team/{teamname}
Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串
path	<i>需要 teamname</i>	团队名称	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22.6. updateOrganizationTeam

更新指定团队的机构范围内权限。

PUT /api/v1/organization/{orgname}/team/{teamname}
Authorizations: oauth2_implicit(org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要机构名称</i>	机构名称	字符串

类型	Name	描述	模式
path	<i>需要</i> teamname	团队名称	字符串

请求正文模式(application/json)

团队描述

名称	描述	模式
角色 <i>可选</i>	应该应用到团队的机构广泛权限	字符串
description <i>可选</i>	团队的 markdown 描述	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23. TRIGGER

创建、列出和管理构建触发器。

2.23.1. activateBuildTrigger

激活指定的构建触发器。

POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/activate
 Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

类型	Name	描述	模式
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

请求正文模式(application/json)

名称	描述	模式
配置 <i>可选</i>	任意 json.	对象
pull_robot <i>optional</i>	用于拉取镜像的机器机器的名称。	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.2. listTriggerRecentBuilds

列出由指定触发器启动的构建。

GET /api/v1/repository/{repository}/trigger/{trigger_uuid}/builds

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

查询参数

类型	Name	描述	模式
query	限制 <i>可选</i>	要返回的最大构建数	整数

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.3. manuallyStartBuildTrigger

从指定的触发器手动启动构建。

POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/start
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

请求正文模式(application/json)

用于激活构建触发器的可选运行参数

名称	描述	模式
branch_name <i>可选</i>	(仅限 SCM) , 如果指定, 要构建的分支名称。	字符串
commit_sha <i>optional</i>	(仅自定义) 如果指定, 则用于签出 git 存储库的 ref/SHA1。	字符串

名称	描述	模式
Refs <i>可选</i>	(仅限 SCM) 如果指定, 则要构建的 ref。	

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.4. deleteBuildTrigger

删除指定的构建触发器。

DELETE /api/v1/repository/{repository}/trigger/{trigger_uuid}

Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.23.5. getBuildTrigger

获取指定构建触发器的信息。

GET /api/v1/repository/{repository}/trigger/{trigger_uuid}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.6. updateBuildTrigger

更新指定的构建触发器。

PUT /api/v1/repository/{repository}/trigger/{trigger_uuid}
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

类型	Name	描述	模式
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串

请求正文模式(application/json)
更新构建触发器的选项

名称	描述	模式
已启用 <i>可选</i>	是否启用构建触发器	布尔值

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.7. listBuildTriggers

列出指定存储库的触发器。

GET /api/v1/repository/{repository}/trigger/
Authorizations: oauth2_implicit(repo:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
200	成功调用	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24. USER

管理当前用户。

2.24.1. listStarredRepos

列出所有星级存储库。

GET `/api/v1/user/starred`

Authorizations: `oauth2_implicit(user:admin)`

查询参数

类型	Name	描述	模式
query	<code>next_page</code> <i>optional</i>	下一页页面的页面令牌	字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.2. createStar

星级存储库。

POST `/api/v1/user/starred`

Authorizations: oauth2_implicit (存储库 : read)

请求正文模式(application/json)

名称	描述	模式
namespace 可选	仓库所属的命名空间	字符串
仓库 可选	仓库名称	字符串

响应

HTTP 代码	描述	模式
201	成功创建	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.3. getLoggedInUser

获取经过身份验证的用户的用户信息。

GET /api/v1/user/

Authorizations: oauth2_implicit (用户 : read)

响应

HTTP 代码	描述	模式
200	成功调用	UserView
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.4. deleteStar

从存储库中删除星级。

DELETE /api/v1/user/starred/{repository}

Authorizations: oauth2_implicit(user:admin)

路径参数

类型	Name	描述	模式
path	仓库	仓库的完整路径。例如 namespace/name	字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	必需的会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.5. getUserInformation

获取指定用户的用户信息。

GET /api/v1/users/{username}

授权：

路径参数

类型	Name	描述	模式
path	需要用户名		字符串

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	必需的会话	ApiError

HTTP 代码	描述	模式
403	未授权访问	ApiError
404	未找到	ApiError

2.25. 定义

2.25.1. ApiError

名称	描述	模式
status <i>可选</i>	响应的状态代码。	整数
type <i>可选</i>	引用错误的类型。	字符串
详情 <i>可选</i>	有关错误特定实例的详情。	字符串
标题 <i>可选</i>	用于标识错误类型的唯一错误代码。	字符串
error_message <i>optional</i>	弃用; 详情的别名	字符串
error_type <i>可选</i>	弃用; 详情的别名	字符串

2.25.2. UserView

名称	描述	模式
已验证 <i>可选</i>	是否验证用户的电子邮件地址	布尔值
匿名 <i>可选</i>	如果这个用户数据代表客户机用户，则为 true	布尔值
email <i>可选</i>	用户的电子邮件地址	字符串
avatar <i>可选</i>	代表用户图标的 Avatar 数据	对象

名称	描述	模式
机构 <i>可选</i>	有关用户所属机构的信息	对象数组
登录 <i>可选</i>	用户通过身份验证的外部登录供应商列表	对象数组
can_create_repo <i>optional</i>	用户是否具有创建存储库的权限	布尔值
preferred_namespace <i>optional</i>	如果为 true，则用户的命名空间是要显示的首选命名空间	布尔值

2.25.3. ViewMirrorConfig

名称	描述	模式
is_enabled <i>可选</i>	用于启用或禁用同步。	布尔值
external_reference <i>可选</i>	外部存储库的位置。	字符串
external_registry_username <i>可选</i>	用于与外部 registry 进行身份验证的用户名。	
external_registry_password <i>可选</i>	用于与外部 registry 进行身份验证的密码。	
sync_start_date <i>optional</i>	下次确定此存储库准备好同步的时间。	字符串
sync_interval <i>可选</i>	next_start_date 开始同步后的秒数。	整数
robot_username <i>optional</i>	用于镜像推送的机器人的用户名。	字符串
root_rule <i>optional</i>	用于决定应同步哪些标签的 glob-patterns 列表。	对象

名称	描述	模式
<code>external_registry_config</code> <i>optional</i>		对象

2.25.4. ApiErrorDescription

名称	描述	模式
<code>type</code> <i>可选</i>	引用错误类型资源	字符串
<code>标题</code> <i>可选</i>	错误标题。可用于唯一标识错误类型。	字符串
<code>description</code> <i>可选</i>	有关错误的更多详细描述，其中可能包括修复此问题的帮助。	字符串