



Red Hat Quay 3.11

Red Hat Quay API 指南

Red Hat Quay API 指南

法律通告

Copyright © 2024 Red Hat, Inc.

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	3
第 1 章 使用 RED HAT QUAY API	4
1.1. 从 QUAY.IO 访问 QUAY API	4
1.2. 创建 OAUTH 访问令牌	4
1.3. 从 WEB 浏览器访问 QUAY API	5
1.4. 从命令行访问 RED HAT QUAY API	6
第 2 章 RED HAT QUAY 应用程序编程接口(API)	13
2.1. 授权	13
2.2. APPSPECIFICTOKENS	13
2.3. BUILD	16
2.4. DISCOVERY	20
2.5. 错误	21
2.6. GLOBALMESSAGES	22
2.7. LOGS	23
2.8. 清单(INVENTORY)	32
2.9. MIRROR	35
2.10. NAMESPACEQUOTA	40
2.11. 机构	49
2.12. 权限	63
2.13. POLICY	69
2.14. PROTOTYPE	79
2.15. 软件仓库	82
2.16. REPOSITORYNOTIFICATION	87
2.17. REPOTOKEN	92
2.18. 机器人	95
2.19. SEARCH	104
2.20. SECSCAN	106
2.21. 超级用户	107
2.22. TAG	124
2.23. 团队	127
2.24. TRIGGER	133
2.25. USER	139
2.26. 定义	142
第 3 章 API 配置示例	145
3.1. EXTERNAL_REGISTRY_CONFIG OBJECT REFERENCE	145
3.2. RULE_RULE 对象引用	145

前言

Red Hat Quay 应用程序编程接口(API)是一个 OAuth 2 RESTful API，它由一组端点组成，用于为 Red Hat Quay 添加、显示、更改和删除功能。

Red Hat Quay 遵循 [Semantic Versioning \(SemVer\)规格](#)。每个主版本、次版本和补丁版本都会满足以下条件：

- Red Hat Quay 的主版本可能包括不兼容的 API 更改。例如，Red Hat Quay 2.0 的 API 与 Red Hat Quay 3.0 不同。
- Red Hat Quay 的次版本（例如 3.y）以向后兼容的方式添加功能。
- Red Hat Quay 的补丁版本（如 3.y.z）引入了向后兼容的程序错误修复。

目前，Red Hat Quay 将 **api/v1** 端点用于 3.y.z 版本。

本指南描述了 **api/v1** 端点和用于访问这些端点的基于浏览器的示例。

第 1 章 使用 RED HAT QUAY API

Red Hat Quay 提供了完整的 OAuth 2、RESTful API :

- 可从 URL <https://<yourquayhost>/api/v1> 中每个 Red Hat Quay 实例的端点提供
- 允许您通过浏览器连接到端点，通过启用 Swagger UI 获取、删除、发布和放置 Red Hat Quay 设置
- 可以被发出 API 调用和使用 OAuth 令牌的应用程序访问
- 以 JSON 的形式发送和接收数据

以下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。下一节会列出并描述了 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，您可以从网页浏览器浏览 Quay.io 提供的 Red Hat Quay API :

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

出现的 API Explorer 显示 Quay.io API 端点。您将看到 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）。

在 API Explorer 中，您可以获取并有时更改信息：

- 账单、订阅和计划
- 存储库构建和构建触发器
- 错误消息和全局消息
- 存储库镜像、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
- 更多信息

选择打开一个端点，以查看端点的每个部分的 Model Schema。打开端点，输入任何所需参数（如存储库名称或镜像），然后选择 **Try it out!** 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

OAuth 访问令牌是允许您以安全的方式访问受保护的资源的凭证。使用 Red Hat Quay，您必须先创建 OAuth 访问令牌，然后才能访问机构的 API 端点。

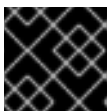
使用以下步骤创建 OAuth 访问令牌。

先决条件

- 您已以管理员身份登录到 Red Hat Quay。

步骤

1. 在主页上，选择一个 Organization。
2. 在导航窗格中，选择 **Applications**。
3. 点 **Create New Application** 并提供新应用程序名称，然后按 **Enter**。
4. 在 **OAuth Applications** 页面中，选择应用程序的名称。
5. 可选。输入以下信息：
 - a. 应用程序名称
 - b. 主页 URL
 - c. 描述
 - d. **avatar E-mail**
 - e. 重定向/Callback URL 前缀
6. 在导航窗格中，选择 **Generate Token**。
7. 选中以下选项框：
 - a. 管理机构
 - b. 管理软件仓库
 - c. 创建软件仓库
 - d. 查看所有可见的存储库
 - e. 对任何可访问的软件仓库的读/写
 - f. 超级用户访问
 - g. 管理用户
 - h. 读取用户信息
8. 点 **Generate Access Token**。您将被重定向到一个新页面。
9. 查看您允许的权限，然后点 **Authorize Application**。点 **Authorize Application** 来确认您的决定。
10. 您将被重定向到 **Access Token** 页面。复制并保存访问令牌。



重要

这是复制和保存访问令牌的唯一机会。在离开此页后无法保留它。

1.3. 从 WEB 浏览器访问 QUAY API

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

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

这种访问 API 的方式不包括在 Red Hat Quay 安装中提供的超级用户端点。以下是通过运行 `swagger-ui` 容器镜像访问本地系统中运行的 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
```

运行 `swagger-ui` 容器后，打开 Web 浏览器来 `localhost` 端口 8888，以通过 `swagger-ui` 容器查看 API 端点。

为了避免日志中出现错误，如 "API 调用 必须通过 X-Requested-With 标头调用（如果从浏览器调用），请在集群中的所有节点上添加以下行到集群中所有节点上的 `config.yaml`，并重启 Red Hat Quay：

```
BROWSER_API_CALLS_XHR_ONLY: false
```

1.4. 从命令行访问 RED HAT QUAY API

您可以使用 `curl` 命令通过 Red Hat Quay 集群的 API 进行 GET、PUT、POST 或 DELETE 设置。将 `<token>` 替换为之前创建的 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 书所述：
 - 使用配置编辑器 UI 或
 - 直接编辑 **config.yaml** 文件，使用 `选项使用配置 API 验证（和下载）更新的配置捆绑包`
- 为超级用户名称创建用户帐户：
 - 获取以上所述的授权令牌，并使用 **curl** 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

```

- 返回的内容包括为新用户帐户生成的密码：

```

{
  "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

可以使用 internal API `/api/v1/superuser/logs` 列出当前系统的使用情况日志。结果被分页，因此在以下示例中创建了超过 20 个仓库，以显示如何使用多个调用来访问整个结果集。

1.4.3.1. 分页示例

第一次调用

```

$ 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 进行第二次调用

```

$ 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

要在组织 **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 9rJYBR3v3pXcj5XqlA2XX6Thkww4gld4TCYLLWDF" \
-H "Content-type: application/json" \
-d '{"group_dn": "cn=ldapgroup,ou=Users"}' \
http://quay1-server:8080/api/v1/organization/testadminorg/team/newteam/syncing

```

禁用同一团队的同步：

```

$ curl -X DELETE -H "Authorization: Bearer 9rJYBR3v3pXcj5XqlA2XX6Thkww4gld4TCYLLWDF" \
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 和其他所需文件的 **tar** 或 **zip** 存档。**file_id** 参数是我们较旧的构建系统的一部分。它无法再使用。如果 Dockerfile 位于子目录中，则需要同时指定它。

归档应该可以被公开访问。OAuth 应用应具有 "Administer Organization" 范围，因为只有机构管理员有权访问机器人的帐户令牌。否则，通过只向机器人授予构建访问权限（无需访问自身），并使用它来获取镜像内容，从而获得机器人权限。如果出现错误，请检查返回的 json 块，并确保正确传递存档位置、拉取机器人和其他参数。单击单个构建页面右上角的“下载日志”，以检查日志以了解更多信息。

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. 创建私有软件仓库

```
$ 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
```

1.4.9. 创建已镜像的存储库

最小配置

```
curl -X POST
-H "Authorization: Bearer ${bearer_token}"
-H "Content-Type: application/json"
--data '{"external_reference": "quay.io/minio/mc", "external_registry_username": "", "sync_interval":
600, "sync_start_date": "2021-08-06T11:11:39Z", "root_rule": {"rule_kind": "tag_glob_csv",
"rule_value": [ "latest" ]}, "robot_username": "orga+robot"}'
https://${quay_registry}/api/v1/repository/${orga}/${repo}/mirror | jq
```

扩展配置

```
$ curl -X POST
-H "Authorization: Bearer ${bearer_token}"
-H "Content-Type: application/json"
--data '{"is_enabled": true, "external_reference": "quay.io/minio/mc", "external_registry_username":
"username", "external_registry_password": "password", "external_registry_config":
{"unsigned_images": true, "verify_tls": false, "proxy": {"http_proxy": "http://proxy.tld", "https_proxy":
"https://proxy.tld", "no_proxy": "domain"}}, "sync_interval": 600, "sync_start_date": "2021-08-
06T11:11:39Z", "root_rule": {"rule_kind": "tag_glob_csv", "rule_value": [ "*" ]}, "robot_username":
"orga+robot"}' https://${quay_registry}/api/v1/repository/${orga}/${repo}/mirror | jq
```


第 2 章 RED HAT QUAY 应用程序编程接口(API)

此 API 允许您执行使用 Red Hat Quay 存储库、用户和机构所需的许多操作。

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	此应用将能够查看和拉取对授予用户或机器人帐户可见的所有存储库
repo:write	此应用将能够查看、推送和拉取给授予用户或机器人帐户具有写入访问权限的所有存储库
repo:admin	此应用将具有对授予用户或机器人帐户具有访问权限的所有存储库的管理员访问权限
repo:create	此应用将能够在 中创建存储库到授予用户或机器人帐户的任何命名空间中，这些命名空间被允许创建存储库
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	此应用将可以管理您的组织，包括创建人人、创建团队、调整团队成员资格和更改账单设置。在授予此权限之前，您应该对请求应用具有绝对信任。
super:user	此应用程序将可以管理您的安装，包括管理用户、管理机构和其他在超级用户面板中的功能。在授予此权限之前，您应该对请求应用具有绝对信任。
user:admin	此应用将可以管理您的帐户，包括创建机器人并将其权限授予您的存储库。在授予此权限之前，您应该对请求应用具有绝对信任。

2.2. APPSPECIFICTOKENS

管理当前用户的应用程序特定令牌。

2.2.1. createAppToken

为用户创建新应用程序特定令牌。

POST /api/v1/user/apptoken

授权：oauth2_implicit (user:admin)

请求正文架构(application/json)

新令牌的描述。

名称	描述	模式
标题 必需	友好名称，以帮助识别令牌	字符串

响应

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

2.2.2. listAppTokens

列出用户的应用程序特定令牌。

GET /api/v1/user/apptoken

授权：oauth2_implicit (user:admin)

查询参数

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

响应

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

HTTP 代码	描述	模式
404	未找到	ApiError

2.2.3. getAppToken

返回用户的特定应用程序令牌。

GET /api/v1/user/apptoken/{token_uuid}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	token_uuid <i>required</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.2.4. revokeAppToken

为用户撤销特定的应用程序令牌。

DELETE /api/v1/user/apptoken/{token_uuid}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	token_uuid <i>required</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3. BUILD

创建、列出、取消和获取存储库构建的状态/日志。

2.3.1. getRepoBuildStatus

返回构建 `uids` 指定的构建状态。

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

授权： `oauth2_implicit (repo:read)`

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 <code>namespace/name</code>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.2. getRepoBuildLogs

返回构建 uuid 指定的构建的构建日志。

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

授权：oauth2_implicit (repo:read)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.3. getRepoBuild

返回有关构建的信息。

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

授权：oauth2_implicit (repo:read)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.4. cancelRepoBuild

取消存储库构建。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.3.5. requestRepoBuild

请求构建存储库并从指定的输入推送。

POST /api/v1/repository/{repository}/build/

授权：oauth2_implicit (repo:write)

路径参数

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

请求正文架构(application/json)

新存储库构建的描述。

名称	描述	模式
file_id <i>optional</i>	构建规格上传时生成的文件 id	字符串
archive_url <i>optional</i>	要构建的 .tar.gz 的 URL。必须以 "http" 或 "https" 开头。	字符串
子目录 <i>可选</i>	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>	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	string
标签 <i>可选</i>	构建的镜像要推送到的标签。如果未指定，则使用 "latest"。	字符串数组 非空字符串

响应

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

2.3.6. getRepoBuilds

获取存储库构建列表。

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

授权：oauth2_implicit (repo:read)

路径参数

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

查询参数

类型	Name	描述	模式
query	从 可选	返回自给定的 unix 时间码后的所有构建	整数
query	限制 可选	要返回的构建的最大数量	整数

响应

HTTP 代码	描述	模式
200	成功调用	
400	错误请求	ApiError
401	需要会话	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 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	需要会话	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	标识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
401	需要会话	ApiError

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

2.6. GLOBALMESSAGES

消息 API。

2.6.1. createGlobalMessage

创建消息。

POST /api/v1/messages

授权：oauth2_implicit (super:user)

请求正文架构(application/json)

创建新消息

名称	描述	模式
message <i>必需</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.6.2. getGlobalMessages

返回超级用户消息。

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	需要会话	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}
 授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	UUID 必需		字符串

响应

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

2.7. LOGS

访问机构或存储库的使用日志。

2.7.1. getAggregateUserLogs

返回当前用户的聚合日志。

GET /api/v1/user/aggregatelogs

授权：oauth2_implicit (user:admin)

查询参数

类型	Name	描述	模式
query	执行者 <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："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

响应

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

2.7.2. exportUserLogs

返回当前用户的聚合日志。

POST /api/v1/user/exportlogs

授权：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："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

请求正文架构(application/json)

配置导出日志操作

名称	描述	模式
<code>callback_url</code> <i>optional</i>	使用导出日志的链接调用的回调 URL	字符串
<code>callback_email</code> <i>optional</i>	发送到导出日志的链接的电子邮件地址	字符串

响应

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

2.7.3. listUserLogs

列出当前用户的日志。

GET /api/v1/user/logs

授权：oauth2_implicit (user:admin)

查询参数

类型	Name	描述	模式
query	<code>next_page</code> <i>optional</i>	下一页的页面令牌	字符串
query	执行者 <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："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

响应

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

2.7.4. getAggregateOrgLogs

获取指定机构的聚合日志。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName 必需	机构名称	字符串

查询参数

类型	Name	描述	模式
query	执行者 可选	过滤日志的用户名。	字符串
query	ENDTIME 可选	日志的更新时间。格式："%m/%d/%Y" (UTC)。	字符串
query	startTime 可选	日志的最早时间。格式："%m/%d/%Y" (UTC)。	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.7.5. exportOrgLogs

导出指定机构的日志。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	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：“%m/%d/%Y” (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式：“%m/%d/%Y” (UTC)。	字符串

请求正文架构(application/json)

配置导出日志操作

名称	描述	模式
callback_url <i>optional</i>	使用导出日志的链接调用的回调 URL	字符串
callback_email <i>optional</i>	发送到导出日志的链接的电子邮件地址	字符串

响应

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

2.7.6. listOrgLogs

列出指定机构的日志。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页的页面令牌	字符串
query	执行者 <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："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.7.7. getAggregateRepoLogs

返回指定存储库的聚合日志。

GET /api/v1/repository/{repository}/aggregatelogs

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

响应

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

HTTP 代码	描述	模式
404	未找到	ApiError

2.7.8. exportRepoLogs

为指定存储库对日志进行导出排队。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

查询参数

类型	Name	描述	模式
query	ENDTIME 可选	日志的最新时间。格式："%m/%d/%Y" (UTC)。	字符串
query	startTime 可选	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

请求正文架构(application/json)

配置导出日志操作

名称	描述	模式
callback_url <i>optional</i>	使用导出日志的链接调用的回调 URL	字符串
callback_email <i>optional</i>	发送到导出日志的链接的电子邮件地址	字符串

响应

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

HTTP 代码	描述	模式
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.7.9. listRepoLogs

列出指定存储库的日志。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页的页面令牌	字符串
query	ENDTIME <i>可选</i>	日志的最新时间。格式："%m/%d/%Y" (UTC)。	字符串
query	startTime <i>可选</i>	日志的最早时间。格式："%m/%d/%Y" (UTC)。	字符串

响应

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

HTTP 代码	描述	模式
404	未找到	ApiError

2.8. 清单(INVENTORY)

管理存储库的清单。

2.8.1. getManifestLabel

使用清单下的特定 ID 检索标签。

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

授权：oauth2_implicit (repo:read)

路径参数

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

响应

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

2.8.2. deleteManifestLabel

从清单中删除现有标签。

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

授权：oauth2_implicit (repo:write)

路径参数

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

响应

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

2.8.3. addManifestLabel

将新标签添加到标签清单中。

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

授权：oauth2_implicit (repo:write)

路径参数

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

请求正文架构(application/json)

为清单添加标签

名称	描述	模式
key <i>必需</i>	标签的密钥	string
<i>必需的值</i>	标签的值	string
media_type <i>required</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.4. listManifestLabels

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

授权：oauth2_implicit (repo:read)

路径参数

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

查询参数

类型	Name	描述	模式
query	<i>过滤 可选</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.8.5. getRepoManifest

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

授权：oauth2_implicit (repo:read)

路径参数

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

响应

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

2.9. MIRROR

2.9.1. syncCancel

更新给定存储库的镜像配置的 sync_status。

POST /api/v1/repository/{repository}/mirror/sync-cancel

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.2. syncNow

更新给定存储库的镜像配置的 sync_status。

POST /api/v1/repository/{repository}/mirror/sync-now

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.9.3. getRepoMirrorConfig

返回一个给定存储库的 Mirror 配置。

GET /api/v1/repository/{repository}/mirror

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.4. changeRepoMirrorConfig

允许用户修改存储库的镜像配置。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

更新存储库镜像配置。

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

响应

HTTP 代码	描述	模式
201	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.9.5. createRepoMirrorConfig

为给定的存储库创建一个 RepoMirrorConfig。

POST /api/v1/repository/{repository}/mirror

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

创建存储库镜像配置。

名称	描述	模式
is_enabled <i>optional</i>	用于启用或禁用同步。	布尔值
external_reference <i>required</i>	外部存储库的位置。	字符串
external_registry_username <i>optional</i>	用于与外部 registry 进行身份验证的用户名。	
external_registry_password <i>optional</i>	用于与外部 registry 进行身份验证的密码。	
sync_start_date <i>required</i>	决定此存储库下次准备好同步的时间。	字符串
sync_interval <i>required</i>	next_start_date 开始同步后的秒数。	整数
robot_username <i>required</i>	用于镜像推送的机器人的用户名。	字符串
root_rule <i>required</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10. NAMESPACEQUOTA

2.10.1. listUserQuota

GET /api/v1/user/quota

授权：oauth2_implicit (user:admin)

响应

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

2.10.2. getOrganizationQuotaLimit

GET /api/v1/organization/{orgname}/quota/{quota_id}/limit/{limit_id}

授权：

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串

类型	Name	描述	模式
path	limit_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

响应

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

2.10.3. changeOrganizationQuotaLimit

PUT /api/v1/organization/{orgname}/quota/{quota_id}/limit/{limit_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	limit_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

请求正文架构(application/json)

更改机构配额限制的描述

名称	描述	模式
type <i>可选</i>	配额限制类型："Warning" 或 "Reject"	字符串

名称	描述	模式
<code>threshold_percentage</code> <i>optional</i>	配额阈值，以配额百分比	整数

响应

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

2.10.4. deleteOrganizationQuotaLimit

DELETE /api/v1/organization/{orgname}/quota/{quota_id}/limit/{limit_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	<code>quota_id</code> <i>required</i>		字符串
path	<code>limit_id</code> <i>required</i>		字符串
path	<code>OrgName</code> <i>必需</i>		字符串

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	需要会话	ApiError

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

2.10.5. createOrganizationQuotaLimit

POST /api/v1/organization/{orgname}/quota/{quota_id}/limit

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

请求正文架构(application/json)

新机构配额限制的描述

名称	描述	模式
type <i>必需</i>	配额限制类型："Warning" 或 "Reject"	string
threshold_perce nt <i>required</i>	配额阈值，以配额百分比	整数

响应

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

2.10.6. listOrganizationQuotaLimit

GET /api/v1/organization/{orgname}/quota/{quota_id}/limit

授权：

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

响应

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

2.10.7. getUserQuotaLimit

GET /api/v1/user/quota/{quota_id}/limit/{limit_id}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	limit_id <i>required</i>		字符串

响应

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

2.10.8. listUserQuotaLimit

GET /api/v1/user/quota/{quota_id}/limit

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串

响应

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

2.10.9. getOrganizationQuota

GET /api/v1/organization/{orgname}/quota/{quota_id}

授权：

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

响应

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

2.10.10. changeOrganizationQuota

PUT /api/v1/organization/{orgname}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

请求正文架构(application/json)

新机构配额的描述

名称	描述	模式
limit_bytes <i>optional</i>	允许机构的字节数	整数

响应

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

2.10.11. deleteOrganizationQuota

DELETE /api/v1/organization/{orgname}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串
path	OrgName <i>必需</i>		字符串

响应

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

2.10.12. createOrganizationQuota

创建新的机构配额。

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

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	OrgName 必需		字符串

请求正文架构(application/json)

新机构配额的描述

名称	描述	模式
limit_bytes <i>required</i>	允许机构的字节数	整数

响应

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

2.10.13. listOrganizationQuota

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

授权：

路径参数

类型	Name	描述	模式
path	OrgName 必需		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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.10.14. getUserQuota

GET /api/v1/user/quota/{quota_id}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串

响应

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

2.11. 机构

管理机构、成员和 OAuth 应用。

2.11.1. createOrganization

创建新组织。

POST /api/v1/organization/

授权：oauth2_implicit (user:admin)

请求正文架构(application/json)
新组织的描述。

名称	描述	模式
name <i>必需</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>optional</i>	(may 被禁用) 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
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.2. validateProxyCacheConfig

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

授权：

路径参数

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

请求正文架构(application/json)
机构的代理缓存配置

名称	描述	模式
<code>upstream_registry</code> <i>required</i>	要缓存的上游 registry 的名称	字符串

响应

HTTP 代码	描述	模式
202	成功创建	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.3. getOrganizationCollaborators

列出指定机构的外部协作者。

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

授权： `oauth2_implicit (org:admin)`

路径参数

类型	Name	描述	模式
path	<code>OrgName</code> <i>必需</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	需要会话	ApiError

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

2.11.4. getOrganizationApplication

使用指定组织下的指定 `client_id` 检索应用。

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

授权： `oauth2_implicit (org:admin)`

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
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
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}`

授权： `oauth2_implicit (org:admin)`

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
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(application/json)
更新的应用程序的描述。

名称	描述	模式
name <i>必需</i>	应用程序的名称	字符串
redirect_uri <i>required</i>	应用的 OAuth 重定向的 URI	字符串
application_uri <i>required</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>optional</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	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.6. deleteOrganizationApplication

删除此机构下的应用程序。

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

授权：oauth2_implicit (org:admin)

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
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.7. createOrganizationApplication

在此机构中创建新应用。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(application/json)

新机构应用程序的描述。

名称	描述	模式
name <i>必需</i>	应用程序的名称	字符串

名称	描述	模式
<code>redirect_uri</code> <i>optional</i>	应用的 OAuth 重定向的 URI	字符串
<code>application_uri</code> <i>optional</i>	应用程序主页的 URI	字符串
<code>description</code> <i>可选</i>	应用程序的人类可读描述	字符串
<code>avatar_email</code> <i>optional</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.8. getOrganizationApplications

列出指定机构的应用程序。

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

授权：`oauth2_implicit (org:admin)`

路径参数

类型	Name	描述	模式
path	<code>OrgName</code> <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.9. getProxyCacheConfig

检索组织的代理缓存配置。

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

授权：

路径参数

类型	Name	描述	模式
path	OrgName 必需	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.10. deleteProxyCacheConfig

删除机构的代理缓存配置。

DELETE /api/v1/organization/{orgname}/proxycache

授权：

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.11. createProxyCacheConfig

为机构创建代理缓存配置。

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

授权：

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(application/json)

机构的代理缓存配置

名称	描述	模式
upstream_registry <i>required</i>	要缓存的上游 registry 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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.12. getOrganizationMember

检索组织成员的详细信息。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	memberName <i>必需</i>	机构成员的用户名	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.13. removeOrganizationMember

从机构中删除成员，撤销所有存储库特权，并将它从组织中的所有团队中删除。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	memberName 必需	机构成员的用户名	string
path	OrgName 必需	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.14. getOrganizationMembers

列出指定机构的人员。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName 必需	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

HTTP 代码	描述	模式
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.15. 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	OrgName 必需	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.16. changeOrganizationDetails

更改指定机构的详情。

PUT /api/v1/organization/{orgname}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(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>	组织是否希望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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.17. deleteAdminedOrganization

删除指定的机构。

DELETE /api/v1/organization/{orgname}
授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.11.18. 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	需要会话	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
 授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要用户名</i>	权限应用到的用户的用户名	字符串
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.2. getUserPermissions

获取指定用户的权限。

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

路径参数

类型	Name	描述	模式
path	<i>需要用户名</i>	应用权限的用户的用户名	字符串
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.3. changeUserPermissions

更新现有存储库的目标。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要用户名	应用权限的用户的用户名	字符串
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串

请求正文架构(application/json)

用户权限的描述。

名称	描述	模式
需要角色	用于用户的角色	字符串

响应

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

HTTP 代码	描述	模式
404	未找到	ApiError

2.12.4. deleteUserPermissions

删除用户的权限。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要用户名</i>	应用权限的用户的用户名	字符串
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.12.5. getTeamPermissions

获取指定团队的权限。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 namespace/name	字符串

类型	Name	描述	模式
path	团队名称 必需	权限应用到的团队名称	string

响应

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

2.12.6. changeTeamPermissions

更新现有团队权限。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	团队名称 必需	权限应用到的团队名称	string

请求正文架构(application/json)

团队权限的描述。

名称	描述	模式
需要角色	用于团队的角色	字符串

响应

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

2.12.7. deleteTeamPermissions

删除指定团队的权限。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	团队名称 必需	权限应用到的团队名称	string

响应

HTTP 代码	描述	模式
204	已删除	
400	错误请求	ApiError
401	需要会话	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/

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	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/

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13. POLICY

2.13.1. createOrganizationAutoPrunePolicy

为机构创建一个自动修剪策略

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName 必需	机构名称	字符串

请求正文架构(application/json)

要应用到用户命名空间的策略配置

Name	描述	模式
方法所需	用于修剪标签的方法(number_of_tags、creation_date)	string
必需的值	用于修剪方法的值（标签数，如 10, time delta，如 7d (7 天)）	

响应

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

2.13.2. listOrganizationAutoPrunePolicies

列出机构的自动修剪策略

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.3. getOrganizationAutoPrunePolicy

获取机构的自动修剪策略

GET /api/v1/organization/{orgname}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.13.4. deleteOrganizationAutoPrunePolicy

删除机构的自动修剪策略

DELETE /api/v1/organization/{orgname}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.5. updateOrganizationAutoPrunePolicy

更新机构的自动修剪策略

PUT /api/v1/organization/{orgname}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string

请求正文架构(application/json)
要应用到用户命名空间的策略配置

Name	描述	模式
方法所需	用于修剪标签的方法(number_of_tags、creation_date)	string
必需的值	用于修剪方法的值 (标签数, 如 10, time delta, 如 7d (7 天))	

响应

HTTP 代码	描述	模式
204	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.6. createRepositoryAutoPrunePolicy

为存储库创建一个自动修剪策略

POST /api/v1/repository/{repository}/autoprunepolicy/
授权: oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)
要应用到用户命名空间的策略配置

Name	描述	模式
方法 可选	用于修剪标签的方法(number_of_tags、 creation_date)	string
value 可选	用于修剪方法的值 (标签数, 如 10, time delta, 如 7d (7 天))	

响应

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

2.13.7. listRepositoryAutoPrunePolicies

列出存储库的自动修剪策略

GET /api/v1/repository/{repository}/autoprunepolicy/

授权 : oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.13.8. getRepositoryAutoPrunePolicy

获取存储库的自动修剪策略

GET /api/v1/repository/{repository}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.9. deleteRepositoryAutoPrunePolicy

删除存储库的自动修剪策略

DELETE /api/v1/repository/{repository}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.13.10. updateRepositoryAutoPrunePolicy

更新存储库的自动修剪策略

PUT /api/v1/repository/{repository}/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串

请求正文架构(application/json)

要应用到用户命名空间的策略配置

Name	描述	模式
方法 <i>可选</i>	用于修剪标签的方法(number_of_tags、creation_date)	string
value <i>可选</i>	用于修剪方法的值 (标签数, 如 10, time delta, 如 7d (7 天))	

响应

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

2.13.11. createUserAutoPrunePolicy

为当前登录的用户创建自动修剪策略

POST /api/v1/user/autoprunepolicy/

授权：oauth2_implicit (user:admin)

请求正文架构(application/json)

要应用到用户命名空间的策略配置

Name	描述	模式
方法所需	用于修剪标签的方法(number_of_tags、 creation_date)	string
必需的值	用于修剪方法的值（标签数，如 10, time delta，如 7d (7 天)）	

响应

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

2.13.12. listUserAutoPrunePolicies

列出当前登录用户的自动修剪策略

GET /api/v1/user/autoprunepolicy/

授权：oauth2_implicit (user:admin)

响应

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

2.13.13. getUserAutoPrunePolicy

获取当前登录用户的 auto-prune 策略

GET /api/v1/user/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string

响应

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

2.13.14. deleteUserAutoPrunePolicy

删除当前登录用户的自动修剪策略

DELETE /api/v1/user/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string

响应

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

2.13.15. updateUserAutoPrunePolicy

更新当前登录用户的自动修剪策略

PUT /api/v1/user/autoprunepolicy/{policy_uuid}

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	policy_uuid <i>required</i>	策略的唯一 ID	string

请求正文架构(application/json)

要应用到用户命名空间的策略配置

Name	描述	模式
方法所需	用于修剪标签的方法(number_of_tags、 creation_date)	string

Name	描述	模式
必需的值	用于修剪方法的值 (标签数, 如 10, time delta, 如 7d (7 天))	

响应

HTTP 代码	描述	模式
204	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14. PROTOTYPE

管理添加到存储库的默认权限。

2.14.1. updateOrganizationPrototypePermission

更新现有权限原型的角色。

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

授权: oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要原型</i>	原型的 ID	string
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(application/json)

新原型角色的描述

名称	描述	模式
角色 <i>可选</i>	应该应用到权限的角色	字符串

响应

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

2.14.2. deleteOrganizationPrototypePermission

删除现有的权限原型。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	<i>需要原型</i>	原型的 ID	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.3. createOrganizationPrototypePermission

创建新的权限原型。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName <i>必需</i>	机构名称	字符串

请求正文架构(application/json)

新原型的描述

名称	描述	模式
<i>需要角色</i>	应该应用到委派的角色	字符串
activating_user <i>optional</i>	创建规则应应用到的用户的存储库	object
<i>需要委托</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.14.4. getOrganizationPrototypePermissions

列出此组织的现有原型。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
----	------	----	----

类型	Name	描述	模式
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15. 软件仓库

列出、创建和管理存储库。

2.15.1. createRepo

创建新存储库。

POST /api/v1/repository

授权：oauth2_implicit (repo:create)

请求正文架构(application/json)

新存储库的描述

名称	描述	模式
<i>需要</i> 仓库	仓库名称	string
<i>所需</i> 可见性	可见性存储库将以什么开头	字符串
namespace <i>可选</i>	应该在其中创建存储库的命名空间。如果省略，则使用调用者的用户名	string
描述 <i>必需</i>	用于仓库的 Markdown 编码描述	字符串
repo_kind <i>optional</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.2. listRepos

在各种情况下，获取对当前用户可见的存储库列表。

GET /api/v1/repository

授权：oauth2_implicit (repo:read)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页的页面令牌	字符串
query	repo_kind <i>optional</i>	要返回的仓库类型	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>可选</i>	是否包含存储库最后一次修改的时间。	布尔值
query	public <i>可选</i>	添加任何用户可见的仓库，因为其为公共	布尔值
query	starred <i>可选</i>	过滤用户返回的存储库	布尔值
query	namespace <i>可选</i>	过滤返回到此命名空间的仓库	字符串

响应

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

2.15.3. changeRepoVisibility

更改存储库的可见性。

POST /api/v1/repository/{repository}/changevisibility

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

更改存储库的可见性。

名称	描述	模式
所需可见性	可见性存储库将以什么开头	字符串

响应

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

2.15.4. changeRepoState

更改存储库的状态。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

更改存储库的状态。

名称	描述	模式
State 必需	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.5. getRepo

获取指定的软件仓库。

GET /api/v1/repository/{repository}

授权：oauth2_implicit (repo:read)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.15.6. updateRepo

更新指定存储库中的描述。

PUT /api/v1/repository/{repository}

授权：oauth2_implicit (**repo:write**)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 namespace/name	字符串

请求正文架构(**application/json**)

可以在存储库中更新的字段。

名称	描述	模式
描述 <i>必需</i>	用于仓库的 Markdown 编码描述	字符串

响应

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

2.15.7. deleteRepository

删除存储库。

DELETE /api/v1/repository/{repository}

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16. REPOSITORYNOTIFICATION

列出、创建和管理存储库事件/通知。

2.16.1. testRepoNotification

将此存储库的测试通知排队。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	UUID 必需	通知的 UUID	string

响应

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

2.16.2. getRepoNotification

获取指定通知的信息。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	UUID 必需	通知的 UUID	string

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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.3. deleteRepoNotification

删除指定的通知。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	UUID 必需	通知的 UUID	string

响应

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

2.16.4. resetRepositoryNotificationFailures

将存储库通知重置为 0 个失败。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

响应

HTTP 代码	描述	模式
204	成功创建	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.5. createRepoNotification

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 namespace/name	字符串

请求正文架构(application/json)

在仓库上创建通知的信息

名称	描述	模式
事件 <i>必需</i>	通知将响应的事件	string
方法 <i>所需</i>	通知方法（如电子邮件或 Web 回调）	string
配置 <i>必需</i>	用于特定通知方法的 JSON 配置信息	object

名称	描述	模式
eventConfig <i>必需</i>	用于通知的特定事件的 JSON 配置信息	对象
title <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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.16.6. listRepoNotifications

列出指定存储库的通知。

GET /api/v1/repository/{repository}/notification/

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.17. REPOTOKEN

管理存储库访问令牌(DEPRECATED)。

2.17.1. getTokens

获取指定的存储库令牌信息。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 namespace/name	字符串
path	<i>需要代码</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.2. changeToken

更新指定存储库令牌的权限。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(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

响应

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

2.17.3. deleteToken

删除存储库令牌。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	需要代码	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.17.4. createToken

创建新存储库令牌。

POST /api/v1/repository/{repository}/tokens/

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

新令牌的描述。

名称	描述	模式
friendlyName 必需	友好名称，以帮助识别令牌	字符串

响应

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

2.17.5. listRepoTokens

列出指定存储库的令牌。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18. 机器人

管理用户和组织机器人帐户。

2.18.1. getUserRobots

列出用户可用的机器人。

GET /api/v1/user/robots

授权：oauth2_implicit (user:admin)

查询参数

类型	Name	描述	模式
query	限制 可选	如果指定，要返回的机器人数量。	整数
query	令牌 可选	如果为 false，则不会返回机器人的令牌。	布尔值
query	权限 可选	是否包括机器人具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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.2. getOrgRobotPermissions

返回机构机器人的存储库权限列表。

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

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortcode <i>required</i>	机器人的短名称，没有任何用户或机构前缀	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.3. regenerateOrgRobotToken

为机构机器人重新生成令牌。

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

授权：oauth2_implicit (org:admin)

路径参数

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

响应

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

2.18.4. getUserRobotPermissions

返回用户机器人的存储库权限列表。

GET /api/v1/user/robots/{robot_shortname}/permissions

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	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
200	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.5. regenerateUserRobotToken

为用户的机器人重新生成令牌。

POST /api/v1/user/robots/{robot_shortname}/regenerate

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.6. getOrgRobot

使用指定名称返回机构的机器人。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器人的短名称, 没有任何用户或机构前缀	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.7. createOrgRobot

在组织中创建一个新的机器人。

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

授权: oauth2_implicit (org:admin)

路径参数

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

请求正文架构(application/json)

创建机器人的可选数据

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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.8. deleteOrgRobot

删除现有的机构机器人。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	机器人的短名称，没有任何用户或机构前缀	string
path	OrgName <i>必需</i>	机构名称	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.9. getOrgRobots

列出机构的机器人。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	OrgName 必需	机构名称	字符串

查询参数

类型	Name	描述	模式
query	限制 可选	如果指定，要返回的机器人数量。	整数
query	令牌 可选	如果为 false，则不会返回机器人的令牌。	布尔值
query	权限 可选	是否包括机器人具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	需要会话	ApiError

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

2.18.10. getUserRobot

使用指定名称返回用户的机器人。

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

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	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
200	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.11. createUserRobot

使用指定名称创建新用户 robot。

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

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
----	------	----	----

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

请求正文架构(application/json)
创建机器人的可选数据

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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.18.12. deleteUserRobot

删除现有的机器人。

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

路径参数

类型	Name	描述	模式
path	robot_shortname <i>required</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.19. SEARCH

对所有 registry 上下文执行搜索。

2.19.1. 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>	是否包含使用元数据	布尔值
query	页 <i>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	需要会话	ApiError

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

2.19.2. conductSearch

获取与指定查询匹配的实体和资源列表。

GET /api/v1/find/all

授权：oauth2_implicit (repo: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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.19.3. 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	前缀 <i>必需</i>		字符串

查询参数

类型	Name	描述	模式
query	includeOrgs <i>可选</i>	是否包含机构名称。	布尔值
query	includeTeams <i>可选</i>	是否包含团队名称。	布尔值
query	namespace <i>可选</i>	查询机构实体时要使用的命名空间。	字符串

响应

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

2.20. SECSCAN

列出和管理存储库漏洞和其他安全信息。

2.20.1. getRepoManifestSecurity

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

授权：oauth2_implicit (repo:read)

路径参数

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

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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21. 超级用户

超级用户 API。

2.21.1. createInstallUser

创建新用户。

POST /api/v1/superuser/users/
授权：oauth2_implicit (super:user)

请求正文架构(application/json)
创建用户的数据

名称	描述	模式
需要用户名	正在创建的用户的用户名	字符串
email 可选	正在创建的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	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.2. listAllUsers

返回系统中所有用户的列表。

GET /api/v1/superuser/users/

授权：oauth2_implicit (super:user)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页的页面令牌	字符串
query	限制 <i>可选</i>	限制为每个页面要返回的结果数。最大 100。	整数
query	禁用 <i>可选</i>	如果为 false，则仅返回启用的用户。	布尔值

响应

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

2.21.3. listAllLogs

列出当前系统的使用情况日志。

GET /api/v1/superuser/logs

授权：oauth2_implicit (super:user)

查询参数

类型	Name	描述	模式
query	next_page <i>optional</i>	下一页的页面令牌	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
query	startTime <i>可选</i>	从中获取日志的最早时间(%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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.4. createServiceKey

POST /api/v1/superuser/keys

授权：oauth2_implicit (super:user)

请求正文架构(application/json)

创建服务密钥的描述

名称	描述	模式
服务必需	使用此密钥进行身份验证的服务	字符串

名称	描述	模式
name <i>可选</i>	服务键的友好名称	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，密钥的额外备注	string
过期 <i>必需</i>	过期日期作为 unix 时间戳	

响应

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

2.21.5. listServiceKeys

GET /api/v1/superuser/keys

授权：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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.6. changeUserQuotaSuperUser

PUT /api/v1/superuser/organization/{namespace}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace <i>必需</i>		字符串
path	quota_id <i>required</i>		字符串

请求正文架构(application/json)

新机构配额的描述

名称	描述	模式
limit_bytes <i>optional</i>	允许机构的字节数	整数

响应

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

2.21.7. deleteUserQuotaSuperUser

DELETE /api/v1/superuser/organization/{namespace}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

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

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串

响应

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

2.21.8. createUserQuotaSuperUser

POST /api/v1/superuser/organization/{namespace}/quota

授权：oauth2_implicit (super:user)

路径参数

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

请求正文架构(application/json)

新机构配额的描述

名称	描述	模式
limit_bytes <i>required</i>	允许机构的字节数	整数

响应

HTTP 代码	描述	模式
201	成功创建	

HTTP 代码	描述	模式
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.9. listUserQuotaSuperUser

GET /api/v1/superuser/organization/{namespace}/quota

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace 必需		字符串

响应

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

2.21.10. changeOrganizationQuotaSuperUser

PUT /api/v1/superuser/users/{namespace}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace 必需		字符串

类型	Name	描述	模式
path	quota_id <i>required</i>		字符串

请求正文架构(application/json)
新机构配额的描述

名称	描述	模式
limit_bytes <i>optional</i>	允许机构的字节数	整数

响应

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

2.21.11. deleteOrganizationQuotaSuperUser

DELETE /api/v1/superuser/users/{namespace}/quota/{quota_id}

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace <i>必需</i>		字符串
path	quota_id <i>required</i>		字符串

响应

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

2.21.12. createOrganizationQuotaSuperUser

POST /api/v1/superuser/users/{namespace}/quota
 授权：oauth2_implicit (super:user)

路径参数

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

请求正文架构(application/json)
 新机构配额的描述

名称	描述	模式
limit_bytes <i>optional</i>	允许机构的字节数	整数

响应

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

2.21.13. listOrganizationQuotaSuperUser

GET /api/v1/superuser/users/{namespace}/quota

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace 必需		字符串

响应

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

2.21.14. changeOrganization

更新有关指定用户的信息。

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

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	name 必需	受管机构的名称	字符串

请求正文架构(application/json)

现有机构的更新描述

名称	描述	模式
email 可选	机构联系电子邮件	字符串

名称	描述	模式
<code>invoice_email</code> <i>optional</i>	组织是否希望接收发票的电子邮件	布尔值
<code>invoice_email_address</code> <i>optional</i>	接收发票的电子邮件地址	
<code>tag_expiration_s</code> <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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.15. deleteOrganization

删除指定的机构。

DELETE `/api/v1/superuser/organizations/{name}`

授权：`oauth2_implicit (super:user)`

路径参数

类型	Name	描述	模式
path	<code>name</code> <i>必需</i>	受管机构的名称	字符串

响应

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

HTTP 代码	描述	模式
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.16. approveServiceKey

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

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	需要kid	服务键的唯一标识符	字符串

请求正文架构(application/json)

批准服务密钥的信息

名称	描述	模式
notes 可选	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.17. deleteServiceKey

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

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	需要kid	服务键的唯一标识符	字符串

响应

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

2.21.18. updateServiceKey

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

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	需要kid	服务键的唯一标识符	字符串

请求正文架构(application/json)

服务键的更新描述

名称	描述	模式
name <i>可选</i>	服务键的友好名称	字符串
元数据 <i>可选</i>	此键元数据的键/值对	对象
过期 <i>可选</i>	过期日期作为 unix 时间戳	

响应

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

2.21.19. getServiceKey

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

路径参数

类型	Name	描述	模式
path	<i>需要kid</i>	服务键的唯一标识符	字符串

响应

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

2.21.20. getRepoBuildStatusSuperUser

返回构建 uuids 指定的构建状态。

GET /api/v1/superuser/{build_uuid}/status
 授权：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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.21. getRepoBuildSuperUser

返回有关构建的信息。

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

授权：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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.22. getRepoBuildLogsSuperUser

返回构建 uuid 指定的构建的构建日志。

GET /api/v1/superuser/{build_uuid}/logs

授权：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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.23. getRegistrySize

GET /api/v1/superuser/registrysize/

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace <i>必需</i>		string

镜像 registry 大小的描述

Name	描述	模式
size_bytes* <i>optional</i>	允许机构的字节数	整数
last_ran		整数

Name	描述	模式
Queued		布尔值
running		布尔值

响应

HTTP 代码	描述	模式
200	已创建	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.21.24. postRegistrySize

POST /api/v1/superuser/registrysize/

授权：oauth2_implicit (super:user)

路径参数

类型	Name	描述	模式
path	namespace 必需		字符串

请求正文架构(application/json)

镜像 registry 大小的描述

Name	描述	模式
last_ran		整数
Queued		布尔值
running		布尔值

响应

HTTP 代码	描述	模式
201	已创建	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22. TAG

管理存储库的标签。

2.22.1. restoreTag

将存储库标签恢复回存储库中的旧镜像。

POST /api/v1/repository/{repository}/tag/{tag}/restore

授权：oauth2_implicit (repo:write)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	需要标签	标签的名称	string

请求正文架构(application/json)

将标签恢复到特定镜像

名称	描述	模式
manifest_digest <i>required</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

HTTP 代码	描述	模式
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.22.2. changeTag

更改标签指向或创建新标签的镜像。

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

授权：oauth2_implicit (repo:write)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	需要标签	标签的名称	string

请求正文架构(application/json)

对特定标签进行更改

名称	描述	模式
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
201	成功调用	
400	错误请求	ApiError
401	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.22.3. deleteFullTag

删除指定的存储库标签。

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

授权：oauth2_implicit (repo:write)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。例如 namespace/name	字符串
path	需要标签	标签的名称	string

响应

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

2.22.4. listRepoTags

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

授权：oauth2_implicit (repo:read)

路径参数

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

查询参数

类型	Name	描述	模式
query	onlyActiveTags <i>optional</i>	仅过滤为活动的标签。	布尔值
query	页 <i>可选</i>	结果的页面索引。默认 1。	整数
query	限制 <i>可选</i>	限制为每个页面要返回的结果数。最大 100。	整数
query	filter_tag_name <i>optional</i>	语法 : <op><name> 根据操作过滤标签名称。<op> 可以是 'like' 或 'eq'。	string
query	specificTag <i>可选</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23. 团队

创建、列出和管理机构的团队。

2.23.1. 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	团队名称 <i>必需</i>	团队名称	字符串
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.2. updateOrganizationTeamMember

向现有团队添加或邀请成员。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	团队名称 <i>必需</i>	团队名称	字符串
path	memberName <i>必需</i>	团队成员的用户名	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.3. deleteOrganizationTeamMember

删除团队的成员。

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}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	团队名称 <i>必需</i>	团队名称	字符串
path	memberName <i>必需</i>	团队成员的用户名	string
path	OrgName <i>必需</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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.23.4. getOrganizationTeamMembers

检索指定团队的成员列表。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	团队名称 必需	团队名称	字符串
path	OrgName 必需	机构名称	字符串

查询参数

类型	Name	描述	模式
query	includePending 可选	是否包含待处理的成员	布尔值

响应

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

2.23.5. inviteTeamMemberEmail

邀请到现有团队的电子邮件地址。

PUT /api/v1/organization/{orgname}/team/{teamname}/invite/{email}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	需要电子邮件		string
path	团队名称 必需		字符串
path	OrgName 必需		字符串

响应

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

2.23.6. deleteTeamMemberEmailInvite

删除一个邀请加入团队的电子邮件地址。

DELETE /api/v1/organization/{orgname}/team/{teamname}/invite/{email}

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	需要电子邮件		string
path	团队名称 必需		字符串
path	OrgName 必需		字符串

响应

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

2.23.7. updateOrganizationTeam

更新指定团队的机构范围权限。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	团队名称 必需	团队的名称	字符串
path	OrgName 必需	机构名称	字符串

请求正文架构(application/json)

团队描述

名称	描述	模式
需要角色	机构应该应用到团队的权限	字符串
description 可选	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

HTTP 代码	描述	模式
401	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.23.8. deleteOrganizationTeam

删除指定的团队。

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

授权：oauth2_implicit (org:admin)

路径参数

类型	Name	描述	模式
path	团队名称 必需	团队名称	字符串
path	OrgName 必需	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24. TRIGGER

创建、列出和管理构建触发器。

2.24.1. activateBuildTrigger

激活指定的构建触发器。

POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/activate

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

名称	描述	模式
配置 <i>必需</i>	任意 json。	对象
pull_robot <i>optional</i>	用于拉取镜像的机器人名称。	字符串

响应

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

2.24.2. listTriggerRecentBuilds

列出由指定触发器启动的构建。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

类型	Name	描述	模式
path	<i>需要仓库</i>	存储库的完整路径。例如 namespace/name	字符串

查询参数

类型	Name	描述	模式
query	限制 <i>可选</i>	要返回的构建的最大数量	整数

响应

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

2.24.3. manuallyStartBuildTrigger

从指定的触发器手动启动构建。

POST /api/v1/repository/{repository}/trigger/{trigger_uuid}/start

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(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, 则构建分支的名称。	字符串
commit_sha <i>optional</i>	(仅限自定义) 如果指定, 则使用 ref/SHA1 来签出 git 存储库。	字符串
Refs <i>可选</i>	(仅限 SCM) 如果指定, 则构建 ref。	

响应

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

2.24.4. getBuildTrigger

获取指定构建触发器的信息。

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

授权: `oauth2_implicit (repo:admin)`

路径参数

类型	Name	描述	模式
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串
path	<i>需要仓库</i>	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.5. updateBuildTrigger

更新指定的构建触发器。

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

授权：oauth2_implicit (repo:admin)

路径参数

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

请求正文架构(application/json)

更新构建触发器的选项

名称	描述	模式
需要启用	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	需要会话	ApiError
403	未授权访问	ApiError

HTTP 代码	描述	模式
404	未找到	ApiError

2.24.6. deleteBuildTrigger

删除指定的构建触发器。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	trigger_uuid <i>required</i>	构建触发器的 UUID	字符串
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.24.7. listBuildTriggers

列出指定存储库的触发器。

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

授权：oauth2_implicit (repo:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.25. USER

管理当前用户。

2.25.1. createStar

星级存储库。

POST /api/v1/user/starred

授权：oauth2_implicit (repo:read)

请求正文架构(application/json)

名称	描述	模式
namespace <i>必需</i>	仓库所属的命名空间	string
<i>需要仓库</i>	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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.25.2. listStarredRepos

列出所有星级存储库。

GET /api/v1/user/starred

授权：oauth2_implicit (user:admin)

查询参数

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

响应

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

2.25.3. getLoggedInUser

获取经过身份验证的用户的用户信息。

GET /api/v1/user/

授权：oauth2_implicit (user:read)

响应

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

2.25.4. deleteStar

从存储库中删除星号。

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

授权：oauth2_implicit (user:admin)

路径参数

类型	Name	描述	模式
path	需要仓库	存储库的完整路径。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	需要会话	ApiError
403	未授权访问	ApiError
404	未找到	ApiError

2.25.5. 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	需要会话	ApiError

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

2.26. 定义

2.26.1. 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>	有关错误的具体实例的详细信息。	字符串
title <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; 详细的别名	字符串
error_type <i>optional</i>	弃用; 详细的别名	字符串

2.26.2. UserView

名称	描述	模式
验证的 <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一个客户机用户, 则为 true	布尔值
email <i>可选</i>	用户电子邮件地址	字符串

名称	描述	模式
avatar <i>可选</i>	代表用户图标的 avatar 数据	对象
机构 <i>可选</i>	有关用户所属的机构的信息	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

2.26.3. ViewMirrorConfig

名称	描述	模式
is_enabled <i>optional</i>	用于启用或禁用同步。	布尔值
external_reference <i>optional</i>	外部存储库的位置。	字符串
external_registry_username <i>optional</i>	用于与外部 registry 进行身份验证的用户名。	
external_registry_password <i>optional</i>	用于与外部 registry 进行身份验证的密码。	
sync_start_date <i>optional</i>	决定此存储库下次准备好同步的时间。	字符串
sync_interval <i>optional</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

名称	描述	模式
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

2.26.4. ApiErrorDescription

名称	描述	模式
type <i>可选</i>	对错误类型资源的引用	字符串
title <i>可选</i>	错误的标题。可用于唯一标识错误类型。	字符串
description <i>可选</i>	有关错误的详细描述，可能包含有助于修复问题的帮助。	字符串

第 3 章 API 配置示例

3.1. EXTERNAL_REGISTRY_CONFIG OBJECT REFERENCE

```
{
  "is_enabled": True,
  "external_reference": "quay.io/redhat/quay",
  "sync_interval": 5000,
  "sync_start_date": datetime(2020, 0o1, 0o2, 6, 30, 0),
  "external_registry_username": "fakeUsername",
  "external_registry_password": "fakePassword",
  "external_registry_config": {
    "verify_tls": True,
    "unsigned_images": False,
    "proxy": {
      "http_proxy": "http://insecure.proxy.corp",
      "https_proxy": "https://secure.proxy.corp",
      "no_proxy": "mylocalhost",
    },
  },
}
```

3.2. RULE_RULE 对象引用

```
{
  "root_rule": {"rule_kind": "tag_glob_csv", "rule_value": ["latest", "foo", "bar"]},
}
```