



Red Hat Quay 3

Red Hat Quay API ガイド

Red Hat Quay API ガイド

法律上の通知

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.

概要

Red Hat Quay API の使用

目次

はじめに	3
第1章 RED HAT QUAY API の使用	4
1.1. QUAY.IO からの QUAY API へのアクセス	4
1.2. OAUTH アクセストークンの作成	4
1.3. WEB ブラウザーからの QUAY API へのアクセス	6
1.4. コマンドラインでの RED HAT QUAY API へのアクセス	6
第2章 RED HAT QUAY アプリケーションプログラミングインターフェイス (API)	14
2.1. 承認	14
2.2. APPSPECIFICTOKENS	14
2.3. BUILD	17
2.4. DISCOVERY	22
2.5. ERROR	22
2.6. GLOBALMESSAGES	23
2.7. LOGS	25
2.8. MANIFEST	33
2.9. MIRROR	37
2.10. NAMESPACEQUOTA	41
2.11. 組織	51
2.12. パーミッション	64
2.13. POLICY	70
2.14. PROTOTYPE	77
2.15. REPOSITORY	80
2.16. REPOSITORYNOTIFICATION	85
2.17. REPOTOKEN	90
2.18. ROBOT	93
2.19. SEARCH	102
2.20. SECSCAN	104
2.21. スーパーユーザー	105
2.22. TAG	122
2.23. TEAM	125
2.24. TRIGGER	131
2.25. USER	137
2.26. 定義	140
第3章 API 設定の例	143
3.1. EXTERNAL_REGISTRY_CONFIG オブジェクトリファレンス	143
3.2. RULE_RULE オブジェクトリファレンス	143

はじめに

Red Hat Quay アプリケーションプログラミングインターフェイス (API) は、Red Hat Quay 機能を追加、表示、変更、および削除するための一連のエンドポイントで設定される OAuth 2 RESTful API です。

Red Hat Quay は [Semantic Versioning \(SemVer\) 仕様](#) に準拠しています。次の条件は、メジャー、マイナー、およびパッチリリースごとに満たされます。

- Red Hat Quay のメジャーバージョンには、互換性のない API の変更が含まれている場合があります。たとえば、Red Hat Quay 2.0 の API は Red Hat Quay 3.0 とは異なります。
- Red Hat Quay のマイナーバージョン (3.y など) は、下位互換性のある方法で機能を追加します。
- Red Hat Quay のパッチバージョン (3.y.z など) には、下位互換性のあるバグ修正が導入されています。

現在、Red Hat Quay は 3.y.z リリースに **api/v1** エンドポイントを使用しています。

このガイドでは、**api/v1** エンドポイントと、それらのエンドポイントにアクセスするためのブラウザーベースの例について説明します。

第1章 RED HAT QUAY API の使用

Red Hat Quay は、完全な OAuth 2 RESTful API を提供します。

- 各 Red Hat Quay インスタンスのエンドポイント (URL <https://<yourquayhost>/api/v1>) から利用できます。
- Swagger UI を有効にして、ブラウザ経由でエンドポイントに接続し、Red Hat Quay の設定を取得、削除、投稿、および配置できます。
- API 呼び出しを実行し、OAuth トークンを使用するアプリケーションからアクセスできます。
- JSON としてデータを送受信します。

以下のテキストは、Red Hat Quay API にアクセスし、API を使用して Red Hat Quay クラスターで設定を表示して変更する方法を説明します。次のセクションでは、API エンドポイントをリスト表示し、説明します。

1.1. QUAY.IO からの QUAY API へのアクセス

独自の Red Hat Quay クラスターがまだ実行されていない場合に、Web ブラウザーから Quay.io で利用可能な Red Hat Quay API を確認できます。

<https://docs.quay.io/api/swagger/>

表示される API Explorer には Quay.io API エンドポイントが表示されます。Quay.io で有効でない Red Hat Quay 機能のスーパーユーザー API エンドポイントまたはエンドポイント (リポジトリミラーリングなど) は表示されません。

API Explorer から、以下に関する情報を取得し、変更できます。

- 請求、サブスクリプション、およびプラン
- リポジトリビルドおよびビルドトリガー
- エラーメッセージおよびグローバルメッセージ
- リポジトリイメージ、マニフェスト、パーミッション、通知、脆弱性、およびイメージの署名
- 使用状況に関するログ
- 組織、メンバー、および OAuth アプリケーション
- ユーザーとロボットアカウント
- その他

エンドポイントを選択して開き、エンドポイントの各部分のモデルスキーマを表示します。エンドポイントを開き、必要なパラメーター (リポジトリ名またはイメージなど) を入力し、**Try it out!** ボタンを選択して Quay.io エンドポイントに関連する設定を照会するか、変更します。

1.2. OAUTH アクセストークンの作成

OAuth アクセストークンは、保護されたリソースへのセキュアなアクセスを可能にする認証情報です。Red Hat Quay では、組織の API エンドポイントにアクセスする前に、OAuth アクセストークンを作成する必要があります。

OAuth アクセストークンを作成するには、次の手順を実行します。

前提条件

- Red Hat Quay に管理者としてログインしている。

手順

1. メインページで、Organization を選択します。
2. ナビゲーションペインで、**Applications** を選択します。
3. **Create New Application** をクリックし、新しいアプリケーション名を入力して、**Enter** を押します。
4. **OAuth Applications** ページで、アプリケーションの名前を選択します。
5. オプション: 以下の情報を入力します。
 - a. **Application Name**
 - b. **Homepage URL**
 - c. **Description**
 - d. **Avatar E-mail**
 - e. **Redirect/Callback URL prefix**
6. ナビゲーションペインで、**Generate Token** を選択します。
7. 次のオプションのチェックボックスをオンにします。
 - a. **Administer Organization**
 - b. **Administer Repositories**
 - c. **Create Repositories**
 - d. **View all visible repositories**
 - e. **Read/Write to any accessible repositories**
 - f. **Super User Access**
 - g. **Administer User**
 - h. **Read User Information**
8. **Generate Access Token** をクリックします。新しいページにリダイレクトされます。
9. 許可する権限を確認し、**Authorize Application** をクリックします。**Authorize Application** をクリックして決定した内容を確認します。

10. **Access Token** ページにリダイレクトされます。アクセストークンをコピーして保存します。



重要

これは、アクセストークンをコピーして保存する唯一の機会です。このページを離れると再取得できません。

1.3. WEB ブラウザーからの QUAY API へのアクセス

Swagger を有効にし、Web ブラウザーを使用して独自の Red Hat Quay インスタンスの API にアクセスできます。この URL は、Red Hat Quay API を UI および以下の URL 経由で公開します。

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

この方法で API にアクセスしても、Red Hat Quay インストールで利用可能なスーパーユーザーエンドポイントにはアクセスできません。以下は、swagger-ui コンテナイメージを実行してローカルシステムで実行されている Red Hat Quay API インターフェイスにアクセスする例です。

```
# 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
```

Swagger-ui コンテナが実行された状態で、Web ブラウザーを localhost ポート 8888 で開き、swagger-ui コンテナ経由で API エンドポイントを表示します。

API calls must be invoked with an X-Requested-With header if called from a browser などのエラーを回避するには、以下の行をクラスター内の全ノードの **config.yaml** に追加し、Red Hat Quay を再起動します。

```
BROWSER_API_CALLS_XHR_ONLY: false
```

1.4. コマンドラインでの RED HAT QUAY API へのアクセス

curl コマンドを使用して、Red Hat Quay クラスターの API を使用して GET、PUT、POST、または DELETE 操作を実行できます。**<token>** は、以下の例の設定を取得または変更するために作成した OAuth アクセストークンに置き換えます。

1.4.1. スーパーユーザー情報の取得

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

以下に例を示します。

```
$ 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. API を使用したスーパーユーザーの作成

- Quay のデプロイで説明されているようにスーパーユーザー名を設定します。
 - 設定エディター UI を使用します。または、
 - 設定 API を使用して更新された設定バンドルを検証 (およびダウンロード) して、**config.yaml** ファイルを直接編集します。
- スーパーユーザー名のユーザーアカウントを作成します。
 - 上記のように承認トークンを取得し、**curl** を使用してユーザーを作成します。

```

$ 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

```

- 返されるコンテンツには、新規ユーザーアカウント用に生成されたパスワードが含まれません。

```

{
  "username": "quaysuper",
  "email": "quaysuper@example.com",
  "password": "EH67NB3Y6PTBED8H0HC6UVHGGGA3ODSE",
  "encrypted_password":
  "fn37AZAUQH0PTsU+vlO9IS0QxPW9A/boXL4ovZjIFtlUPrBz9i4j9UDOqMjuxQ/0HTfy38go
KEpG8zYXVeQh3IOFzuOjSvKic2Vq7xdtQsU="
}

```

ユーザーのリストを要求すると、**quaysuper** がスーパーユーザーとして表示されます。

```

$ 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. 使用状況ログのリスト表示

内部 API `/api/v1/superuser/logs` を使用して、現在のシステムの使用状況ログをリスト表示できます。結果はページネーションされます。以下の例では、20 以上のリポジトリを作成し、複数の呼び出しを使用して結果セット全体にアクセスする方法を紹介しています。

1.4.3.1. ページネーションの例

最初の呼び出し

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

初期出力

```

{
  "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"
}

```

next_page を使用した 2 回目の呼び出し

```

$ 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

```

2 回目の呼び出しからの出力

```

{
  "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. ディレクトリーの同期

LDAP の対応するグループ名が **ldapgroup** となる **testadminorg** 組織の **newteam** チームのディレクトリーの同期を有効にするには、以下を実行します。

```

$ curl -X POST -H "Authorization: Bearer 9rJYBR3v3pXcj5XqIA2XX6Thkww4gld4TCYLLWDF" \
  -H "Content-type: application/json" \
  -d '{"group_dn": "cn=ldapgroup,ou=Users"}' \
  http://quay1-server:8080/api/v1/organization/testadminorg/team/newteam/syncing

```

同じチームの同期を無効にするには、以下を実行します。

```

$ curl -X DELETE -H "Authorization: Bearer 9rJYBR3v3pXcj5XqIA2XX6Thkww4gld4TCYLLWDF" \
  http://quay1-server:8080/api/v1/organization/testadminorg/team/newteam/syncing

```

1.4.5. API を使用したリポジトリビルドの作成

指定の入力からリポジトリをビルドし、ビルドにカスタムタグを付けるには、`requestRepoBuild` エンドポイントを使用できます。以下のデータを使用できます。

```

{
  "docker_tags": [

```

```

"string"
  ],
  "pull_robot": "string",
  "subdirectory": "string",
  "archive_url": "string"
}

```

archive_url パラメーターは、Dockerfile とビルドに必要な他のファイルが含まれる **tar** または **zip** アーカイブを参照する必要があります。**file_id** パラメーターは、以前のビルドシステムに含まれていましたが、これは今後使用できません。Dockerfile がサブディレクトリーにある場合は、これも指定する必要があります。

アーカイブは一般に公開されている必要があります。組織の管理者のみがロボットのアカウントトークンにアクセスできるため、OAuth アプリには Administer Organization スコープが必要です。そうしないと、誰かがロボットに対するビルドアクセスを割り当てるだけで、ロボットパーミッションを取得し、そのパーミッションを使用してイメージコンテンツを取得できるようになります。エラーが発生した場合は、返される json ブロックを確認し、アーカイブの場所、プルロボット、およびその他のパラメーターが正しく指定されていることを確認します。個別のビルドページの右上にある Download logs をクリックし、詳細なメッセージがないかログを確認します。

1.4.6. 組織のロボットの作成

```

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

```

1.4.7. ビルドのトリガー

```

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

```

要求のある 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. プライベートリポジトリーの作成

```

$ 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. ミラーリングされたリポジトリーの作成

最小設定

```

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
```

拡張設定

```
$ 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 アプリケーションプログラミングインターフェイス (API)

この API を使用すると、Red Hat Quay リポジトリ、ユーザー、および組織の使用に必要な多くの操作を実行できます。

2.1. 承認

oauth2_implicit

スコープ

以下のスコープは、API エンドポイントへのアクセスを制御するのに使用されます。

スコープ	Description
repo:read	このアプリケーションは、付与ユーザーまたはロボットアカウントに表示されるすべてのリポジトリを表示し、プルできます。
repo:write	このアプリケーションは、付与ユーザーまたはロボットアカウントに書き込みのアクセス権が割り当てられているリポジトリをすべて表示、プッシュ、およびプルできます。
repo:admin	このアプリケーションには、付与ユーザーまたはロボットアカウントがアクセスできるすべてのリポジトリに管理者権限が割り当てられます。
repo:create	このアプリケーションは、付与ユーザーまたはロボットアカウントがリポジトリを作成できる namespace にリポジトリを作成できます。
user:read	このアプリケーションは、ユーザー名やメールアドレスなどのユーザー情報を読み取ることができます。
org:admin	このアプリケーションは、ロボットの作成、チームの作成、チームのメンバーシップの調整、請求設定の変更など、組織の管理を行うことができます。このパーミッションを付与する前に、要求元アプリケーションに絶対の信頼が必要です。
super:user	このアプリケーションは、ユーザーの管理、スーパーユーザーパネルにある組織およびその他の機能の管理など、インストールを管理できます。このパーミッションを付与する前に、要求元アプリケーションに絶対の信頼が必要です。
user:admin	このアプリケーションは、ロボットの作成、リポジトリへのパーミッションを付与するなど、アカウントを管理できます。このパーミッションを付与する前に、要求元アプリケーションに絶対の信頼が必要です。

2.2. APPSPECIFICTOKENS

現在のユーザーのアプリケーション固有のトークンを管理します。

2.2.1. createAppToken

ユーザー向けに新規アプリケーション固有のトークンを作成します。

POST /api/v1/user/apptoken

承認: `oauth2_implicit(user:admin)`

要求の body スキーマ (`application/json`)

新しいトークンの説明。

名前	説明	スキーマ
<code>friendlyName</code> 任意	トークンの特定に役立つ分かりやすい名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.2.2. listAppTokens

ユーザーのアプリケーション固有のトークンをリスト表示します。

GET /api/v1/user/apptoken

承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	<code>expiring</code> 任意	true の場合には、まもなく有効期限が切れるトークンのみを返します。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.2.3. getAppToken

ユーザーの特定のアプリケーショントークンを返します。

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	token_uuid 必須	アプリケーション固有のトークンの uuid	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.2.4. revokeAppToken

ユーザーの特定のアプリケーショントークンを取り消します。

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	token_uuid 必須	アプリケーション固有のトークンの uuid	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3. BUILD

リポジトリビルドの作成、リスト表示、キャンセル、およびステータス/ログの取得

2.3.1. getRepoBuildStatus

ビルドの uuid で指定したビルドのステータスを返します。

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3.2. getRepoBuildLogs

ビルド uuid で指定したビルドのビルドログを返します。

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3.3. getRepoBuild

ビルドに関する情報を返します。

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3.4. cancelRepoBuild

リポジトリビルドをキャンセルします。

DELETE `/api/v1/repository/{repository}/build/{build_uuid}`承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3.5. requestRepoBuild

指定の入力からリポジトリをビルドし、プッシュすることを要求します。

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

承認: oauth2_implicit (repo:write)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

新規リポジトリビルドの説明。

名前	説明	スキーマ
file_id 任意	ビルド仕様のアップロード時に生成されたファイル ID	string
archive_url 任意	ビルドする .tar.gz の URL。http または https で開始する必要があります。	string
subdirectory 任意	Dockerfile があるサブディレクトリー。このサブディレクトリーまたは dockerfile_path のみを指定できます。	string
dockerfile_path 任意	docker ファイルへのパス。このパスまたはサブディレクトリーのみを指定できます。	string
context 任意	dockerfile のコンテキストを指定します。これは任意です。	string
pull_robot 任意	プル認証情報として使用する Quay ロボットアカウントのユーザー名	string

名前	説明	スキーマ
docker_tags 任意	ビルドイメージがプッシュされるタグ。指定のない場合は、latest が使用されます。	string の配列 non-empty unique

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.3.6. getRepoBuilds

リポジトリビルドのリストを取得します。

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

承認: oauth2_implicit(repo:read)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	since 任意	指定の unix タイムコード以降の全ビルドを返します。	integer
query	limit 任意	返すビルドの最大数	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.4. DISCOVERY

API 検出情報。

2.4.1. discovery

Swagger API 形式で利用可能なすべての API エンドポイントをリスト表示します。

GET /api/v1/discovery

承認:

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	internal 任意	内部 API を含めるかどうか。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.5. ERROR

エラー詳細 API。

2.5.1. getErrorDescription

エラーの詳細情報を取得します。

GET /api/v1/error/{error_type}

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	error_type 必須	エラーのタイプを特定するエラーコード。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	ApiErrorDescription
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.6. GLOBALMESSAGES

メッセージ API。

2.6.1. createGlobalMessage

メッセージを作成します。

POST /api/v1/messages

承認: `oauth2_implicit(super:user)`

要求の body スキーマ (application/json)

新規メッセージの作成

名前	説明	スキーマ
message 任意	単一メッセージ	object

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.6.2. getGlobalMessages

スーパーユーザーのメッセージを返します。

GET /api/v1/messages

承認:

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.6.3. deleteGlobalMessage

メッセージを削除します。

DELETE /api/v1/message/{uuid}

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	uuid 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7. LOGS

組織またはリポジトリの使用状況ログにアクセスします。

2.7.1. getAggregateUserLogs

現在のユーザーの集計ログを返します。

GET /api/v1/user/aggregatelogs

承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	performer 任意	ログをフィルターするユーザー名。	string
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.2. exportUserLogs

現在のユーザーの集計ログを返します。

POST /api/v1/user/exportlogs

承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

要求の body スキーマ (application/json)

エクスポートログ操作の設定

名前	説明	スキーマ
callback_url 任意	エクスポートされたログに対するリンクで呼び出すコールバック URL	string
callback_email 任意	エクスポートされたログに対するリンクをメールで送信するメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError

HTTP コード	Description	スキーマ
404	結果が見つからない	ApiError

2.7.3. listUserLogs

現在のユーザーのログをリスト表示します。

GET /api/v1/user/logs

承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	<code>next_page</code> 任意	次のページのページトークン	string
query	<code>performer</code> 任意	ログをフィルターするユーザー名。	string
query	<code>endtime</code> 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	<code>starttime</code> 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.4. getAggregateOrgLogs

指定した組織の集計ログを取得します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	performer 任意	ログをフィルターするユーザー名。	string
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.5. exportOrgLogs

指定した組織のログをエクスポートします。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

要求の body スキーマ (application/json)

エクスポートログ操作の設定

名前	説明	スキーマ
callback_url 任意	エクスポートされたログに対するリンクで呼び出すコールバック URL	string
callback_email 任意	エクスポートされたログに対するリンクをメールで送信するメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.6. listOrgLogs

指定した組織のログをリスト表示します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string
query	performer 任意	ログをフィルターするユーザー名。	string
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.7. getAggregateRepoLogs

指定したリポジトリの集計ログを返します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
-----	----	----	------

タイプ	名前	説明	スキーマ
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.8. exportRepoLogs

指定したリポジトリのログのエクスポートをキューに入れます。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

要求の body スキーマ (application/json)

エクスポートログ操作の設定

名前	説明	スキーマ
callback_url 任意	エクスポートされたログに対するリンクで呼び出すコールバック URL	string
callback_email 任意	エクスポートされたログに対するリンクをメールで送信するメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.7.9. listRepoLogs

指定したリポジトリのログをリスト表示します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string
query	endtime 任意	ログの最新時間。形式: "%m/%d/%Y"(UTC)	string
query	starttime 任意	ログの最初の時間。形式: "%m/%d/%Y"(UTC)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.8. MANIFEST

リポジトリのマニフェストを管理します。

2.8.1. getManifestLabel

マニフェストで特定の ID のラベルを取得します。

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

承認: oauth2_implicit(repo:read)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	manifestref 必須	マニフェストのダイジェスト	string
path	labelid 必須	ラベルの ID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.8.2. deleteManifestLabel

マニフェストから既存のラベルを削除します。

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

承認: `oauth2_implicit (repo:write)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	manifestref 必須	マニフェストのダイジェスト	string
path	labelid 必須	ラベルの ID	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.8.3. addManifestLabel

タグマニフェストに新規ラベルを追加します。

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

承認: `oauth2_implicit (repo:write)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	manifestref 必須	マニフェストのダイジェスト	string

要求の body スキーマ (application/json)

マニフェストにラベルを追加します。

名前	説明	スキーマ
key 任意	ラベルのキー	string
value 任意	ラベルの値	string
media_type 任意	このラベルのメディアタイプ	

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.8.4. listManifestLabels

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

タイプ	名前	説明	スキーマ
path	manifestref 必須	マニフェストのダイジェスト	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	filter 必須	これが指定されている場合には、指定の接頭辞に一致するラベルのみが返されます。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.8.5. getRepoManifest

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	manifestref 必須	マニフェストのダイジェスト	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.9. MIRROR

2.9.1. syncCancel

指定したリポジトリのミラーリング設定の `sync_status` を更新します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.9.2. syncNow

指定したリポジトリのミラーリング設定の `sync_status` を更新します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.9.3. getRepoMirrorConfig

指定したリポジトリのミラーリング設定を返します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	ViewMirrorConfig
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.9.4. changeRepoMirrorConfig

ユーザーがリポジトリのミラーリング設定を変更できるようにします。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリミラーリング設定を更新します。

名前	説明	スキーマ
is_enabled 任意	同期を有効または無効にするために使用されます。	boolean
external_reference 任意	外部リポジトリの場所。	string
external_registry_username 任意	外部レジストリーでの認証に使用されるユーザー名。	
external_registry_password 任意	外部レジストリーでの認証に使用されるパスワード。	
sync_start_date 任意	このリポジトリの次の同期時間を決定します。	string
sync_interval 任意	next_start_date が同期を開始するまでの秒数。	integer
robot_username 任意	イメージのプッシュに使用されるロボットのユーザー名。	string
root_rule 任意	同期する必要のあるタグの判別に使用する glob パターンのリスト。	object
external_registry_config 任意		object

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.9.5. createRepoMirrorConfig

特定のリポジトリの RepoMirrorConfig を作成します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリのミラーリング設定を作成します。

名前	説明	スキーマ
is_enabled 任意	同期を有効または無効にするために使用されます。	boolean
external_reference 任意	外部リポジトリの場所。	string
external_registry_username 任意	外部レジストリーでの認証に使用されるユーザー名。	
external_registry_password 任意	外部レジストリーでの認証に使用されるパスワード。	
sync_start_date 任意	このリポジトリの次の同期時間を決定します。	string

名前	説明	スキーマ
sync_interval 任意	next_start_date が同期を開始するまでの秒数。	integer
robot_username 任意	イメージのプッシュに使用されるロボットのユーザー名。	string
root_rule 任意	同期する必要のあるタグの判別に使用する glob パターンのリスト。	object
external_registry _config 任意		object

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10. NAMESPACEQUOTA

2.10.1. listUserQuota

GET /api/v1/user/quota

承認: oauth2_implicit(user:admin)

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.2. getOrganizationQuotaLimit

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	limit_id 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.3. changeOrganizationQuotaLimit

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	limit_id 必須		string
path	orgname 必須		string

要求の body スキーマ (application/json)
変更する組織のクォータ制限の説明

名前	説明	スキーマ
type 任意	割り当て制限のタイプ: 警告または拒否	string
threshold_percent 必須	クォータのしきい値 (クォータのパーセント単位)	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.4. deleteOrganizationQuotaLimit

DELETE /api/v1/organization/{orgname}/quota/{quota_id}/limit/{limit_id}
承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	limit_id 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.5. createOrganizationQuotaLimit

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	orgname 必須		string

要求の body スキーマ (application/json)

新しい組織のクォータ制限の説明

名前	説明	スキーマ
type 任意	割り当て制限のタイプ: 警告または拒否	string
threshold_percent 必須	クォータのしきい値 (クォータのパーセント単位)	integer

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.6. listOrganizationQuotaLimit

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.7. getUserQuotaLimit

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	limit_id 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.8. listUserQuotaLimit

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.9. getOrganizationQuota

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.10. changeOrganizationQuota

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	orgname 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.11. deleteOrganizationQuota

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.12. createOrganizationQuota

新しい組織のクォータを作成します。

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError

HTTP コード	Description	スキーマ
404	結果が見つからない	ApiError

2.10.13. listOrganizationQuota

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.10.14. getUserQuota

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	quota_id 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11. 組織

組織、メンバーおよび OAuth アプリケーションを管理します。

2.11.1. createOrganization

新しい組織を作成します。

POST /api/v1/organization/

承認: `oauth2_implicit(user:admin)`

要求の body スキーマ (application/json)

新しい組織の説明。

名前	説明	スキーマ
name 任意	組織のユーザー名	string
email 任意	組織の連絡先メール	string
recaptcha_response 任意	検証用の recaptcha 応答コード (無効となっている場合もある)。	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.2. validateProxyCacheConfig

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須		string

要求の body スキーマ (application/json)

組織のプロキシキャッシュ設定

名前	説明	スキーマ
upstream_registry 任意	キャッシュされるアップストリームレジストリーの名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.3. getOrganizationCollaborators

指定した組織の外部コラボレーターをリスト表示します。

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

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.4. getOrganizationApplication指定の組織、指定の `client_id` のアプリケーションを取得します。**GET /api/v1/organization/{orgname}/applications/{client_id}**承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	client_id 必須	OAuth クライアント ID	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.5. updateOrganizationApplication

この組織のアプリケーションを更新します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	client_id 必須	OAuth クライアント ID	string
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

更新アプリケーションの説明。

名前	説明	スキーマ
name 任意	アプリケーションの名前	string
redirect_uri 任意	アプリケーションの OAuth リダイレクトの URI	string
application_uri 任意	アプリケーションのホームページの URI	string
description 任意	アプリケーションの人間が判読できる説明	string
avatar_email 任意	アプリケーションに使用するアバターのメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.6. deleteOrganizationApplication

この組織のアプリケーションを削除します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	client_id 必須	OAuth クライアント ID	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.7. createOrganizationApplication

この組織に新しいアプリケーションを作成します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

新しい組織アプリケーションを説明します。

名前	説明	スキーマ
name 任意	アプリケーションの名前	string
redirect_uri 任意	アプリケーションの OAuth リダイレクトの URI	string
application_uri 任意	アプリケーションのホームページの URI	string
description 任意	アプリケーションの人間が判読できる説明	string
avatar_email 任意	アプリケーションに使用するアバターのメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.8. getOrganizationApplications

指定した組織のアプリケーションをリスト表示します。

GET /api/v1/organization/{orgname}/applications承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.9. getProxyCacheConfig

組織のプロキシキャッシュ設定を取得します。

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.10. deleteProxyCacheConfig

組織のプロキシキャッシュ設定を削除します。

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.11. createProxyCacheConfig

組織のプロキシキャッシュ設定を作成します。

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

組織のプロキシーキャッシュ設定

名前	説明	スキーマ
upstream_registry 必須	キャッシュされるアップストリームレジストリーの名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.12. getOrganizationMember

組織のメンバーの詳細を取得します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	membername 必須	組織メンバーのユーザー名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.13. removeOrganizationMember

組織からメンバーを削除し、そのリポジトリの特権をすべて破棄して組織の全チームから削除します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	membername 必須	組織メンバーのユーザー名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.14. getOrganizationMembers

指定した組織のメンバー (人間) をリスト表示します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.15. getOrganization

指定した組織の詳細を取得します。

GET /api/v1/organization/{orgname}

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.16. changeOrganizationDetails

指定した組織の詳細を変更します。

PUT /api/v1/organization/{orgname}

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

既存組織の更新の説明

名前	説明	スキーマ
email 任意	組織の連絡先メール	string
invoice_email 任意	組織が請求書のメールを受信する必要があるかどうか	boolean
invoice_email_ad dress 任意	請求書を受信するメールアドレス	
tag_expiration_s 任意	タグの有効期限 (秒)	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.17. deleteAdmindedOrganization

指定した組織を削除します。

DELETE /api/v1/organization/{orgname}

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.11.18. getApplicationInformation

指定したアプリケーションに関する情報を取得します。

GET /api/v1/app/{client_id}

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	client_id 必須	OAuth クライアント ID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12. パーミッション

リポジトリのパーミッションを管理します。

2.12.1. getUserTransitivePermission

指定したユーザーのパーミッションを取得します。

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

パスパラメーター

タイプ	名前	説明	スキーマ
path	username 必須	パーミッションが適用されるユーザー名	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.2. getUserPermissions

指定したユーザーのパーミッションを取得します。

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

パスパラメーター

タイプ	名前	説明	スキーマ
path	username 必須	パーミッションが適用されるユーザーのユーザー名	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.3. changeUserPermissions

既存リポジトリのパーミッションを更新します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	username 必須	パーミッションが適用されるユーザーのユーザー名	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

ユーザーパーミッションの説明。

名前	説明	スキーマ
role 任意	ユーザーに使用するロール	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.4. deleteUserPermissions

ユーザーのパーミッションを削除します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	username 必須	パーミッションが適用されるユーザーのユーザー名	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.5. getTeamPermissions

指定したチームのパーミッションをフェッチします。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	teamname 必須	パーミッションが適用されるチームの名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.6. changeTeamPermissions

既存のチームパーミッションを更新します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	teamname 必須	パーミッションが適用されるチームの名前	string

要求の body スキーマ (application/json)

チームパーミッションの説明。

名前	説明	スキーマ
role 任意	チームに使用するロール	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.7. deleteTeamPermissions

指定したチームのパーミッションを削除します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	teamname 必須	パーミッションが適用されるチームの名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.8. listRepoTeamPermissions

すべてのチームパーミッションをリスト表示します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリーの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.12.9. listRepoUserPermissions

全ユーザーパーミッションをリスト表示します。

GET /api/v1/repository/{repository}/permissions/user/

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリーの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13. POLICY

2.13.1. listOrganizationAutoPrunePolicies

組織の自動プルーニングポリシーを一覧表示します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.2. createOrganizationAutoPrunePolicy

組織の自動プルーニングポリシーを作成します。

POST /api/v1/organization/{orgname}/autoprunepolicy/承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

ユーザーの namespace に適用されるポリシー設定

名前	説明	スキーマ
method 任意	タグのプルーニングに使用するメソッド (number_of_tags、creation_date)	string
value 任意	プルーニング方法に使用する値 (タグの数 (例: 10)、時間デルタ (例: 7d (7 日)))	

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.3. deleteOrganizationAutoPrunePolicy

組織の自動プルーニングポリシーを削除します。

DELETE /api/v1/organization/{orgname}/autoprunepolicy/{policy_uuid}承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

タイプ	名前	説明	スキーマ
path	policy_uuid required	ポリシーの一意の ID	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.4. updateOrganizationAutoPrunePolicy

組織の自動プルーニングポリシーを更新します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string
path	policy_uuid required	ポリシーの一意の ID	string

要求の body スキーマ (application/json)

ユーザーの namespace に適用されるポリシー設定

名前	説明	スキーマ
method 任意	タグのプルーニングに使用するメソッド (number_of_tags、creation_date)	string
value 任意	プルーニング方法に使用する値 (タグの数 (例: 10)、時間デルタ (例: 7d (7 日)))	

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.5. getOrganizationAutoPrunePolicy

組織の自動プルーニングポリシーを取得します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string
path	policy_uuid required	ポリシーの一意の ID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.6. listUserAutoPrunePolicies

現在ログインしているユーザーの自動プルーニングポリシーを一覧表示します。

GET /api/v1/user/autoprunepolicy/承認: `oauth2_implicit(user:admin)`

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.7. createUserAutoPrunePolicy

現在ログインしているユーザーの自動プルーニングポリシーを作成します。

POST /api/v1/user/autoprunepolicy/承認: `oauth2_implicit(user:admin)`要求の body スキーマ (`application/json`)

ユーザーの namespace に適用されるポリシー設定

名前	説明	スキーマ
method 任意	タグのプルーニングに使用するメソッド (<code>number_of_tags</code> 、 <code>creation_date</code>)	string
value 任意	プルーニング方法に使用する値 (タグの数 (例: 10)、時間デルタ (例: 7d (7 日)))	

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError

HTTP コード	Description	スキーマ
404	結果が見つからない	ApiError

2.13.8. deleteUserAutoPrunePolicy

現在ログインしているユーザーの自動プルーニングポリシーを削除します。

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

承認: oauth2_implicit(user:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	policy_uuid required	ポリシーの一意的 ID	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.9. updateUserAutoPrunePolicy

現在ログインしているユーザーの自動プルーニングポリシーを更新します。

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

承認: oauth2_implicit(user:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	policy_uuid required	ポリシーの一意的 ID	string

要求の body スキーマ (application/json)

ユーザーの namespace に適用されるポリシー設定

名前	説明	スキーマ
method 任意	タグのプルーニングに使用するメソッド (number_of_tags、 creation_date)	string
value 任意	プルーニング方法に使用する値 (タグの数 (例: 10)、時間デルタ (例: 7d (7 日)))	

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.13.10. getUserAutoPrunePolicy

現在ログインしているユーザーの自動プルーニングポリシーを取得します。

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	policy_uuid required	ポリシーの一意的 ID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError

HTTP コード	Description	スキーマ
404	結果が見つからない	ApiError

2.14. PROTOTYPE

リポジトリに追加されたデフォルトのパーミッションを管理します。

2.14.1. updateOrganizationPrototypePermission

既存のパーミッションプロトタイプのロールを更新します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	prototypeid 必須	プロトタイプの ID	string
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

新しいプロトタイプロールの説明

名前	説明	スキーマ
role 任意	パーミッションに適用する必要のあるロール	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.14.2. deleteOrganizationPrototypePermission

既存のパーミッションのプロトタイプを削除します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	prototypeid 必須	プロトタイプの ID	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.14.3. createOrganizationPrototypePermission

新規パーミッションのプロトタイプを作成します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

要求の body スキーマ (`application/json`)

新しいプロトタイプの説明

名前	説明	スキーマ
role 任意	委譲に適用する必要があるロール	string
activating_user 任意	ルールが適用されるユーザーを作成するリポジトリー	object
delegate 任意	ルールでアクセス権を付与するユーザーまたはチームに関する情報	object

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.14.4. getOrganizationPrototypePermissions

この組織の既存のプロトタイプをリスト表示します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15. REPOSITORY

リポジトリをリスト表示、作成、管理します。

2.15.1. createRepo

新規リポジトリを作成します。

POST /api/v1/repository

承認: `oauth2_implicit(repo:create)`

要求の **body** スキーマ (`application/json`)

新規リポジトリの説明

名前	説明	スキーマ
repository 必須	リポジトリ名	string
visibility 必須	リポジトリの初期の表示内容	string
namespace 必須	リポジトリを作成する namespace。省略すると、呼び出し元のユーザー名が使用されます。	string
description 必須	リポジトリの Markdown でエンコードされた説明	string
repo_kind 任意	リポジトリの種類	

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.2. listRepos

さまざまな状況で現在のユーザーに表示されるリポジトリのリストを取得します。

GET /api/v1/repository

承認: `oauth2_implicit(repo:read)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string
query	repo_kind 任意	返すリポジトリの種類	string
query	popularity 任意	リポジトリの人気指標を含めるかどうか。	boolean
query	last_modified 任意	リポジトリの最終変更時を含めるかどうか。	boolean
query	public 任意	公開されているリポジトリをユーザーに表示するように追加します。	boolean
query	starred 任意	ユーザーがスターをつけたリポジトリを返すようにフィルタリングします。	boolean
query	namespace 任意	この namespace に返されたリポジトリをフィルターします。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.3. changeRepoVisibility

リポジトリの可視性を変更します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリの可視性を変更します。

名前	説明	スキーマ
visibility 任意	リポジトリの初期の表示内容	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.4. changeRepoState

リポジトリの状態を変更します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリの状態を変更します。

名前	説明	スキーマ
state 任意	プッシュが許可されるかどうかを決定します。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.5. getRepo

指定したリポジトリを取得します。

GET /api/v1/repository/{repository}

承認: oauth2_implicit(repo:read)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	includeTags 任意	リポジトリタグを含めるかどうか	boolean
query	includeStats 任意	アクション統計を含めるかどうか。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.6. updateRepo

指定したリポジトリにある説明を更新します。

PUT /api/v1/repository/{repository}

承認: `oauth2_implicit (repo:write)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリで更新できるフィールド。

名前	説明	スキーマ
description 任意	リポジトリの Markdown でエンコードされた説明	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.15.7. deleteRepository

リポジトリを削除します。

DELETE /api/v1/repository/{repository}

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16. REPOSITORYNOTIFICATION

リポジトリイベント/通知をリスト表示、作成、管理します。

2.16.1. testRepoNotification

このリポジトリのテスト通知をキューに入れます。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	uuid 必須	通知の UUID	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16.2. getRepoNotification

指定した通知の情報を取得します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	uuid 必須	通知の UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16.3. deleteRepoNotification

指定した通知を削除します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	uuid 必須	通知の UUID	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16.4. resetRepositoryNotificationFailures

リポジトリの失敗通知を 0 件にリセットします。

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

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	uuid 必須	通知の UUID	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16.5. createRepoNotification**POST /api/v1/repository/{repository}/notification/**承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

リポジトリでの通知作成に関する情報

名前	説明	スキーマ
event 任意	通知が応答するイベント	string

名前	説明	スキーマ
method 任意	通知の方法 (メールや Web コールバックなど)	string
config 任意	特定の通知メソッドに関する JSON 設定情報	object
eventConfig 任意	特定の通知イベントに関する JSON 設定情報	object
title 任意	人間が判読できる通知のタイトル	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.16.6. listRepoNotifications

指定したリポジトリの通知をリスト表示します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.17. REPOTOKEN

リポジトリアクセストークンの管理 (非推奨)

2.17.1. getTokens

指定したリポジトリトークン情報を取得します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	code 必須	トークンコード	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.17.2. changeToken

指定したリポジトリトークンのパーミッションを更新します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	code 必須	トークンコード	string

要求の body スキーマ (application/json)

トークンパーミッションの説明

名前	説明	スキーマ
role 任意	トークンに使用するロール	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.17.3. deleteToken

リポジトリトークンを削除します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
-----	----	----	------

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	code 必須	トークンコード	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.17.4. createToken

新規リポジトリトークンを作成します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

新しいトークンの説明。

名前	説明	スキーマ
friendlyName 任意	トークンの特定に役立つ分かりやすい名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.17.5. listRepoTokens

指定したリポジトリのトークンをリスト表示します。

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

承認: oauth2_implicit(repo:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18. ROBOT

ユーザーおよび組織のロボットアカウントを管理します。

2.18.1. getUserRobots

ユーザーが利用可能なロボットをリスト表示します。

GET /api/v1/user/robots承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	limit 任意	これが指定されている場合には、返すべきロボットの数。	integer
query	token 任意	False の場合には、ロボットのトークンは返されません。	boolean
query	permissions 任意	ロボットのパーミッションがあるリポジトリおよびチームを含めるかどうか。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.2. getOrgRobotPermissions

組織のロボットのリポジトリパーミッションリストを返します。

GET /api/v1/organization/{orgname}/robots/{robot_shortname}/permissions承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.3. regenerateOrgRobotToken

組織のロボットのトークンを再生成します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.4. getUserRobotPermissions

ユーザーのロボットのリポジトリパーミッションリストを返します。

GET /api/v1/user/robots/{robot_shortname}/permissions承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.5. regenerateUserRobotToken

ユーザーのロボットのトークンを再生成します。

POST /api/v1/user/robots/{robot_shortname}/regenerate承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.6. getOrgRobot

指定された名前の組織のロボットを返します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.7. createOrgRobot

組織に新しいロボットを作成します。

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

承認: oauth2_implicit(org:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

ロボット作成のデータ (任意)

名前	説明	スキーマ
description 任意	ロボットのテキスト形式の説明 (任意)	string
unstructured_metadata 任意	ロボットの非構造化メタデータ (任意)	object

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.8. deleteOrgRobot

既存の組織ロボットを削除します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
-----	----	----	------

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.9. getOrgRobots

組織のロボットをリスト表示します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	orgname 必須	組織の名前	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	limit 任意	これが指定されている場合には、返すべきロボットの数。	integer
query	token 任意	False の場合には、ロボットのトークンは返されません。	boolean

タイプ	名前	説明	スキーマ
query	permissions 任意	ロボットのパーミッションがあるリポジトリおよびチームを含めるかどうか。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.10. getUserRobot

指定の名前のユーザーのロボットを返します。

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.11. createUserRobot

指定の名前で新たなユーザーロボットを作成します。

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

承認: oauth2_implicit(user:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string

要求の body スキーマ (application/json)

ロボット作成のデータ (任意)

名前	説明	スキーマ
description 任意	ロボットのテキスト形式の説明 (任意)	string
unstructured_metadata 任意	ロボットの非構造化メタデータ (任意)	object

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.18.12. deleteUserRobot

既存のロボットを削除します。

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

承認: oauth2_implicit(user:admin)

パスパラメーター

タイプ	名前	説明	スキーマ
path	robot_shortname 必須	ユーザーまたは組織の接頭辞を含まない、ロボットの省略名	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.19. SEARCH

すべてのレジストリーコンテキストに対して検索を実行します。

2.19.1. conductRepoSearch

指定したクエリーに一致するアプリケーションおよびリポジトリーのリストを取得します。

GET /api/v1/find/repositories

承認:

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	includeUsage 任意	使用状況のメタデータを含めるかどうか。	boolean
query	page 任意	ページ。	integer
query	query 任意	検索クエリー	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.19.2. conductSearch

指定したクエリーに一致するエンティティおよびリソースのリストを取得します。

GET /api/v1/find/all

承認: `oauth2_implicit(repo:read)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	query 任意	検索クエリー	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.19.3. getMatchingEntities

指定した接頭辞に一致するエンティティのリストを取得します。

GET /api/v1/entities/{prefix}

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	prefix 必須		string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	includeOrgs 任意	組織名を含めるかどうか。	boolean
query	includeTeams 任意	チーム名を含めるかどうか。	boolean
query	namespace 任意	組織エンティティのクエリー時に使用する namespace。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.20. SECSCAN

リポジトリの脆弱性およびその他のセキュリティ情報をリスト表示および管理します。

2.20.1. getRepoManifestSecurity

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

タイプ	名前	説明	スキーマ
path	manifestref 必須	マニフェストのダイジェスト	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	vulnerabilities 任意	脆弱性情報を含めます	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21. スーパーユーザー

スーパーユーザー API。

2.21.1. createInstallUser

新しいユーザーを作成します。

POST /api/v1/superuser/users/

承認: `oauth2_implicit(super:user)`

要求の body スキーマ (application/json)

ユーザー作成のデータ

名前	説明	スキーマ
username 任意	作成されるユーザーの名前	string
email 任意	作成中のユーザーのメールアドレス	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.2. listAllUsers

システム内の全ユーザーリストを返します。

GET /api/v1/superuser/users/

承認: `oauth2_implicit(super:user)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string
query	limit 任意	ページごとに返す結果の数を制限します。最大 100 です。	integer
query	disabled 任意	false の場合には、有効なユーザーのみが返されま す。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.3. listAllLogs

現在のシステムの使用状況ログをリスト表示します。

GET /api/v1/superuser/logs

承認: oauth2_implicit(super:user)

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string
query	page 任意	ログのページ番号	integer
query	endtime 任意	最後にログを取得した時間 (%m/%d/%Y %Z)	string
query	starttime 任意	最初にログを取得する時間 (%m/%d/%Y %Z)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.4. createServiceKey

POST /api/v1/superuser/keys

承認: oauth2_implicit(super:user)

要求の body スキーマ (application/json)

サービスキーの作成の説明

名前	説明	スキーマ
service 任意	このキーで認証されるサービス	string

名前	説明	スキーマ
name 任意	サービスキーの判別しやすい名前	string
metadata 任意	このキーのメタデータのキー/値のペア	object
notes 任意	指定すると、キーにメモを追加で指定できます。	string
expiration 任意	unix タイムスタンプとしての有効期限	

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.5. listServiceKeys

GET /api/v1/superuser/keys

承認: `oauth2_implicit(super:user)`

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.6. changeUserQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string
path	quota_id 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.7. deleteUserQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

タイプ	名前	説明	スキーマ
path	quota_id 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.8. createUserQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
201	作成に成功	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.9. listUserQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.10. changeOrganizationQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

タイプ	名前	説明	スキーマ
path	quota_id 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.11. deleteOrganizationQuotaSuperUser

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string
path	quota_id 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.12. createOrganizationQuotaSuperUser

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

要求の body スキーマ (application/json)

新しい組織の割り当ての説明

名前	説明	スキーマ
limit_bytes 任意	組織で許可されているバイト数	integer

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.13. listOrganizationQuotaSuperUser

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.14. changeOrganization

指定したユーザーの情報を更新します。

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	name 必須	管理する組織の名前	string

要求の body スキーマ (application/json)

既存組織の更新の説明

名前	説明	スキーマ
email 任意	組織の連絡先メール	string

名前	説明	スキーマ
invoice_email 任意	組織が請求書のメールを受信する必要があるかどうか	boolean
invoice_email_ad dress 任意	請求書を受信するメールアドレス	
tag_expiration_s 任意	タグの有効期限 (秒)	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.15. deleteOrganization

指定した組織を削除します。

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	name 必須	管理する組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError

HTTP コード	Description	スキーマ
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.16. approveServiceKey

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	kid 必須	サービスキーの一意識別子	string

要求の body スキーマ (application/json)

サービスキーの承認に関する情報

名前	説明	スキーマ
notes 任意	オプションの承認に関する注意事項	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.17. deleteServiceKey

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	kid 必須	サービスキーの一意識別子	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.18. updateServiceKey

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	kid 必須	サービスキーの一意識別子	string

要求の body スキーマ (application/json)

サービスキーの更新の説明

名前	説明	スキーマ
name 任意	サービスキーの判別しやすい名前	string
metadata 任意	このキーのメタデータのキー/値のペア	object
expiration 任意	unix タイムスタンプとしての有効期限	

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.19. getServiceKey

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

パスパラメーター

タイプ	名前	説明	スキーマ
path	kid 必須	サービスキーの一意識別子	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.20. getRepoBuildStatusSuperUser

ビルドの uuid で指定したビルドのステータスを返します。

GET /api/v1/superuser/{build_uuid}/status
承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.21. getRepoBuildSuperUser

ビルドに関する情報を返します。

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

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.22. getRepoBuildLogsSuperUser

ビルド uuid で指定したビルドのビルドログを返します。

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

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	build_uuid 必須	ビルドの UUID	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.23. getRegistrySize

GET /api/v1/superuser/registrysize/

承認: oauth2_implicit(super:user)

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

要求の body スキーマ (application/json)

イメージレジストリーサイズの説明

名前	説明	スキーマ
size_bytes* optional	組織で許可されているバイト数	integer
last_ran		integer

名前	説明	スキーマ
queued		boolean
running		boolean

応答

HTTP コード	Description	スキーマ
201	作成済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.21.24. postRegistrySize

POST /api/v1/superuser/registrysize/

承認: `oauth2_implicit(super:user)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	namespace 必須		string

要求の body スキーマ (application/json)

イメージレジストリーサイズの説明

名前	説明	スキーマ
size_bytes* optional	組織で許可されているバイト数	integer
last_ran		integer
queued		boolean
running		boolean

応答

HTTP コード	Description	スキーマ
201	作成済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.22. TAG

リポジトリのタグを管理します。

2.22.1. restoreTag

リポジトリの以前のイメージに、リポジトリタグを復元します。

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

承認: oauth2_implicit (repo:write)

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	tag 必須	タグの名前	string

要求の body スキーマ (application/json)

タグを特定のイメージに復元します。

名前	説明	スキーマ
manifest_digest 任意	指定されている場合には、使用する必要があるマニフェストダイジェスト。	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.22.2. changeTag

タグが参照するイメージを変更するか、新規タグを作成します。

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

承認: `oauth2_implicit (repo:write)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	tag 必須	タグの名前	string

要求の body スキーマ (application/json)

特定のタグを変更します。

名前	説明	スキーマ
manifest_digest 任意	(指定されている場合) タグが参照すべきマニフェストダイジェスト	
expiration 任意	(指定されている場合) イメージの有効期限	

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.22.3. deleteFullTag

指定したリポジトリタグを削除します。

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

承認: `oauth2_implicit(repo:write)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string
path	tag 必須	タグの名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.22.4. listRepoTags

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

承認: `oauth2_implicit(repo:read)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	onlyActiveTags 任意	アクティブなタグだけに絞り込みます。	boolean
query	page 任意	結果のページインデックス。デフォルトは1です。	integer
query	limit 任意	ページごとに返す結果の数を制限します。最大 100 です。	integer
query	filter_tag_name 任意	構文: <op><name> 操作に基づいてタグ名をフィルターします。<op> には like または eq を指定できます。	string
query	specificTag 任意	特定のタグだけに絞り込みます。	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23. TEAM

組織のチームを作成、リスト表示、および管理します。

2.23.1. getOrganizationTeamPermissions

組織のチームのリポジトリパーミッションのリストを返します。

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

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.2. updateOrganizationTeamMember

既存のチームにメンバーを追加または調整します。

PUT /api/v1/organization/{orgname}/team/{teamname}/members/{membername}承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	membername 必須	チームメンバーのユーザー名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.3. deleteOrganizationTeamMember

チームのメンバーを削除します。

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}
承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	membername 必須	チームメンバーのユーザー名	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError

HTTP コード	Description	スキーマ
404	結果が見つからない	ApiError

2.23.4. getOrganizationTeamMembers

指定したチームのメンバーのリストを取得します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	orgname 必須	組織の名前	string

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	includePending 任意	保留中のメンバーを含めるかどうか。	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.5. inviteTeamMemberEmail

既存のチームに招待するメールアドレスを追加します。

PUT /api/v1/organization/{orgname}/team/{teamname}/invite/{email}承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	email 必須		string
path	teamname 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.6. deleteTeamMemberEmailInvite

チームへの参加を招待するメールアドレスを削除します。

DELETE /api/v1/organization/{orgname}/team/{teamname}/invite/{email}承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	email 必須		string
path	teamname 必須		string
path	orgname 必須		string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.7. updateOrganizationTeam

指定したチームの組織全体のパーミッションを更新します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	orgname 必須	組織の名前	string

要求の body スキーマ (application/json)

チームの説明

名前	説明	スキーマ
role 任意	チームに適用する必要がある組織全体のパーミッション	string
description 任意	チームの Markdown の説明	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.23.8. deleteOrganizationTeam

指定したチームを削除します。

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

承認: `oauth2_implicit(org:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	teamname 必須	チームの名前	string
path	orgname 必須	組織の名前	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24. TRIGGER

ビルドトリガーを作成、リスト表示、管理します。

2.24.1. activateBuildTrigger

指定したビルドトリガーをアクティブにします。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

名前	説明	スキーマ
config 任意	任意の json	object
pull_robot 任意	イメージのプルに使用するロボットの名前	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.2. listTriggerRecentBuilds

指定したトリガーで開始するビルドをリスト表示します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

クエリパラメーター

タイプ	名前	説明	スキーマ
query	limit 任意	返すビルドの最大数	integer

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.3. manuallyStartBuildTrigger

指定したトリガーからビルドを手動で開始します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

ビルドトリガーをアクティベートするための実行パラメーター (任意)

名前	説明	スキーマ
branch_name 任意	(SCM のみ) 指定した場合のビルドするブランチの名前。	string
commit_sha 任意	(カスタムのみ) 指定されている場合は、git リポジトリのチェックアウトに使用する ref/SHA1。	string
refs 任意	(SCM のみ) 指定されている場合のビルドの ref。	

応答

HTTP コード	Description	スキーマ
201	作成に成功	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.4. getBuildTrigger

指定したビルドトリガーの情報を取得します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.5. updateBuildTrigger

指定したビルドトリガーを更新します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

要求の body スキーマ (application/json)

ビルドトリガーを更新するオプション

名前	説明	スキーマ
enabled 任意	ビルドトリガーが有効かどうか	boolean

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError

HTTP コード	Description	スキーマ
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.6. deleteBuildTrigger

指定したビルドトリガーを削除します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	trigger_uuid 必須	ビルドトリガーの UUID	string
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.24.7. listBuildTriggers

指定したリポジトリのトリガーをリスト表示します。

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

承認: `oauth2_implicit(repo:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.25. USER

現行ユーザーを管理します。

2.25.1. createStar

リポジトリを表示します。

POST /api/v1/user/starred

承認: `oauth2_implicit(repo:read)`

要求の body スキーマ (application/json)

名前	説明	スキーマ
namespace 任意	リポジトリが属する namespace	string
repository 任意	リポジトリ名	string

レスポンス

HTTP コード	Description	スキーマ
201	作成に成功	

HTTP コード	Description	スキーマ
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.25.2. listStarredRepos

星付きのリポジトリをすべてリスト表示します。

GET /api/v1/user/starred

承認: `oauth2_implicit(user:admin)`

クエリーパラメーター

タイプ	名前	説明	スキーマ
query	next_page 任意	次のページのページトークン	string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.25.3. getLoggedInUser

認証ユーザーのユーザー情報を取得します。

GET /api/v1/user/

承認: `oauth2_implicit (user:read)`

応答

HTTP コード	Description	スキーマ
200	正常な呼び出し	UserView
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.25.4. deleteStar

リポジトリから星を削除します。

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

承認: `oauth2_implicit(user:admin)`

パスパラメーター

タイプ	名前	説明	スキーマ
path	repository 必須	リポジトリの完全パス (例: namespace/name)	string

レスポンス

HTTP コード	Description	スキーマ
204	削除済み	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.25.5. getUserInformation

指定したユーザーのユーザー情報を取得します。

GET /api/v1/users/{username}

承認:

パスパラメーター

タイプ	名前	説明	スキーマ
path	username 必須		string

レスポンス

HTTP コード	Description	スキーマ
200	正常な呼び出し	
400	Bad Request (不適切な要求)	ApiError
401	セッションが必要	ApiError
403	不正アクセス	ApiError
404	結果が見つからない	ApiError

2.26. 定義

2.26.1. ApiError

名前	説明	スキーマ
status 任意	応答のステータスコード	integer
type 任意	エラーのタイプへの参照。	string
detail 任意	エラーの特定インスタンスについての詳細。	string
title 任意	エラーのタイプを識別する一意のエラーコード。	string
error_message 任意	非推奨。	string
error_type 任意	非推奨。	string

2.26.2. UserView

名前	説明	スキーマ
verified 任意	ユーザーのメールアドレスが検証済みかどうか。	boolean
anonymous 任意	このユーザーデータがゲストユーザーを表す場合は True	boolean
email 任意	ユーザーのメールアドレス	string
avatar 任意	ユーザーのアイコンを表すアバターデータ	object
organizations 任意	ユーザーがメンバーとなっている組織に関する情報	object の配列。
logins 任意	ユーザーが認証された外部ログインプロバイダーのリスト	object の配列。
can_create_repo 任意	ユーザーにリポジトリ作成のパーミッションが割り当てられているかどうか。	boolean
preferred_names pace 任意	true の場合、ユーザーの namespace が表示されることが優先される namespace。	boolean

2.26.3. ViewMirrorConfig

名前	説明	スキーマ
is_enabled 任意	同期を有効または無効にするために使用されます。	boolean
external_referen ce 任意	外部リポジトリの場所。	string
external_registry _username 任意	外部レジストリーでの認証に使用されるユーザー名。	
external_registry _password 任意	外部レジストリーでの認証に使用されるパスワード。	
sync_start_date 任意	このリポジトリの次の同期時間を決定します。	string

名前	説明	スキーマ
sync_interval 任意	next_start_date が同期を開始するまでの秒数。	integer
robot_username 任意	イメージのプッシュに使用されるロボットのユーザー名。	string
root_rule 任意	同期する必要のあるタグの判別に使用する glob パターンのリスト。	object
external_registry _config 任意		object

2.26.4. ApiErrorDescription

名前	説明	スキーマ
type 任意	エラータイプリソースへの参照。	string
title 任意	エラーのタイトル。エラーの種類を一意に識別するために使用できます。	string
description 任意	問題の修正時に役立つ可能性のある詳細な説明。	string

第3章 API 設定の例

3.1. EXTERNAL_REGISTRY_CONFIG オブジェクトリファレンス

```
{
  "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 オブジェクトリファレンス

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