



The REST API is now versioned. For more information, see "About API versioning."

# **Packages**

GitHub Docs

Use the REST API to interact with GitHub Packages.

# About GitHub Packages &

GitHub Packages supports a range of package managers for publishing packages. For more information, see "Introduction to GitHub Packages."

After you publish a package, you can use the REST API to manage the package in your GitHub repositories and organizations. For more information, see "<u>Deleting and restoring a package</u>."

To use the REST API to manage GitHub Packages, you must authenticate using a personal access token (classic).

- To access package metadata, your token must include the read:packages scope.
- To delete packages and package versions, your token must include the read:packages and delete:packages scopes.
- To restore packages and package versions, your token must include the read:packages and write:packages scopes.

If your package is in a registry that supports granular permissions, then your token does not need the repo scope to access or manage this package. If your package is in a registry that only supports repository-scoped permissions, then your token must also include the repo scope since your package inherits permissions from a GitHub repository. For a list of registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Get list of conflicting packages during Docker migration for organization *₽*

Lists all packages that are in a specific organization, are readable by the requesting user, and that encountered a conflict during a Docker migration. To use this endpoint, you must authenticate using an access token with the read:packages scope.

Parameters for "Get list of conflicting packages during Docker migration for organization"

## **Headers**

accept string

Setting to application/vnd.github+json is recommended.

# Path parameters

org string Required

The organization name. The name is not case sensitive.

# HTTP response status codes for "Get list of conflicting packages during Docker migration for organization"

Status code	Description
200	OK
401	Requires authentication
403	Forbidden

# Code samples for "Get list of conflicting packages during Docker migration for organization"



#### Response

```
Example response Response schema

Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": "MDEy0k9yZZFuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.githubusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/github", "followers_url": "https://HOSTNAME/users/github/followers", "following_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/susers/github/orgs", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "**

**Subscriptions_url": "https://HOSTNAME/users/github/orgs", "**

**Towner**: "avatar_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "**

**Towner**: "avatar_url": "https://HOSTNAME/users/github/orgs", "**

**Towner**: "bello_docker", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": "", "url": "https://HOSTNAME/users_id": "", "url": "https://HOSTNAME/users_id": "", "url": "https://HOSTNAME/users_id": "", "url": "https://HOSTNAME/users/github/orgs", "**

**Towner**: "bello_docker", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": "", "url": "", "url":
```

# List packages for an organization &

Lists packages in an organization readable by the user.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "List packages for an organization"

#### **Headers**

accept string
Setting to application/vnd.github+json is recommended.

#### **Path parameters**

org string Required

The organization name. The name is not case sensitive.

#### **Query parameters**

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of:  $\ensuremath{\mathsf{npm}}$  ,  $\ensuremath{\mathsf{maven}}$  ,  $\ensuremath{\mathsf{rubygems}}$  ,  $\ensuremath{\mathsf{docker}}$  ,  $\ensuremath{\mathsf{nuget}}$  ,  $\ensuremath{\mathsf{container}}$ 

# visibility string

The selected visibility of the packages. This parameter is optional and only filters an existing result set.

The internal visibility is only supported for GitHub Packages registries that allow for granular permissions. For other ecosystems internal is synonymous with private. For the list of GitHub Packages registries that support granular permissions, see "About permissions for GitHub Packages."

Can be one of: public , private , internal

page integer

Page number of the results to fetch.

Default: 1

per\_page integer

The number of results per page (max 100).

Default: 30

## HTTP response status codes for "List packages for an organization"

Status code	Description
200	ОК
400	The value of per_page multiplied by page cannot be greater than 10000.
401	Requires authentication
403	Forbidden

# Code samples for "List packages for an organization"



#### Response

Example response Response schema

```
Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": "MDEyOk9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.githubusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/github", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "gists_url": "https://HOSTNAME/users/github/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "
```

# Get a package for an organization &

Gets a specific package in an organization.

To use this endpoint, you must authenticate using an access token with the read:packages scope. If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "Get a package for an organization"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

## Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

org string Required

The organization name. The name is not case sensitive.

## HTTP response status codes for "Get a package for an organization"

Status code Description

200 OK

## Code samples for "Get a package for an organization"

CURL JavaScript GitHub CLI

CURL JavaScript GitHub CLI

CURL JavaScript GitHub CLI

CURL JavaScript GitHub CLI

CURL JavaScript GitHub-Ani-Version: 2022-11-28"

```
http(s)://HOSTNAME/api/v3/orgs/ORG/packages/PACKAGE_TYPE/PACKAGE_NAME
```

#### Response

```
Example response Response schema

Status: 200

{ "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": "MDEy0k9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.githubusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/github", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "following_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": ""url": "https://HOSTNAME/users/github", "url": ""url": "https://HOSTNAME/users/github/starrid": "", "url": "https://HOSTNAME/users/github/following{/other_user}, "gists_url": "https://HOSTNAME/users/github/starred{/owner}, "gists_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "package_type": "container", "owner": { "login": "github", "id": 9919, "node_id": ""url": "https://HOSTNAME/users/github": ""url": ""url": "https://HOSTNAME/users/github/starred": "", "url": ""url": ""u
```

# Delete a package for an organization &

Deletes an entire package in an organization. You cannot delete a public package if any version of the package has more than 5,000 downloads. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must have admin permissions in the organization and authenticate using an access token with the read:packages and delete:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package you want to delete. For the list of these registries, see "About permissions for GitHub Packages."

## Parameters for "Delete a package for an organization"

#### **Headers**

```
accept string
Setting to application/vnd.github+json is recommended.
```

# Path parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

```
package_name string Required
```

The name of the package.

org string Required

#### HTTP response status codes for "Delete a package for an organization"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Delete a package for an organization"



#### Response

Status: 204

# Restore a package for an organization &

Restores an entire package in an organization.

You can restore a deleted package under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must have admin permissions in the organization and authenticate using an access token with the read:packages and write:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the <code>package\_type</code> belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package you want to restore. For the list of these registries, see "About permissions for <a href="GitHub Packages">GitHub Packages</a>."

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

# **Path parameters**

# package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

org string Required

The organization name. The name is not case sensitive.

#### **Query parameters**

token string

package token

# HTTP response status codes for "Restore a package for an organization"

# Status code Description No Content Requires authentication Forbidden Resource not found

# Code samples for "Restore a package for an organization"

POST /orgs/{org}/packages/{package\_type}/{package\_name}/restore

cURL JavaScript GitHub CLI

curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/orgs/ORG/packages/PACKAGE TYPE/PACKAGE NAME/restore

ſŪ

# Response

Status: 204

# List package versions for a package owned by an organization &

Lists package versions for a package owned by an organization.

If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

#### Parameters for "List package versions for a package owned by an organization"

#### Headers

accept string

Setting to application/vnd.github+json is recommended.

## Path parameters

# package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

org string Required

The organization name. The name is not case sensitive.

#### **Query parameters**

page integer

Page number of the results to fetch.

Default: 1

per\_page integer

The number of results per page (max 100).

Default: 30

**state** string

The state of the package, either active or deleted.

Default: active

Can be one of: active, deleted

# HTTP response status codes for "List package versions for a package owned by an organization"

Status code Description

200 OK

401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "List package versions for a package owned by an organization"



#### Response

```
Example response Response schema

Status: 200

[ { "id": 245301, "name": "1.0.4", "url": "https://HOSTNAME/orgs/octo-org/packages/npm/hello-world-npm/versions/245301", "package_html_url": "https://github.com/octo-org/hello-world-npm/packages/43752", "created_at": "2019-11-05T22:49:04Z", "updated_at": "2019-11-05T22:49:04Z", "html_url": "https://github.com/octo-org/hello-world-npm/packages/43752?version=1.0.4", "metadata": { "package_type": "npm" } }, { "id": 209672, "name": "1.0.3", "url": "https://HOSTNAME/orgs/octo-org/packages/npm/hello-world-npm/versions/209672", "package_html_url": "https://github.com/octo-org/hello-world-npm/packages/43752", "created_at": "2019-10-29T15:42:11Z", "updated_at": "2019-10-29T15:42:12Z", "html_url": "
```

# Get a package version for an organization &

Gets a specific package version in an organization.

You must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "Get a package version for an organization"

#### **Headers**

accept string
Setting to application/vnd.github+json is recommended.

#### **Path parameters**

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package name string Required

The name of the package.

org string Required

The organization name. The name is not case sensitive.

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Get a package version for an organization"

Status code	Description
200	OK

# Code samples for "Get a package version for an organization"



## Response

```
Example response Response schema

Status: 200

{ "id": 836, "name": "sha256:b3d3e366b55f9a54599220198b3db5da8f53592acbbb7dc7e4e9878762fc5344", "url": "https://HOSTNAME/orgs/github/packages/container/hello_docker/versions/836", "package_html_url": "https://github.com/orgs/github/packages/container/package/hello_docker", "created_at": "2020-05-19T22:19:11Z", "updated_at": "2020-05-19T22:19:11Z", "html_url": "https://github.com/orgs/github/packages/container/hello_docker/836", "metadata": { "package_type": "container", "container": { "tags": [ "latest" ] } }
```

# Delete package version for an organization &

Deletes a specific package version in an organization. If the package is public and the package version has more than 5,000 downloads, you cannot delete the package version. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must have admin permissions in the organization and authenticate using an access token with the read:packages and delete:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package whose version you want to delete. For the list of these registries, see "About permissions for GitHub Packages."

## Parameters for "Delete package version for an organization"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

org string Required

cURL

JavaScript

The organization name. The name is not case sensitive.

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Delete package version for an organization"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Delete package version for an organization"

GitHub CLI

DELETE /orgs/{org}/packages/{package\_type}/{package\_name}/versions/{package\_version\_id}

curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/orgs/ORG/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions/PACKAGE\_VERSION\_ID

ſĊ

#### Response

Status: 204

# Restore package version for an organization &

Restores a specific package version in an organization.

You can restore a deleted package under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must have admin permissions in the organization and authenticate using an access token with the read:packages and write:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package whose version you want to restore. For the list of these registries, see "About permissions for GitHub Packages."

# Parameters for "Restore package version for an organization"

# **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

org string Required

The organization name. The name is not case sensitive.

package\_version\_id integer Required

Unique identifier of the package version.

## HTTP response status codes for "Restore package version for an organization"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

## Code samples for "Restore package version for an organization"



#### Response

Status: 204

# Get list of conflicting packages during Docker migration for authenticated-user $\mathscr O$

Lists all packages that are owned by the authenticated user within the user's namespace, and that encountered a conflict during a Docker migration. To use this endpoint, you must authenticate using an access token with the read:packages scope.

# HTTP response status codes for "Get list of conflicting packages during Docker migration for authenticated-user"

Status code	Description
200	ОК

# Code samples for "Get list of conflicting packages during Docker migration for authenticated-user"



#### Response

```
Example response Response schema

Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "octocat", "id": 9919, "node_id": "MDEyOk9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.octocatusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "following_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "subscriptions_url": "https://HOSTNAME/users/github/orgs_url": "https://HOSTNAME/users/gi
```

# List packages for the authenticated user's namespace ∂

Lists packages owned by the authenticated user within the user's namespace.

To use this endpoint, you must authenticate using an access token with the read:packages scope. If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "List packages for the authenticated user's namespace"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

# Query parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

#### visibility string

The selected visibility of the packages. This parameter is optional and only filters an existing result set.

The internal visibility is only supported for GitHub Packages registries that allow for granular permissions. For other ecosystems internal is synonymous with private. For the list of GitHub Packages registries that support granular permissions, see "About permissions for GitHub Packages."

Can be one of: public, private, internal

```
page integer
```

Page number of the results to fetch.

Default: 1

```
per_page integer
```

The number of results per page (max 100).

Default: 30

## HTTP response status codes for "List packages for the authenticated user's namespace"

Status code	Description
200	ОК
400	The value of per_page multiplied by page cannot be greater than 10000.

## Code samples for "List packages for the authenticated user's namespace"



## Response

```
Example response Response schema

Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "octocat", "id": 9919, "node_id": "MDEy0k9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.octocatusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "following_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "package_type": "container", "owner": { "login": "octocat", "id": 9919, "node_id": "https://hostname/users/github": "", "url": "https://hostname/users/github/starrid": "", "url": "https://hostname/users/github/following{/other_user}, "gists_url": "https://hostname/users/github/starred{/owner}, "gists_url": "https://hostname/users/github/starred{/owner}, "gists_url": "https://hostname/users/github/starred{/owner}, "gists_url": "https://hostname/users/github/orgs", "gists_ur
```

# Get a package for the authenticated user $\mathscr P$

Gets a specific package for a package owned by the authenticated user.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "Get a package for the authenticated user"

#### **Headers**

accept string
Setting to application/vnd.github+json is recommended.

#### Path parameters

. - - - .

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

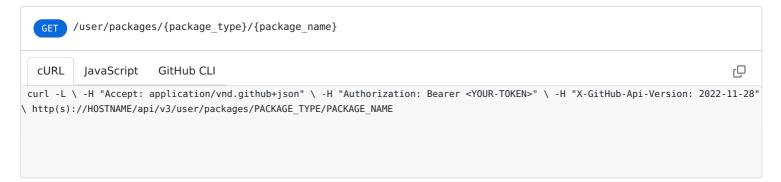
package name string Required

The name of the package.

## HTTP response status codes for "Get a package for the authenticated user"

Status code	Description
200	ОК

## Code samples for "Get a package for the authenticated user"



# Response

```
Example response Response schema

Status: 200

{ "id": 40201, "name": "octo-name", "package_type": "rubygems", "owner": { "login": "octocat", "id": 209477, "node_id": "MDQ6VXNlcjIw0TQ3Nw==", "avatar_url": "https://avatars.githubusercontent.com/u/209477?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url": "https://github.com/octocat", "followers_url": "https://HOSTNAME/users/octocat/following_/other_user}", "gists_url": "https://HOSTNAME/users/octocat/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/octocat/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/octocat/orgs", "subscriptions_url": "https://HOSTNAME/users/octocat/orgs", "
```

# Delete a package for the authenticated user @

Deletes a package owned by the authenticated user. You cannot delete a public package if any version of the package has more than 5,000 downloads. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> and <code>delete:packages</code> scopes. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

#### Parameters for "Delete a package for the authenticated user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### **Path parameters**

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

# HTTP response status codes for "Delete a package for the authenticated user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Delete a package for the authenticated user"



## Response

Status: 204

# Restore a package for the authenticated user &

Restores a package owned by the authenticated user.

You can restore a deleted package under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> and <code>write:packages</code> scopes. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "Restore a package for the authenticated user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

## Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

#### **Query parameters**

token string

package token

# HTTP response status codes for "Restore a package for the authenticated user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

## Code samples for "Restore a package for the authenticated user"



curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/user/packages/PACKAGE\_TYPE/PACKAGE\_NAME/restore

#### Response

Status: 204

# List package versions for a package owned by the authenticated user



Lists package versions for a package owned by the authenticated user.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "List package versions for a package owned by the authenticated user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

# Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

## **Query parameters**

page integer

Page number of the results to fetch.

Default: 1

per\_page integer

The number of results per page (max 100).

Default: 30

state string

The state of the package, either active or deleted

The state of the package, chile active of acteur

Default: active

Can be one of: active, deleted

# HTTP response status codes for "List package versions for a package owned by the authenticated user"

Status code	Description
200	ОК
401	Requires authentication
403	Forbidden
404	Resource not found

## Code samples for "List package versions for a package owned by the authenticated user"



#### Response

```
Example response Response schema

Status: 200

[ { "id": 45763, "name": "sha256:08a44bab0bddaddd8837a8b381aebc2e4b933768b981685a9e088360af0d3dd9", "url": "https://HOSTNAME/users/octocat/packages/container/hello_docker/versions/45763", "package_html_url": "https://github.com/users/octocat/packages/container/package/hello_docker", "created_at": "2020-09-11T21:56:40Z", "updated_at": "2021-02-05T21:32:32Z", "html_url": "https://github.com/users/octocat/packages/container/hello_docker/45763", "metadata": "package_type": "container", "container": { "tags": [ "latest" ] } } }, { "id": 881, "name": "sha256:b3d3e366b55f9a54599220198b3db5da8f53592acbbb7dc7e4e9878762fc5344", "url":
```

# Get a package version for the authenticated user &

Gets a specific package version for a package owned by the authenticated user.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "Get a package version for the authenticated user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Get a package version for the authenticated user"

Status code Description

200 OK

## Code samples for "Get a package version for the authenticated user"

CURL JavaScript GitHub CLI
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/user/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions/PACKAGE\_VERSION\_ID

#### Response

Example response Response schema

Status: 200

{ "id": 214, "name": "sha256:3561f0cff06caccddb99c93bd26e712fcc56a811de0f8ea7a17bb865f30b176a", "url":
"https://HOSTNAME/users/octocat/packages/container/hello\_docker/versions/214", "package\_html\_url":
"https://github.com/users/octocat/packages/container/package/hello\_docker", "created\_at": "2020-05-15T03:46:45Z", "updated\_at":
"2020-05-15T03:46:45Z", "html\_url": "https://github.com/users/octocat/packages/container/hello\_docker/214", "metadata": {
"package\_type": "container", "container": { "tags": [ "1.13.6" ] } }

# Delete a package version for the authenticated user &

Deletes a specific package version for a package owned by the authenticated user. If the package is public and the package version has more than 5,000 downloads, you cannot delete the package version. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must have admin permissions in the organization and authenticate using an access token with the <code>read:packages</code> and <code>delete:packages</code> scopes. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "Delete a package version for the authenticated user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### **Path parameters**

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Delete a package version for the authenticated user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

## Code samples for "Delete a package version for the authenticated user"

DELETE /user/packages/{package\_type}/{package\_name}/versions/{package\_version\_id}

CURL JavaScript GitHub CLI

curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/user/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions/PACKAGE\_VERSION\_ID

# Restore a package version for the authenticated user &

Restores a package version owned by the authenticated user.

You can restore a deleted package version under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> and <code>write:packages</code> scopes. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "Restore a package version for the authenticated user"

## Headers

accept string

Setting to application/vnd.github+json is recommended.

## **Path parameters**

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

404

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Restore a package version for the authenticated user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden

Resource not found

## Code samples for "Restore a package version for the authenticated user"



#### Response

Status: 204

# Get list of conflicting packages during Docker migration for user @

Lists all packages that are in a specific user's namespace, that the requesting user has access to, and that encountered a conflict during Docker migration. To use this endpoint, you must authenticate using an access token with the read:packages scope.

# Parameters for "Get list of conflicting packages during Docker migration for user"

## Headers

accept string

Setting to application/vnd.github+json is recommended.

# Path parameters

username string Required

The handle for the GitHub user account.

# HTTP response status codes for "Get list of conflicting packages during Docker migration for user"

Status code	Description
200	ОК
401	Requires authentication
403	Forbidden

# Code samples for "Get list of conflicting packages during Docker migration for user"



```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/users/USERNAME/docker/conflicts
```

#### Response

```
Example response Response schema

Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "octocat", "id": 9919, "node_id": "MDEy0k9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.octocatusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "following_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/github/or
```

# List packages for a user &

Lists all packages in a user's namespace for which the requesting user has access.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "List packages for a user"

## **Headers**

accept string
Setting to application/vnd.github+json is recommended.

## Path parameters

username string Required

The handle for the GitHub user account.

# Query parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

# visibility string

The selected visibility of the packages. This parameter is optional and only filters an existing result set.

The internal visibility is only supported for GitHub Packages registries that allow for granular permissions. For other ecosystems

internal is synonymous with private . For the list of GitHub Packages registries that support granular permissions, see "About

permissions for GitHub Packages."

Can be one of: public , private , internal

page integer

Page number of the results to fetch.

Default: 1

per page integer

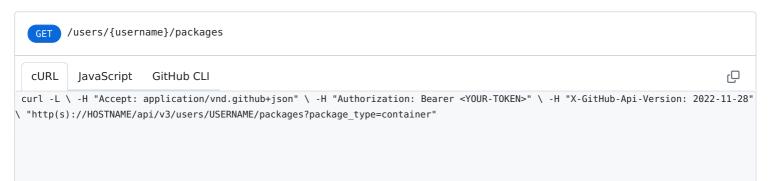
The number of results per page (max 100).

Default: 30

# HTTP response status codes for "List packages for a user"

Status code	Description
200	OK
400	The value of per_page multiplied by page cannot be greater than 10000.
401	Requires authentication
403	Forbidden

# Code samples for "List packages for a user"



#### Response

```
Example response Response schema

Status: 200

[ { "id": 197, "name": "hello_docker", "package_type": "container", "owner": { "login": "octocat", "id": 9919, "node_id": "MDEy0k9yZ2FuaXphdGlvbjk5MTk=", "avatar_url": "https://avatars.octocatusercontent.com/u/9919?v=4", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url": "https://github.com/github", "followers_url": "https://HOSTNAME/users/github/followers", "gists_url": "https://HOSTNAME/users/github/following{/other_user}", "gists_url": "https://HOSTNAME/users/github/starred{/owner}{/repo}", "subscriptions_url": "https://HOSTNAME/users/github/susers/github/orgs", "organizations_url": "https://HOSTNAME/users/github/orgs", "organizations_url": "https://HOSTNAME/users/gith
```

# Get a package for a user ₽

Gets a specific package metadata for a public package owned by a user.

To use this endpoint, you must authenticate using an access token with the read:packages scope. If the package\_type

belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

# Parameters for "Get a package for a user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

# HTTP response status codes for "Get a package for a user"

Status code Description

200 OK

# Code samples for "Get a package for a user"

GET /users/{username}/packages/{package\_type}/{package\_name}

CURL JavaScript GitHub CLI

curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/users/USERNAME/packages/PACKAGE\_TYPE/PACKAGE\_NAME

#### Response

```
Example response Response schema

Status: 200

{ "id": 40201, "name": "octo-name", "package_type": "rubygems", "owner": { "login": "octocat", "id": 209477, "node_id": "MDQ6VXNlcjIw0TQ3Nw==", "avatar_url": "https://avatars.githubusercontent.com/u/209477?v=4", "gravatar_id": "", "url": "https://H0STNAME/users/octocat", "html_url": "https://github.com/octocat", "followers_url": "https://H0STNAME/users/octocat/followers", "following_url": "https://H0STNAME/users/octocat/following{/other_user}", "gists_url": "https://H0STNAME/users/octocat/gists{/gist_id}", "starred_url": "https://H0STNAME/users/octocat/starred{/owner}{/repo}", "subscriptions_url": "https://H0STNAME/users/octocat/orgs", "organizations_url": "https://H0STN
```

# Delete a package for a user &

Deletes an entire package for a user. You cannot delete a public package if any version of the package has more than 5,000 downloads. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must authenticate using an access token with the read:packages and delete:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package you want to delete. For the list of these registries, see "About permissions for GitHub Packages."

## Parameters for "Delete a package for a user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

# package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type <code>maven</code>. Docker images pushed to GitHub's Container registry (<code>ghcr.io</code>) have the type <code>container</code>. You can use the type <code>docker</code> to find images that were pushed to GitHub's Docker registry (<code>docker.pkg.github.com</code>), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

# HTTP response status codes for "Delete a package for a user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Delete a package for a user"

DELETE	/users/{us	sername}/packa	ges/{package_type}/{package_name}	
cURL	JavaScript	GitHub CLI		C
curl -L \	\ -X DELETE \	-H "Accept: ap	plication/vnd.github+json" \ -H "Authorization: Bearer <your-token>" \ -H</your-token>	"X-GitHub-Api-Version:
2022-11-28	3" \ http(s):/	//HOSTNAME/api/	v3/users/USERNAME/packages/PACKAGE_TYPE/PACKAGE_NAME	

# Response

Status: 204

# Restore a package for a user &

Restores an entire package for a user.

You can restore a deleted package under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must authenticate using an access token with the read:packages and write:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package you want to restore. For the list of these registries, see "About permissions for GitHub Packages."

## Parameters for "Restore a package for a user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

## Path parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

Query parameters

token string

HTTP response status codes for "Restore a package for a user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

## Code samples for "Restore a package for a user"



#### Response

package token

Status: 204

# List package versions for a package owned by a user $\operatorname{\mathscr{O}}$

Lists package versions for a public package owned by a specified user.

To use this endpoint, you must authenticate using an access token with the <code>read:packages</code> scope. If the <code>package\_type</code> belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the <code>repo</code> scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "List package versions for a package owned by a user"

#### Headers

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

# HTTP response status codes for "List package versions for a package owned by a user"

Status code	Description
200	OK
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "List package versions for a package owned by a user"

GET /users/{username}/packages/{package\_type}/{package\_name}/versions

CURL JavaScript GitHub CLI

curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/users/USERNAME/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions

## Response

Example response Response schema

Status: 200

[ { "id": 3497268, "name": "0.3.0", "url": "https://HOSTNAME/users/octocat/packages/rubygems/octo-name/versions/3497268", "package\_html\_url": "https://github.com/octocat/octo-name-repo/packages/40201", "license": "MIT", "created\_at": "2020-08-31T15:22:11Z", "updated\_at": "2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "Project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems" } }, { "id": 2020-08-31T15:22:12Z", "description": "project for octocats", "html\_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "metadata": { "package\_type": "rubygems } }, { "id": 2020-08-31T15:22:12Z", "description": "project for octocats", "html.url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "html.url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.3.0", "html.url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0

รชาชรษ, "name": "ช.2.ช", "urt": "nttps://ทบราพลพะ/users/octocat/packages/rubygems/octo-name/versions/รชาชรษ", "package\_ntmt\_urt": "https://github.com/octocat/octo-name-repo/packages/40201", "license": "MIT", "created\_at": "2019-12-01T20:49:29Z", "updated\_at":

# Get a package version for a user &

Gets a specific package version for a public package owned by a specified user.

At this time, to use this endpoint, you must authenticate using an access token with the read:packages scope. If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of GitHub Packages registries that only support repository-scoped permissions, see "About permissions for GitHub Packages."

## Parameters for "Get a package version for a user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package name string Required

The name of the package.

package\_version\_id integer Required

Unique identifier of the package version.

username string Required

The handle for the GitHub user account.

# HTTP response status codes for "Get a package version for a user"

Status code Description

200 OK

## Code samples for "Get a package version for a user"

/users/{username}/packages/{package\_type}/{package\_name}/versions/{package\_version\_id}

cURL JavaScript GitHub CLI

curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/users/USERNAME/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions/PACKAGE\_VERSION\_ID

Q

#### Response

```
Example response

Response schema

Status: 200

{ "id": 387039, "name": "0.2.0", "url": "https://HOSTNAME/users/octocat/packages/rubygems/octo-name/versions/387039", "package_html_url": "https://github.com/octocat/octo-name-repo/packages/40201", "license": "MIT", "created_at": "2019-12-01T20:49:29Z", "updated_at": "2019-12-01T20:49:30Z", "description": "Octo-name client for Ruby", "html_url": "https://github.com/octocat/octo-name-repo/packages/40201?version=0.2.0", "metadata": { "package_type": "rubygems" } }
```

# Delete package version for a user @

Deletes a specific package version for a user. If the package is public and the package version has more than 5,000 downloads, you cannot delete the package version. In this scenario, contact GitHub support for further assistance.

To use this endpoint, you must authenticate using an access token with the read:packages and delete:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package whose version you want to delete. For the list of these registries, see "About permissions for GitHub Packages."

# Parameters for "Delete package version for a user"

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

# Path parameters

```
package_type string Required
```

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

package version id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Delete package version for a user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Delete package version for a user"



# Response

Status: 204

# Restore package version for a user $\mathscr {E}$

Restores a specific package version for a user.

You can restore a deleted package under the following conditions:

- The package was deleted within the last 30 days.
- The same package namespace and version is still available and not reused for a new package. If the same package namespace is not available, you will not be able to restore your package. In this scenario, to restore the deleted package, you must delete the new package that uses the deleted package's namespace first.

To use this endpoint, you must authenticate using an access token with the read:packages and write:packages scopes. In addition:

- If the package\_type belongs to a GitHub Packages registry that only supports repository-scoped permissions, your token must also include the repo scope. For the list of these registries, see "About permissions for GitHub Packages."
- If the package\_type belongs to a GitHub Packages registry that supports granular permissions, you must have admin permissions to the package whose version you want to restore. For the list of these registries, see "About permissions for GitHub Packages."

#### **Headers**

accept string

Setting to application/vnd.github+json is recommended.

#### Path parameters

package\_type string Required

The type of supported package. Packages in GitHub's Gradle registry have the type maven. Docker images pushed to GitHub's Container registry (ghcr.io) have the type container. You can use the type docker to find images that were pushed to GitHub's Docker registry (docker.pkg.github.com), even if these have now been migrated to the Container registry.

Can be one of: npm , maven , rubygems , docker , nuget , container

package\_name string Required

The name of the package.

username string Required

The handle for the GitHub user account.

package\_version\_id integer Required

Unique identifier of the package version.

# HTTP response status codes for "Restore package version for a user"

Status code	Description
204	No Content
401	Requires authentication
403	Forbidden
404	Resource not found

# Code samples for "Restore package version for a user"

post /users/{username}/packages/{package\_type}/{package\_name}/versions/{package\_version\_id}/restore

cURL JavaScript GitHub CLI

curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/users/USERNAME/packages/PACKAGE\_TYPE/PACKAGE\_NAME/versions/PACKAGE\_VERSION\_ID/restore

ſĠ

Response

Status: 204

# Legal

© 2023 GitHub, Inc. <u>Terms</u> <u>Privacy</u> <u>Status</u> <u>Pricing</u> <u>Expert services</u> <u>Blog</u>