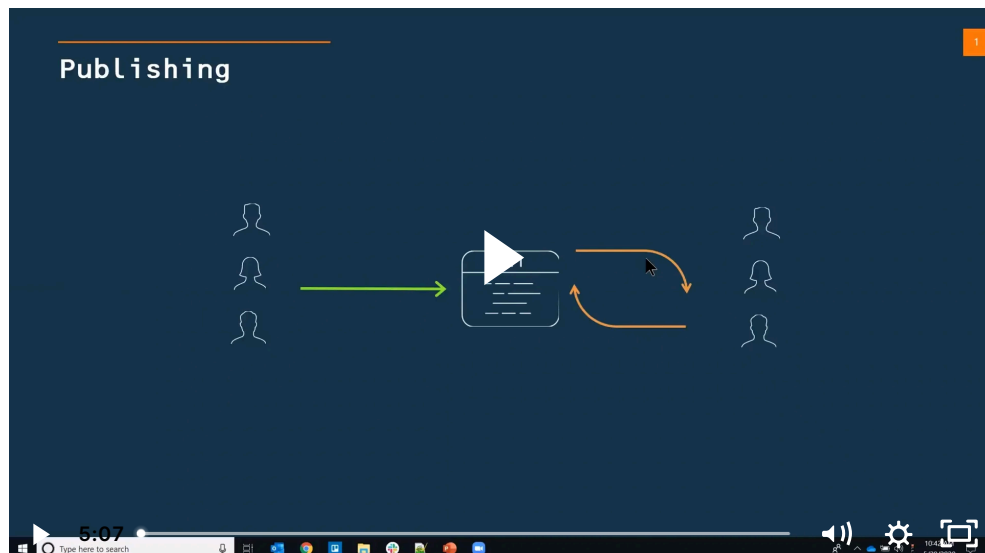# Publishing an API

**Watch the video**



You can publish your API definition when it is ready for production. Publishing is a way to show that the API is stable and that its endpoints can be reliably called from other applications.

Publishing makes the API definition read-only, so any changes you make after that point should be saved as a new version[versioning.html] of the API. Designers[../manage-resource-access/resource-roles.html] will not be able to edit the published version either.
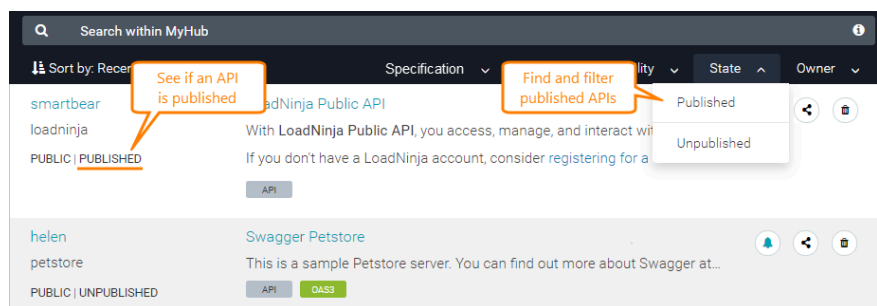
## Difference between published and unpublished APIs

The difference is mostly semantic:

- An *unpublished* API definition is a draft, a work in progress. Every API in SwaggerHub starts as unpublished. Though unpublished APIs are listed in the SwaggerHub registry, they are not ready for API consumers.

- A *published* API definition is a stable version ready for consumption from client applications. It is not supposed to be changed; any changes will be introduced as a new API version.

Technically, the difference is that published API versions cannot be edited (unless you unpublish them). If you want to change something, you should create a new API version and make changes.

SwaggerHub shows the **Published** label next to published APIs and domains. People searching SwaggerHub can also filter the list to show only published APIs.



Click the image to enlarge it.

## Published/unpublished vs public/private

Published and unpublished are not the same as public and private[public-and-private-apis.html]. Public and private indicate who can see the API in SwaggerHub -- everyone or just the selected collaborators. Published and unpublished indicate production readiness – whether the API is ready for consumption or not.

## Publish an API version

The API owner and designers can publish API versions.

1. Go to the API page in SwaggerHub.

2. If the API has several versions, select the version you want to publish.

**3**   Open the version list and click 🔊 **Publish Version**.

**4**   (Optional.) Select **Set as default version** to make this version the default version[versioning.html#set-the-default-version-1116279] of your API.

**5**   Click **Publish Version**.

Once published, the API version **becomes read-only** and cannot be edited. If you need to add new endpoints, change parameters, and so on, you should create a new version of your API and make the changes there. However, if you need to make some important changes to a published API, you can unpublish it[publishing-an-api.html#unpublish-an-api-version] temporarily.

> ✏️ Note
>
> All users sharing an API[../manage-resource-access/resource-roles.html] receive an email notification when an API is published or unpublished.

## Unpublish an API version

If you change your mind and want to pull the API definition back to make some changes, you can unpublish the API. To do this, you must be an **Owner** or **Designer**.

**1**   Open the API in the SwaggerHub Editor.

**2**   If the API has several versions, switch to the version you want to unpublish.

**3**   Open the version list and click 🔊 **Unpublish Version**.

**4**   In the dialog, click **Unpublish Version**.

This API version can now be edited again.

## Publish and unpublish via CLI

Use SwaggerHub CLI[https://github.com/SmartBear/swaggerhub-cli] to publish and unpublish your APIs from the command line.

To publish:

```
swaggerhub api:publish OWNER/API_NAME/VERSION
```

To unpublish:

```
swaggerhub api:unpublish OWNER/API_NAME/VERSION
```

`api:create` and `api:update` also have the `--published=publish|unpublish` option:

```
swaggerhub api:update OWNER/API_NAME/VERSION --file myapi.yaml --published=publish
```

## What's next

Once you have published your API, you can:

- Get the link to the API documentation and share it[share-apis/share-an-api.html] with your API consumers.

- Customize the branding[custom-branding-for-api-documentation.html] of your API documentation.

## See Also

- Creating a New API[creating-a-new-api.html]

- Public and Private APIs[public-and-private-apis.html]

- Sharing APIs[share-apis/share-an-api.html]

- Versioning[versioning.html]

- Custom Branding for API Documentation[custom-branding-for-api-documentation.html]

⊞