

From GitHub to Zenodo

Bibliothèque de l'EPFL

RDM Team go.epfl.ch/rdm

v. 2022

Pros, cons and integration between Github and Zenodo



VS.



Zenodo.org



| | | |
|-------------------|--|--|
| | Code and project hosting repository for version control and collaboration | General purpose open access repository developed under the European OpenAIRE program and operated by CERN (2013) |
| Status | Not open source (Microsoft, 2018) | Open source |
| Size limit | File limit of 100MB (command line) Otherwise Github LFS (stores references, not the file itself) | 50GB per dataset (no limit in dataset) Higher is possible |
| Licencing options | Templates existing allowing the addition of <i>LICENSE.md</i> file No restriction, but open are encouraged (choosealicense.com) | Any of 400 open licenses (from opendefinition.org) Restricted and closed content also supported |
| Cost | Free for individuals and organization, licensing fees for team and companies Fees for versioning control | Free up to 50GB per dataset Free versioning control |
| Longevity | Can block users to access account | 20 yrs + integration in case of shut down |
| Interoperability | Multiple | ORCID, DOI, GitHub |

Both are non-curated deposition databases

But they do not serve the same purpose



Code, data and project
hosting for collaboration

ACTIVE DATA



Zenodo.org

Code, data and project
repository for publication,
citation and reuse

PUBLISHED DATA

Both are non-curated deposition databases

But they do not serve the same purpose



Code, data and project
hosting for collaboration

ACTIVE DATA



- PID for objects, ISO standard to identify object uniquely
- Generally used for citation of journal articles, research reports, datasets, and official publications
- Persistent (data can be accessed in perpetuity)
- Unique
- Resolvable (metadata as description, findable localisation on the long term)

Both are non-curated deposition databases

But they do not serve the same purpose



Code, data and project hosting for collaboration

ACTIVE DATA



Zenodo.org

Code, data and project repository for publication, citation and reuse

PUBLISHED DATA



Zenodo makes open source code citable

Beware of involuntary (?) deletions



- 3 Under Danger Zone, click Delete this repository.

Danger Zone

Make this repository private
Public forks can't be made private. Please [duplicate the repository](#). Make private

Transfer ownership Transfer
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Delete this repository Delete this repository
Once you delete a repository, there is no going back. Please be certain.

- 4 Read the warnings.

- 5 To verify that you're deleting the correct repository, type the name of the repository you want to delete.

Create a beautiful site for your project with our GitHub Pages repository

Are you ABSOLUTELY sure?

Unexpected bad things will happen if you don't read this!

This action **CANNOT** be undone. This will delete the `jhosman/darkroom` repository, wiki, issues, and comments permanently.

Please type in the name of the repository to confirm.

I understand the consequences, delete this repository

General Policies v1.0

Removal

- **Revocation:** Content not considered to fall under the scope of the repository will be removed and associated DOIs issued by Zenodo revoked. Please signal promptly, ideally no later than 24 hours from upload, any suspected policy violation. Alternatively, content found to already have an external DOI will have the Zenodo DOI invalidated and the record updated to indicate the original external DOI. User access may be revoked on violation of Terms of Use.
- **Withdrawal:** If the uploaded research object must later be withdrawn, the reason for the withdrawal will be indicated on a tombstone page, which will henceforth be served in its place. **Withdrawal is considered an exceptional action**, which normally should be requested and fully justified by the original uploader. In any other circumstance reasonable attempts will be made to contact the original uploader to obtain consent. **The DOI and the URL of the original object are retained.**

Zenodo can publish your code in few easy steps



1. Clean and curate your repository in GitHub

Ex. add README, delete unnecessary files, add license...



<https://www.epfl.ch/campus/library/services-researchers/data-publication/zenodo/>



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)



Owner * Repository name *

 chiaragb / test 

Great repository names are short and memorable. Need inspiration? How about [bookish-enigma](#)?

Description (optional)

This is a test repository for the purpose of getting a DOI in Zenodo

-  **Public**
Anyone on the internet can see this repository. You choose who can commit.
-  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

- Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

This repository will be named **test** by default. You can change the default name in your [settings](#).

License ✕

Filter licenses...

- None
- Apache License 2.0
- GNU General Public License v3.0
- MIT License



1. Clean and curate your repository in GitHub
2. Activate the repository on Zenodo (can be tested with the [Sandbox](https://sandbox.zenodo.org/) <https://sandbox.zenodo.org/>)

The screenshot shows the Zenodo account settings page for GitHub integration. The browser address bar shows `sandbox.zenodo.org/account/settings/github/`. The page has a blue header with the Zenodo logo, a search bar, and navigation links for 'Upload' and 'Communities'. Below the header, there is a breadcrumb trail: 'Home / Account / GitHub'. On the left, a 'Settings' sidebar lists options: Profile, Change password, Security, Linked accounts, Applications, Shared links, and GitHub (which is selected). The main content area is titled 'GitHub Repositories' and includes a 'Sync now' button. It features a 'Get started' section with three steps: 1. Flip the switch (with an 'ON' toggle), 2. Create a release (with instructions to go to GitHub and create a release), and 3. Get the badge (with an example DOI badge: `DOI 10.5281/zenodo.8475`). Below this is a 'Repositories' section with a note about third-party access and a list of repositories, including 'chiaragb/test' with an 'ON' toggle.



1. Clean and curate your repository in GitHub
2. Activate the repository on Zenodo
3. Publish the release on GitHub

A screenshot of the GitHub web interface showing the "Releases" page for a repository named "chiaragb/test". The page is in a light gray theme. At the top, there's a navigation bar with "Pull requests", "Issues", "Marketplace", and "Explore". Below that, the repository name "chiaragb/test" is shown with "Public" status and icons for "Pin", "Unwatch" (1), "Fork" (0), and "Star" (0). A secondary navigation bar includes "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". The main content area has tabs for "Releases" (active) and "Tags". There are dropdowns for "Choose a tag" and "Target: main". A text input field contains "Published version of the data 2022.02.16". Below this is a rich text editor with "Write" and "Preview" tabs, and a "Preview" button. The editor contains the text "Description of this release, as precise as possible". There are also sections for "Attach files by dragging & dropping, selecting or pasting them." and "Attach binaries by dropping them here or selecting them.". At the bottom, there's a checkbox for "This is a pre-release" and two buttons: "Publish release" and "Save draft". On the right side, there are two informational sections: "Tagging suggestions" and "Semantic versioning". The footer of the page includes copyright information and various links like "Terms", "Privacy", "Security", "Status", "Docs", "Contact GitHub", "Pricing", "API", "Training", "Blog", and "About".



1. Clean and curate your repository in GitHub
2. Activate the repository on Zenodo
3. Publish the release on GitHub
4. Voilà! Now get the badge on Zenodo

zenodo Search Upload Communities

Home / Account / GitHub

Settings

- Profile
- Change password
- Security
- Linked accounts
- Applications
- Shared links
- GitHub**

GitHub Repositories (updated 56 seconds ago) Sync now ...

Get started

- 1 Flip the switch**
Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.
- 2 Create a release**
Go to GitHub and [create a release](#). Zenodo will automatically download a .zip-ball of each new release and register a DOI.
- 3 Get the badge**
After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.
DOI 10.5281/zenodo.8475
(example)

Enabled Repositories

DOI 10.5072/zenodo.1016148

Repositories

If your organization's repositories do not show up in the list, please ensure you have enabled third-party access to the Zenodo application. Private repositories are not supported.



1. Clean and curate your repository in GitHub
2. Activate the repository on Zenodo
3. Publish the release on GitHub
4. Voilà! Now get the badge on Zenodo...
5. ...and add the DOI to the README.md file in GitHub

The screenshot shows a GitHub repository page for 'chiaragb / test'. The repository is public and has a commit history table with the following entries:

| Commit Message | Commit Hash | Time Ago | Commits |
|---------------------------------|------------------------|----------------|-----------|
| chiaragb Added DOI to README.md | 0b96ff1 | 38 seconds ago | 3 commits |
| LICENSE | Initial commit | 26 minutes ago | |
| README.md | Added DOI to README.md | 38 seconds ago | |

The README.md file content is shown below the commit history. The DOI badge is highlighted in yellow:

```
DOI 10.5072/zenodo.1016148
```

The README content includes the title 'test' and the text: 'This is a test repository for the purpose of getting a DOI in Zenodo'.

The footer of the page shows: © 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



1. Clean and curate your repository in GitHub
2. Activate the repository on Zenodo
3. Publish the release on GitHub
4. Voilà! Now get the badge on Zenodo...
5. ...and add the DOI to the README.md file in GitHub
6. Now data is published in Zenodo!

The screenshot shows the Zenodo website interface. At the top is the Zenodo logo and navigation links: Search, Upload, and Communities. Below the navigation is a breadcrumb trail: Home / Account / GitHub / Repository. On the left is a 'Settings' sidebar with options: Profile, Change password, Security, Linked accounts, Applications, Shared links, and GitHub. The main content area displays the repository 'chiaragb/test' with a DOI badge '10.5072/zenodo.1016148' and a toggle switch set to 'ON'. Below this is a 'GitHub / Releases' section with a 'Create release ...' button. A release is listed: 'v1.0.0 chiaragb/test: Published version of the data 2022.02.16' with a 'Published' status and a timestamp of '12 minutes ago'. The DOI '10.5072/zenodo.1016148' is also displayed.

**Do not forget to check the
metadata in Zenodo...**
(you can modify them)

Available in

GitHub

Indexed in

OpenAIRE



Publication



Poster



Presentation



Dataset



Image



Video/Audio



Software



Lesson



Other

**||||| Digital Object Identifier**

Optional. Did your publisher already assign a DOI to your upload? If not, leave the field empty and we will register a new DOI for you. A DOI allows others to easily and unambiguously cite your upload. Please note that it is NOT possible to edit a Zenodo DOI once it has been registered by us, while it is always possible to edit a custom DOI.

📅 Publication date *

Required. Format: YYYY-MM-DD. In case your upload was already published elsewhere, please use the date of first publication.

📄 Title *

Required.

👤 Authors *

Optional.

[+ Add another author](#)**📝 Description *****B*****I******S*****x₂****x²*****I***_x

Source



**...and to add the Zenodo DOI to
the README file in the GitHub
repository**

README.md

| | |
|-------------------|--|
| Source Code DOI | DOI 10.5281/zenodo.456 |
| Data & Models DOI | DOI 10.5281/zenodo.458 |
| Documentation | docs passing |
| Latest Release | release v0.0.10 |
| License | license LGPL-3.0 |
| Build Status | build passing |
| Static Typing | mypy checked |
| Code Style | code style black imports isort docstrings pydocstyle PEP8 flake8 |

Available in

GitHub

Indexed in

OpenAIRE

Conclusions

They do not serve the same purpose



Code, data and project hosting for collaboration



Zenodo.org

Code, data and project repository for publication, citation and reuse

ACTIVE DATA



PUBLISHED DATA

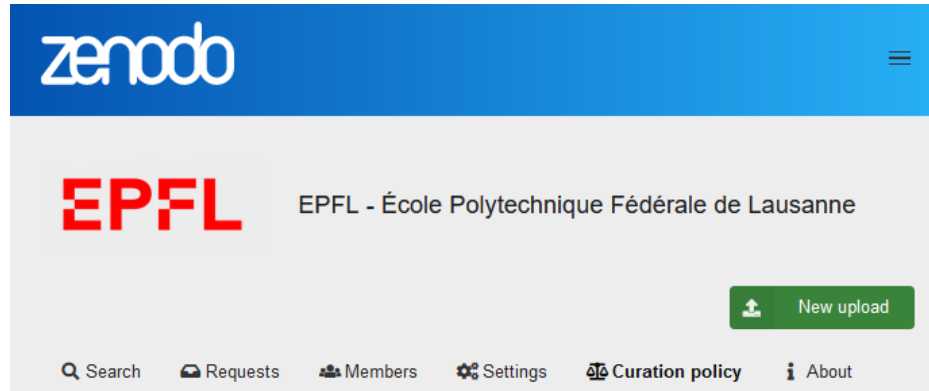


Zenodo makes open source code citable

The EPFL Community on Zenodo

Check out the information and tutorial at :

<https://www.epfl.ch/campus/library/services-researchers/data-publication/zenodo/>



The **MUST** criteria are required in order to accept a submission to the EPFL community.

MUST

1. At least one author is affiliated with EPFL at the time of the submission or creation of the submitted work.
2. The content of the upload must be accessible for review, i.e. Open Access, or Restricted after an access request has been completed. Embargoed uploads will be reviewed after the embargo has expired.
3. The Description of the submitted upload is sufficiently detailed. Mere references to external articles or other resources are not a sufficient description.
4. If no ORCID is listed, the name and surname and EPFL email address of at least one author are specified in the Description.

RECOMMENDED

1. Authors are identified by their ORCID.
2. The title is human-readable on the same level as conventional publications: filenames or coded expressions are deprecated.
3. If existing, references to related publications (e.g., article, source code, other datasets, etc.) are specified in the "Related/alternate identifiers" field, using a DOI if available.
4. In general, a README file is present in the root directory, and in case the submission consists of a compressed file then it is external. The README file is not needed for records consisting

<https://zenodo.org/communities/epfl/>

Best practices for preparing dataset for publication (not only in Zenodo...)

Curation by EPFL RDM team