
Tethys App Store

Release 1.0.0

Rohit Khattar

Mar 08, 2023

TABLE OF CONTENTS

1	Installation	3
1.1	Install using Miniconda (Recommended)	3
1.2	Install from GitHub	3
1.3	Updating Installed App Store	3
1.4	Migrating from Warehouse to App Store	4
2	Application Submission	5
2.1	Application Metadata + setup.py	5
2.2	Steps to Submit	6
3	Production Installation	9
4	Compatibility	11
5	GitHub Install API	13
5.1	/install/git/	13
5.2	/install/git/status?install_id=	13
5.3	/install/git/logs?install_id=	13
6	GitHub Actions	15
6.1	Github action example	15

The Tethys App Store is similar in concept to the iOS App Store for Apple mobile devices or the Google Play Store for Android mobile devices, but exclusively for Tethys Applications. The Tethys App Store aims to make web applications portable by packaging Tethys Applications as Python packages and hosting them on Miniconda repositories. The App Store includes an option for developers to contribute their applications by following a two-step work-flow within the App Store's user interface. The App Store's backend processes take care of all the heavy lifting to correctly prepare the code and make it available as an easily installable Miniconda package.

Warning: This documentation, the app store, and associated computing workflows and actions are under development.

INSTALLATION

This application can be installed on your local Tethys portal in the following ways:

1.1 Install using Miniconda (Recommended)

While using Miniconda install, we need to ensure that the Tethys portal is setup to allow for communication over websockets by setting up an in-memory channel layer:

```
# If you haven't set this already  
tethys settings --set CHANNEL_LAYERS.default.BACKEND channels.layers.InMemoryChannelLayer
```

Following that, installing the app store is a simple conda install command:

```
conda install -c tethysapp app_store
```

1.2 Install from GitHub

```
# Activate tethys environment if not already active  
conda activate tethys  
  
git clone https://github.com/BYU-Hydroinformatics/tethysapp-tethys_app_store.git  
cd tethysapp-tethys_app_store  
tethys install
```

1.3 Updating Installed App Store

If you installed the app store using the Miniconda command, then run the following command to update the app store to the latest version:

```
# Activate Tethys environment if not already active  
conda activate tethys  
  
conda install -c tethysapp app_store
```

In case you installed the app store from GitHub, just pull the latest changes:

```
cd <directory_where_app_store_is_installed>
git pull
```

1.4 Migrating from Warehouse to App Store

In September, 2021 this package went through a name change and all future updates are published as *app-store* and not *warehouse*. If you have an existing version of the *Tethys App Warehouse* installed on your system, please follow the following steps to update it to the *Tethys App Store*. These steps assume you had installed the warehouse using Miniconda.

For GitHub installs, please follow the standard uninstall and install new app procedures.

```
# Activate Tethys environment if not already active
conda activate tethys

tethys uninstall warehouse

conda remove -c tethysplatform --override-channels warehouse

conda install -c tethysapp app_store

# Restart your Tethys Instance (If Running in production)

sudo supervisorctl restart all
```


APPLICATION SUBMISSION

Before attempting to submit your application to the warehouse, ensure that your application fulfills the requirements for Tethys App Store:

- Application is compatible with Tethys 3+
- Application should run on Python 3.7+
- Cleanup old init files if the application was upgraded from an older Tethys 2 compatible version.: <https://gist.github.com/rfun/ca38bb487ca1649be8491227adb7ca37>

2.1 Application Metadata + setup.py

The build process uses the setup.py file to pull the metadata for your application. The following fields are pulled from the setup.py and are displayed in the app store:

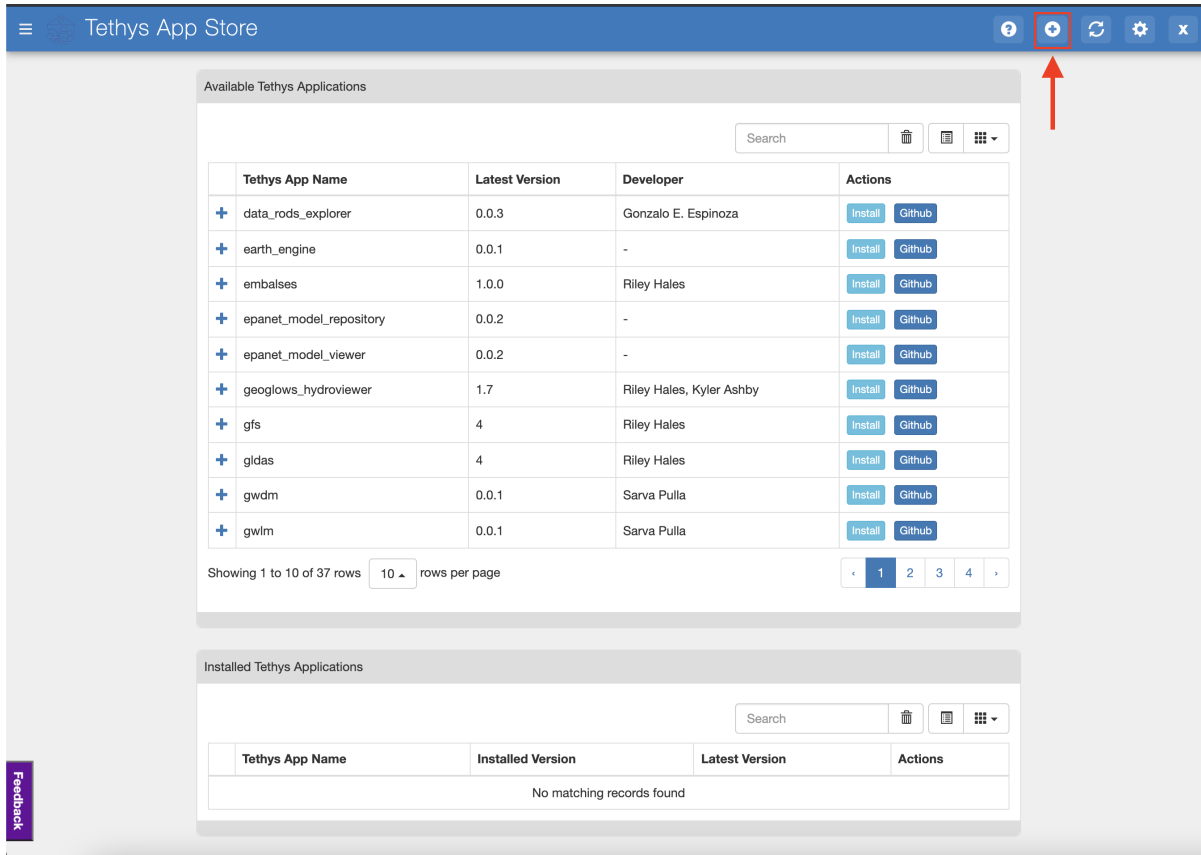
- Application name (Same as release package)
- Version
- Description
- Keywords
- Author Name
- Author Email
- URL
- License

It is recommended to fill in the values in your setup.py so that your application has those details visible in the app store for easier discovery and filtering.

Each time you have a new version for your application, it is recommended to update the version number in your setup.py file so that a new package is built and published.

2.2 Steps to Submit

Developers can submit their applications to the app store by click on the Add App button as highlighted in the image below:



Upon clicking that button, you will be presented with a modal that asks for the link to the GitHub Repository of your Tethys Application. It also instructs to put in an email address which is notified once the build is completed successfully and the application is available on the app store.

Submit your application to the Tethys App Store ✕

Click [Here](#) to get instructions on how to prepare your app for submission to the app store.

Please enter the GitHub URL of your application Repo and an email address which will be notified once the build is complete and your application is available on the Tethys App Store

Notification Email

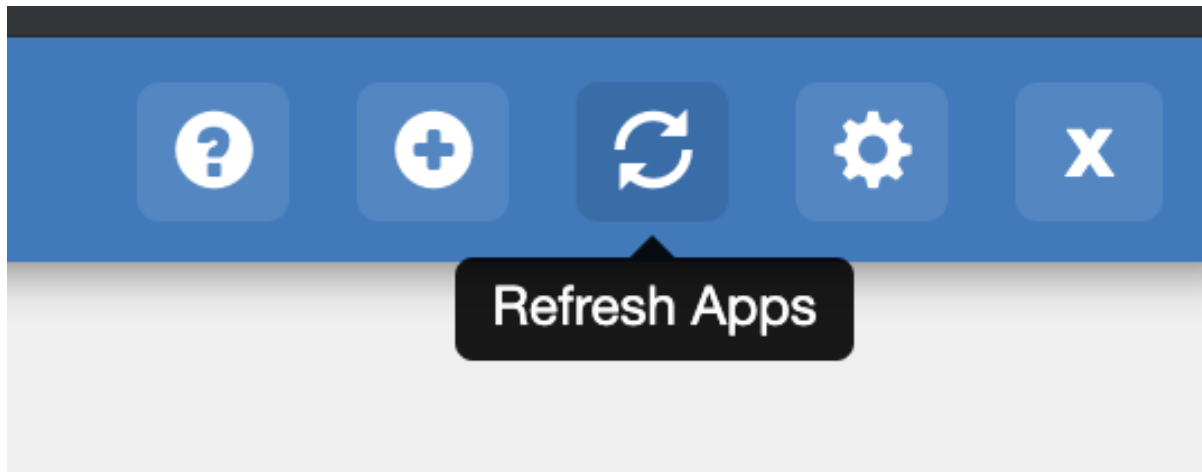
GitHub URL

[Fetch Repository](#)

[Cancel](#)

- Enter Notification email address (Mostly this will be your own or the developer's email address)
- Enter the link to your GitHub Repository
- If there are multiple branches on your GitHub repository, you will be presented with a list of branches on your GitHub repository, Select the branch that you would like to submit to the application app store. The app store uses the Master/Main branch in case only one branch exists.
- After selecting the branch the app store begins the processing. Once the build is done, an email will be sent to the address provided.

Note: After your application is successfully built, you will need to tell the app store to refresh your local list of applications. To do this click on the refresh button as shown in the image below.



PRODUCTION INSTALLATION

On a production server, ensure that you set the custom settings which require the SUDO password to the server user that has the ability to restart the Tethys Process. Usually this is the same as the user you used to setup Tethys. The password is stored in a Database and is only used when we need to restart the server after installing an application so that the changes can be seen.

Also, in case the Tethys portal is run within a Docker container, we need to ensure that the proxy setup on the host machine to forward Tethys requests to the Docker container supports Websockets as well. A working example is here:

```
# top-level http config for websocket headers
# If Upgrade is defined, Connection = upgrade
# If Upgrade is empty, Connection = close
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    listen 8008 default_server;
    listen [::]:8008 default_server ipv6only=on;

    location /tethys {
        rewrite ^/tethys/(.*)$ /$1 break;
        proxy_pass http://127.0.0.1:8006$uri$is_args$args;
        proxy_set_header Host $host;
        proxy_set_header X-Script-Name /tethys;
        proxy_cookie_path / /tethys;

        #WEBSOCKET Support
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }
}
```


COMPATIBILITY

- Python version 3.7 or higher: <https://www.python.org/downloads/>
- Miniconda (Conda) for Python version 3.7 or higher: <https://docs.conda.io/en/latest/miniconda.html>
- Tethys Platform version 3.0 or higher: <http://docs.tethysplatform.org/en/stable/installation.html>
- The software has been tested with Conda environments setup on Ubuntu 16+, MacOS X+, and Windows 10

GITHUB INSTALL API

The app store has added the following methods to support installation via GitHub Actions. The app repository will need to have public access for this installation method to work. This method is meant for apps that are being developed and tested in a specific portal. It facilitates direct app installation that can trigger on a push or release event from a GitHub repository. For final operational installations, we recommend publishing the app package to the Tethysapp conda channel using the Tethys App Store interface.

5.1 /install/git/

You will need to provide the following information as a POST request. The Branch parameter is optional and will pull the master/main branch if no branch is specified.

```
{
  url: "Your GitHub Repository URL",
  branch: "GitHub Branch to be installed",
  develop: True //Optional Parameter. By default the application is NOT installed in ↪develop mode.
}
```

You would also need to provide the following Authorization Header: Token <INSERT YOUR TOKEN HERE>. The token can be obtained for your user on the user management page (http://{{HOST_Portal}}/user/{{username}}/).

This endpoint returns a unique Installation id (install_id) that can be used in the other two requests below

5.2 /install/git/status?install_id=

GET endpoint to get the status report for a specific install_id

5.3 /install/git/logs?install_id=

GET endpoint to get the logs for a specific install_id

GITHUB ACTIONS

The basic workflow for installing *Tethys* applications directly from Github uses three main github actions. the following table includes links to the specific documentation for each of these actions

Table 1: Github actions used for installing Tethys applications from Github

Action	Description	Home
tethys-app-linter	A GitHub Action to check the code integrity of Tethys applications.	Link
flake8-check-action	A GitHub action to run Flake8 against your code	Link
http-request-action	A GitHub action to create HTTP Requests from GitHub Actions.	Link

6.1 Github action example

```
name: tethys-app-installation

on: [release]

jobs:
  app-lint:
    runs-on: ubuntu-latest
    name: Tethys App Lint
    steps:
      - name: Checkout
        uses: actions/checkout@v2
      - name: Run tethys-app-linter
        uses: tethysapp/tethys-app-linter@v1
  flake:
    runs-on: ubuntu-latest
    name: Flake8
    steps:
      - name: Checkout
        uses: actions/checkout@v2
      - name: Run flake
        uses: tonybajan/flake8-check-action@v1.0.0
    with:
      select: E,W,F # check for pep8 and pyflakes errors
      maxlinelength: 120
      reprotoken: ${ secrets.GITHUB_TOKEN }
```

(continues on next page)

(continued from previous page)

```
deployment:
  runs-on: ubuntu-latest
  name: Deployment
  needs: lint
  steps:
  - name: Get Variables
    id: vars
    run: echo ::set-output name=short_ref::${GITHUB_REF##*/}
  - name: Deploy Stage
    uses: fjogeleit/http-request-action@master
    with:
      method: 'POST'
      url: 'https://tethys-staging.byu.edu/apps/warehouse/install/git/'
      data: '{"url": "https://github.com/${{ github.repository }}.git", "branch": "${{↵
↵steps.vars.outputs.short_ref }}"}'
      customHeaders: '{"Authorization": "Token ${{ secrets.TETHYS_AUTH_TOKEN }}"}'
```

Note: An authorization token is needed to send a request to a specific Tethys portal. This token can be obtained from the API Key parameter in the user account home page of the Tethys portal. We recommend this token is added to the secrets of the github repo that is triggering this workflow. For more information on how to do this see [How to create secrets on Github](#)

Note: This Installation method is meant for apps that are being developed and continue to change in a regular basis. It makes it possible to trigger an app installation update based on a push or release GitHub event.
