NASAaccess

Release 3.3.1

Ibrahim Mohammed

Apr 20, 2023

CONTENTS:

Ibrahim N. Mohammed	

CHAPTER

ONE

ABOUT

NASAaccess is a software application in the form of a R package, a conda package and a Tethys web application. **NASAaccess** software can generate gridded ascii tables of climate CMIP5, CMIP6, and earth observation remote sensing data (GPM, TRMM, GLDAS) needed to drive various hydrological models (e.g., VIC, RHESSys, SWAT ...etc.). The **NASAaccess** Tethys web-based application can be used for accessing, reformatting, and visualizing climate and earth observation remote sensing gridded time series data as well.

1.1 NASAaccess R Guide

1.1.1 Prerequisites

On a local machine the user should have installed the following programs as well as setting up a user account. The list below gives a summary of what is needed to be done prior to work with NASAaccess software on any local machine:

- Installing R software
- Installing Rstudio software (Optional)
- NASAaccess R package needs a user registration access with Earthdata. Users should set up a registration account(s) with Earthdata login as well as well as authorizing NASA GES DISC data access. Please refer to https://disc.gsfc.nasa.gov/data-access for further details.
- Installing curl software . Since Mac users have curl as part of macOS build, Windows OS machines users should make sure that their local machines build have curl installed properly.
- Checking if you can run curl from your command prompt. Type *curl –help* and you should see the help pages for the curl program once everything is defined correctly.
- *After successful installation of NASAaccess software package as discussed in next section users should find that a reference file (.netrc) with* Earthdata *credentials stored in it to streamline the retrieval access from* NASA *servers has been created in user Home directory*.

Manual creation of the *.netrc* file

The *.netrc* file and the *_netrc* file (only for Windows OS machines) are generated automatically when installing the *NASAaccess* R software package. However, if the user wants to create theses access files manually here are the steps needed.

1. Define *HOME* variable in your Environment Variables by picking any directory you want to be referenced as your *HOME* directory. For convenient installation, the user should go with the machine default *HOME* directory. In many Winodws OS machines *HOME* directory is the user personal *Documents* folder (i.e., C:\Users\yourname\Documents).

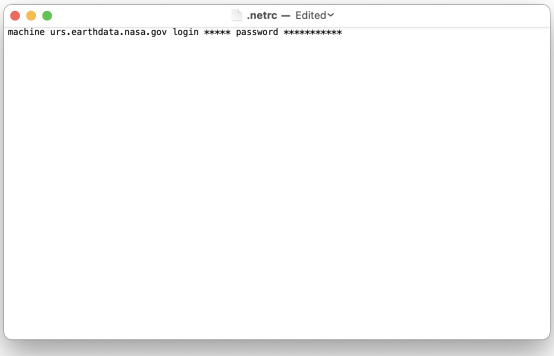
    1. Create .netrc file in your *Home* directory (*_netrc file creation is only needed for Windows OS machines*). Run these commands in your command prompt.

```
cd %HOME%
echo. > .netrc
echo "machine urs.earthdata.nasa.gov login <uid> password <password>" >>␣
↪ .netrc

echo. > _netrc
echo "machine urs.earthdata.nasa.gov login <uid> password <password>" >>␣
↪ _netrc

echo. > .urs_cookies
```

Note: Replace <uid> with your user name and <password> with your Earthdata Login password.

2. Open your .netrc and _netrc file(s) by any text editor and remove the quotations before machine and after your password. The *.netrc* and *_netrc* file(s) should be without any quotation marks to get the curl working. The contents of the *_netrc* and *.netrc* files should be identical.

3. The *.netrc* file at the user machine *Home* directory with the user NASA GES DISC logging information in it is depcited below for your reference. Accessing data at NASA servers is further explained at NASA earth data wiki. The *.netrc* file should look like:



Note: In your *.netrc* file <uid> is your user name and <password> is your Earthdata Login password.

4. For Windows OS machines user the NASA GES DISC logging information should be saved in a file *_netrc* identical to the *.netrc* file explained above.

Curl installation on Windows OS machines

Here are some instructions that might help in installing curl on Windows OS machines:

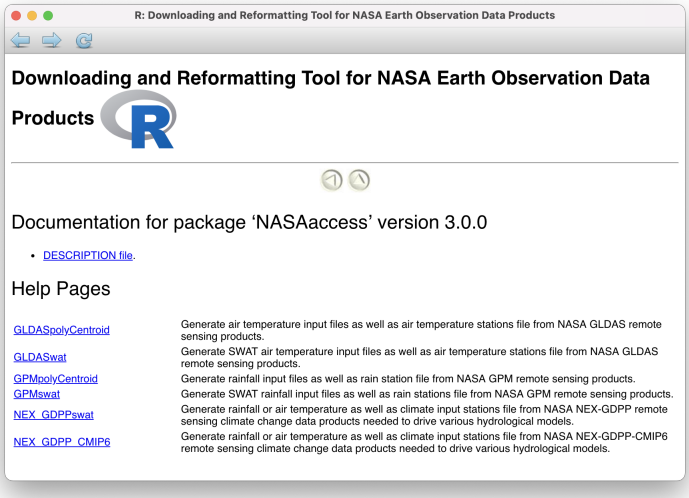
1. Download the 'curl' with the right built for your machine from https://curl.haxx.se/.
2. Unpack the zip file in a location at your discretion.
3. Add the curl.exe file location to your Environment Variables. Once you unpack the zip file you will find the curl.exe file in bin folder.
4. Close the Environment Variables and check if you can run curl from your command prompt. Type curl –help and you should see the help pages for the curl program once everything is defined correctly.

1.1.2 NASAaccess R Package Installation

Within Rstudio or R terminal run the following commands to install NASAaccess:

```
library(devtools)

install_github("nasa/NASAaccess", build_vignettes = TRUE)

library(NASAaccess)
```

Within the Rstudio help tab the user can verify that the package has been installed and browse the help pages of the various functions of NASAaccess. The help pages index should be similar to this



Note: NASAaccess R package version installed here is 3.0.0.

1.1.3 Getting Started with the NASAaccess R package

NASAaccess R package has multiple functions such as *GPMpolyCentroid*, *GPMswat*, and *NEX_GDPP_CMIP6* that download, extract, and reformat rainfall remote sensing and climate change data from NASA servers for grids within a specified watershed shapefile.

Let's explore *GPMpolyCentroid* function at an example watershed near Houston, TX.

```
library(ggmap)
library(raster)
library(ggplot2)
library(rgdal)

#Reading input data
dem_path <- system.file("extdata",
              "DEM_TX.tif",
              package = "NASAaccess")

shape_path <- system.file("extdata",
                "basin.shp",
                package = "NASAaccess")


 dem <- raster(dem_path)



 shape <- readOGR(shape_path)

 shape.df <- ggplot2::fortify(shape)

 #plot the watershed data
 myMap <- get_stamenmap(bbox = c(left = -96,
                      bottom = 29.7,
                      right = -95.2,
                      top = 30),
                                  maptype = "terrain",
                                  crop = TRUE,
                                   zoom = 10)

 ggmap(myMap) +
                 geom_polygon(data = shape.df,
                       aes(x = long, y = lat, group = group),
                       fill = NA, size = 0.5, color = 'red')
```

In order to use NASAaccess we also need a digital elevation model (DEM) raster layer. Let's see the White Oak Bayou watershed DEM and a more closer look at the study watershed example.

```
plot(dem,
      main="White Oak Bayou Watershed with Digital Elevation Model (DEM)",
          col=rev(bpy.colors()),
          xlab='lon',
          ylab='lat',
          legend  = T,
```

(continues on next page)

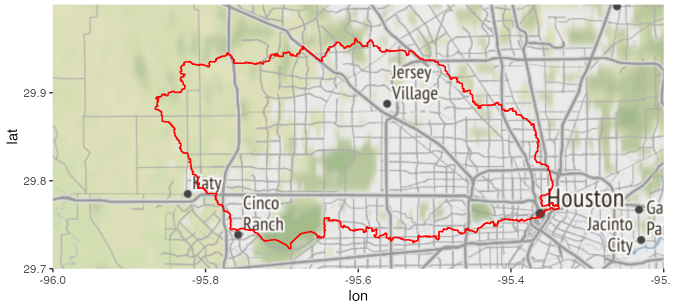


Fig. 1: The geographic layout of the White Oak Bayou watershed. Whiteoak Bayou is a tributary for the Buffalo Bayou River (Harris County, Texas).

(continued from previous page)

```
          legend.args=list(text='Elevation (m)',
             side=4,
             font=2,
             line=2.5,
             cex=0.8))

plot(shape , add = TRUE)
```

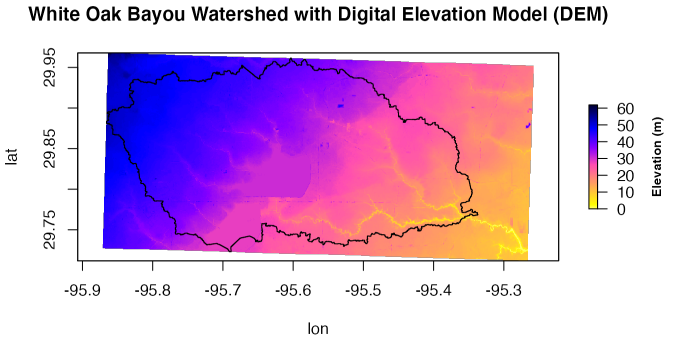


Fig. 2: The White Oak Bayou watershed with Digital elevation model in meters.

Now, let's examine *GPMpolyCentroid* function.

```
GPMpolyCentroid(Dir = "./GPMpolyCentroid/",
                watershed = shape_path,
                DEM = dem_path,
                start = "2019-08-1",
                end = "2019-08-3")
```

Examining the rainfall station file generated by *GPMpolyCentroid*

```
GPMpolyCentroid.precipitationMaster <- system.file('extdata/GPMpolyCentroid',
                                       'precipitationMaster.txt',
```

(continues on next page)

(continued from previous page)

```
                                        package = 'NASAaccess')

GPMpolyCentroid.precipitation.table <- read.csv(GPMpolyCentroid.
↪precipitationMaster)

#plotting
ggplot() +
            geom_polygon(data = shape.df,
                         aes(x = long, y = lat, group = group),
                         fill = NA,
                         colour = 'red') +
            geom_point(data=GPMpolyCentroid.precipitation.table,
                         aes(x=LONG,y=LAT))
```

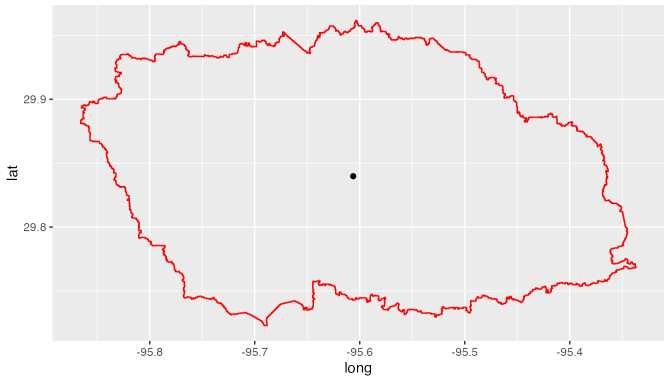


Fig. 3: The White Oak Bayou watershed with GPM remote sensing data.

We note here that GPMpolyCentroid has given us the GPM data grid that falls within a specified watershed and assigns a pseudo rainfall gauge located at the centroid of the watershed a weighted-average daily rainfall data as specified by the time period selected (i.e., 2019-08-01 to 2019-08-03).

Let's examine the rainfall data just obtained by *GPMpolyCentroid* over the White Oak Bayou study watershed during the time period selected.

```
GPMpolyCentroid.precipitation.record <- system.file('extdata/GPMpolyCentroid',
                                                    'precipitation1.txt',
                                                    package = 'NASAaccess')


GPMpolyCentroid.precipitation.data <- read.csv(GPMpolyCentroid.precipitation.
↪record)

#since data started on 2019-08-01

days <- seq.Date(from = as.Date('2019-08-01'),
                 length.out = dim(GPMpolyCentroid.precipitation.data)[1],
                 by = 'day')

#plotting the rainfall time series

```

(continues on next page)

(continued from previous page)

```
plot(days,
          GPMpolyCentroid.precipitation.data [,1],
          pch = 19,
          ylab= '(mm)',
          xlab = '',
          type = 'b',
          main = "White Oak Bayou Watershed precipitation (GPM)")
```

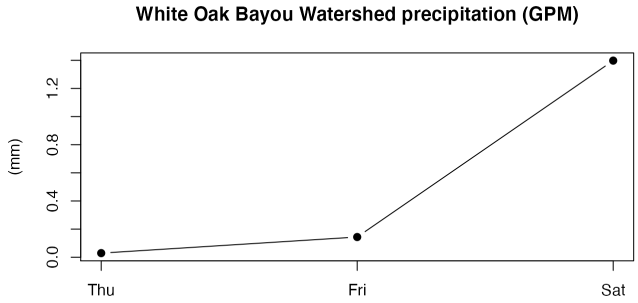


Fig. 4: GPM precipitation time series over the White Oak Bayou watershed during 1-3 August 2019.

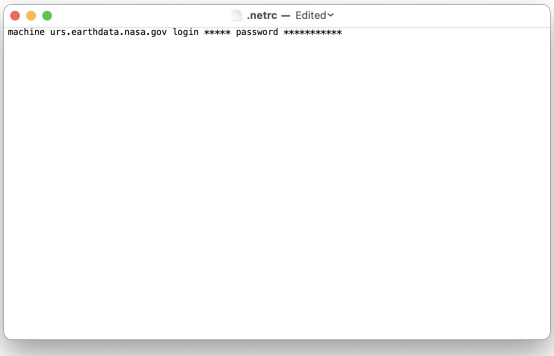
More examples on NASAaccess functionalities can be found Here.

1.2 NASAaccess Python Guide

1.2.1 Prerequisites

On a local machine the user should have installed the following programs as well as setting up a user account with Earthdata. The list below gives a summary of what is needed to be done prior to work with *NASAaccess* software on any local machine:

- Installing Anaconda or miniconda.
- *NASAaccess* software needs a user registration access with Earthdata. Users should set up a registration account(s) with Earthdata login as well as well as authorizing NASA GES DISC data access. Please refer to https://disc.gsfc.nasa.gov/data-access for further details.
- After registration with Earthdata *NASAaccess* software package users should create a reference file (*.netrc*) with Earthdata credentials stored in it to streamline the retrieval access to NASA servers.
  - Creating the *.netrc* file at the user machine *Home* directory and storing the user NASA GES DISC logging information in it is needed to execute the NASAaccess package commands. Accessing data at NASA servers



is further explained at NASA earth data wiki. The manual steps for creating the *.netrc* file has been discussed in NASAaccess R package installation. The *.netrc* file should look like:

Note: In your *.netrc* file <uid> is your user name and <password> is your Earthdata Login password.

– For Windows OS machines user the NASA GES DISC logging information should be saved in a file *_netrc* beside the *.netrc* file explained above. Define a *HOME* variable in your Environment Variables by picking any directory you want to be referenced as your *HOME* directory. In many machines *HOME* directory is already set to be your personal *Documents* folder (i.e., *C:\Users\yourname\Documents*). Store your netrc file(s) in your *Documents* or the specfied *HOME* directory.

1.2.2 NASAaccess Conda Package Installation

Installing the *r-nasaaccess* conda package is obtained by:

```
conda install -c conda-forge r-nasaaccess
```

1.2.3 Getting Started with the NASAaccess Conda package

The *NASAaccess* commands can be easily executed in the conda environment by writing the *NASAaccess* commands to a separate file (e.g., work.R) and running it by calling the Rscript executable in conda.

```
Rscript work.R
```

More examples on NASAaccess functionalities can be found Here.

1.3 NASAaccess Tethys web-based app Guide

1.3.1 About

The **NASAaccess** platform is available as software packages (i.e., R and conda packages) as well as an interactive format web-based environmental modeling application for earth observation data developed in the Tethys Platform framework (https://www.tethysplatform.org/). **NASAaccess** software can generate gridded ascii tables of climate CMIP5, CMIP6, and earth observation remote sensing data (GPM, TRMM, GLDAS) needed to drive various hydrological models (e.g., VIC, RHESSys, SWAT ...etc.). The **NASAaccess** has been envisioned to lower the technical barrier and simplify the process of accessing scalable distributed computing resources and leverage additional software for data and computationally intensive modeling frameworks. **NASAaccess** Tethys web-based application can be used for accessing, reformatting, and visualizing climate and earth observation remote sensing gridded time series data as well.

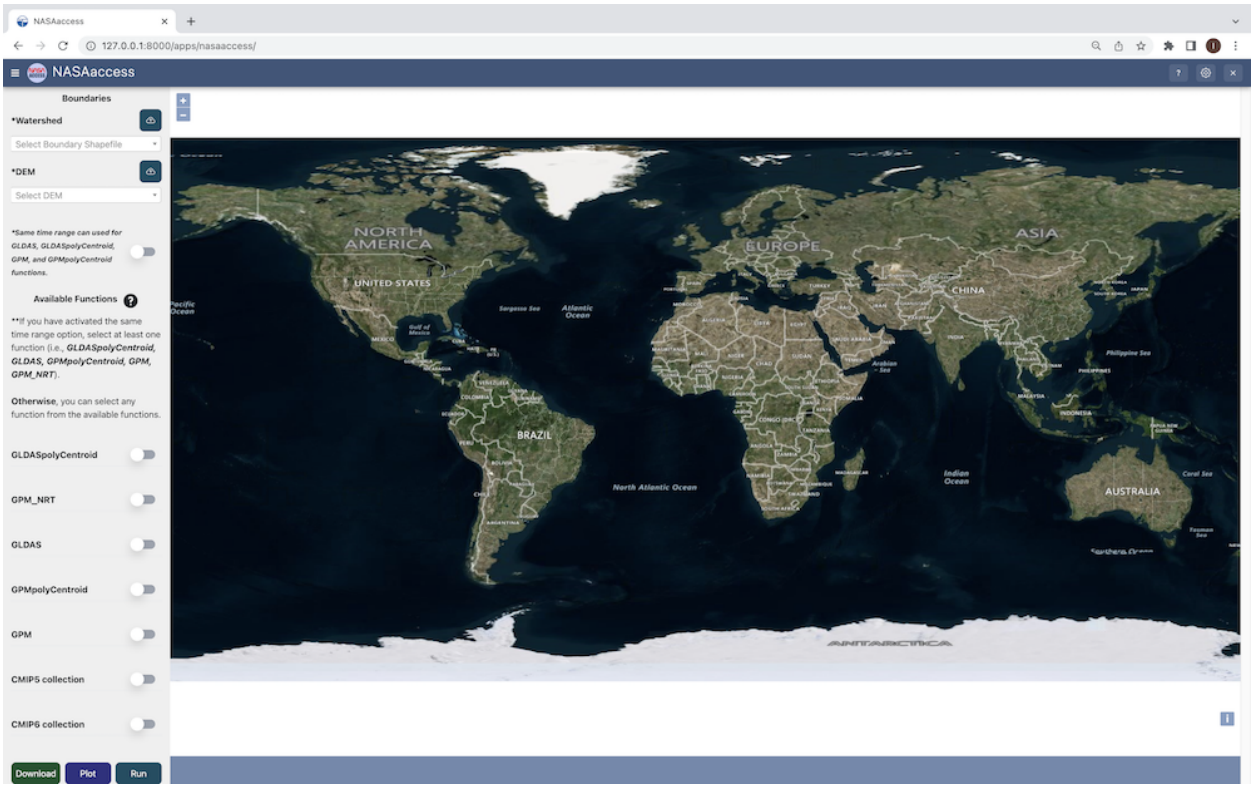


Fig. 5: **NASAaccess** Tethys web-based application home window

1.3.2 How it works

The **NASAaccess** Tethys Application is simply a user interface for passing arguments into the **NASAaccess** functions by calling the *r-nasaaccess* conda package (https://anaconda.org/conda-forge/r-nasaaccess). Using a combination of dropdowns, datepickers, and checkboxes, the app allows the user to select a watershed boundary, DEM, date range, and NASAaccess function(s) to pass to the server for running the selected **NASAaccess** function(s).

1.3.3 Requirements

- **Windows:**
  Setup a VirtualBox Machine via https://www.virtualbox.org/wiki/Downloads
- **Windows/MacOS/Linux:**
  Anaconda or miniconda

1.3.4 Installation/Setup

- **EarthData:**
  NASAaccess needs a user registration access with Earthdata. Users should set up a registration account(s) with Earthdata login as well as authorizing NASA GES DISC data access. Please refer to https://disc.gsfc.nasa.gov/data-access for further details.
  - After registration with Earthdata *NASAaccess* software package users should create a reference file (*.netrc*) with Earthdata credentials stored in it to streamline the retrieval access from NASA servers.
  - Creating the *.netrc* file at the user *Home* directory and storing the user NASA GES DISC logging information in it is needed to execute the *NASAaccess* package commands. Accessing data at NASA servers is further explained at https://wiki.earthdata.nasa.gov/display/EL/How+To+Access+Data+With+cURL+And+Wget. The manual steps for creating *.netrc* file has been discussed in NASAaccess R package installation.
- **Tethys:**

  The **NASAaccess** Tethys Application requires the Tethys Platform to be installed beforehand. The Tethys Platform Framework installation process can be installed in a development and production environment. There is a couple of differences between both installations:
  - The production installation uses a combination of the NGINX and Daphne servers.
  - Changes Are Not Automatically Loaded in the production server, but in the development server
  - Debug Disabled to prevent sensitive information from being leaked in the production server
  - Static Files Collected are collected to one location to be served more efficiently by NGINX.
  - Workspaces are collected to one location so they can be more easily backed up.
  - NGINX is given permission to access the static files and workspaces to be able to serve them.
  - **Development:**
    The installation of tethys in a development environment serves to contribute to the development of new applications and of the Tethys platform itself. The following are the required steps:

    1. Create a new conda environment and install the Tethys Platform by running the following command:

```
conda create -n tethys -c tethysplatform -c conda-forge tethys-
↪platform
```

2. Activate the Tethys conda Environment:

```
conda activate tethys
```

3. Generate a portal_config.yml file containing custom configurations such as the database and other local settings by running the following command:

```
tethys gen portal_config
```

4. Tethys Platform requires a PostgreSQL database server. There are several options for setting up a DB server: local, docker, or dedicated. The Tethys platform can also be used to create a local server that creates and migrates the tables associated with the Tethys platform framework by running:

   a. Local instance

   ```
   tethys db configure
   ```

   b. Docker local instance (requires docker installed beforehand)

   ```
   tethys docker init -c postgis

   tethys docker start -c postgis

   PGPASSWORD=<POSTGRES_PASSWORD> tethys db configure --username
   ↪<TETHYS_DB_USERNAME> --password <TETHYS_DB_PASSWORD> --
   ↪superuser-name <TETHYS_DB_SUPER_USERNAME> --superuser-
   ↪password <TETHYS_DB_SUPER_PASSWORD> --portal-superuser-name
   ↪<PORTAL_SUPERUSER_USERNAME> --portal-superuser-email '<PORTAL_
   ↪SUPERUSER_EMAIL>' --portal-superuser-pass <PORTAL_SUPERUSER_
   ↪PASSWORD>
   ```

5. Install *r-nasaaccess* in the tethsy environment:

```
conda install -c conda-forge r-nasaaccess
```

6. Initialize tables in persistent store databases:

```
tethys syncstores nasaaccess
```

7. Finally start the Tethys development server:

```
tethys manage start
```

– **Production:**

Installation in a production environment can be a manual installation (performing all of the production configuration steps manually) or a docker deployment. The following steps assumed the installation of Tethys in an Ubuntu production server (Note that before installing the Tethys platform, the following requirements needs to be installed).

* Requirements:

  · PostgreSQL

- NGINX
- Supervisor
- conda/mamba

* Installation steps:
  - Tethys Configuration:
  - **Install the Tethys platform via conda or mamba**

```
mamba create -n tethys -c tethysplatform -c conda-forge tethys-
↪platform
```

  - **Generate a portal_config.yml**

```
tethys gen portal_config
```

  - PostgreSQL Configuration:
  - **Set Database Settings in the portal_config.yml**

```
tethys settings --set DATABASES.default.NAME tethys_platform --
↪set DATABASES.default.USER <TETHYS_DB_USERNAME> --set DATABASES.
↪default.PASSWORD <TETHYS_DB_PASSWORD> --set DATABASES.default.
↪HOST <TETHYS_DB_HOST> --set DATABASES.default.PORT <TETHYS_DB_
↪PORT>
```

  - **Initialize, Create, and Migrate tables and users for the Database**

```
PGPASSWORD=<POSTGRES_PASSWORD> tethys db configure --username
↪<TETHYS_DB_USERNAME> --password <TETHYS_DB_PASSWORD> --
↪superuser-name <TETHYS_DB_SUPER_USERNAME> --superuser-password
↪<TETHYS_DB_SUPER_PASSWORD> --portal-superuser-name <PORTAL_
↪SUPERUSER_USERNAME> --portal-superuser-email '<PORTAL_SUPERUSER_
↪EMAIL>' --portal-superuser-pass <PORTAL_SUPERUSER_PASSWORD>
```

  - File Configuration:
  - Configuration Static and Workspace:
  - **Static files**

```
sudo mkdir -p <TETHYS_WORKSPACES_ROOT>
sudo chown -R $USER <TETHYS_WORKSPACES_ROOT>
tethys settings --set STATIC_ROOT /my/custom/static/directory
tethys manage collectstatic
```

  - **Workspaces**

```
sudo mkdir -p <TETHYS_WORKSPACES_ROOT>
sudo chown -R $USER <TETHYS_WORKSPACES_ROOT>
tethys settings --set TETHYS_WORKSPACES_ROOT /my/custom/static/
↪directory
tethys manage collectworkspaces
```

  - NGINX Configuration:
  - **Generate the NGINX configuration file using the tethys gen command**

```
tethys gen nginx --overwrite
```

- **Link the Tethys NGINX Configuration**

```
sudo ln -s <TETHYS_HOME>/tethys_nginx.conf /etc/nginx/sites-
↪enabled/tethys_nginx.conf
```

- **Remove the Default NGINX Configuration**

```
sudo rm /etc/nginx/sites-enabled/default
```

- **Get the name of the nginx user for use**

```
grep 'user .*;' /etc/nginx/nginx.conf | awk '{print $2}' | awk -
↪F';' '{print $1}'
```

- Supervisor Configuration:
- **Use the tethys gen command to generate default versions of these configuration files**

```
tethys gen nginx_service --overwrite
tethys gen asgi_service --overwrite
```

- If the process file is specified to be created at the root /run directory (e.g /run/tethys_asgi%(process_num)d.sock), then no action is required for this step.
- **Link the Tethys Supervisor Configuration Files**

```
sudo ln -s <TETHYS_HOME>/asgi_supervisord.conf /etc/supervisor/
↪conf.d/asgi_supervisord.conf
sudo ln -s <TETHYS_HOME>/nginx_supervisord.conf /etc/
↪supervisor/conf.d/nginx_supervisord.conf
```

- **Setup Tethys Log**

```
sudo mkdir -p /var/log/tethys
sudo touch /var/log/tethys/tethys.log
sudo chown -R <NGINX_USER> /var/log/tethys
```

- **Reload the Configuration**

```
sudo supervisorctl reread
sudo supervisorctl update
```

The steps for a manual and docker installation can be found in the Tethys platform documentation (http://docs.tethysplatform.org/en/stable/).

- GeoServer:

  Installation of GeoServer is necessary in order to use the **NASAaccess** Tethys web-based application. The GeosServer Software can be downloaded and installed on your local machine from (https://geoserver.org) or using the Tethys platform, which allows users to pull and run a GeoServer container. The following commands can be used to install GeoServer through the Tethys Platform, when prompted for settings value, press enter to keep the default values:

```
tethys docker init -c geoserver
tethys docker start -c geoserver
```

If GeoServer was installed from source, start GeoServer by changing into the directory `geoserver/bin` and executing the *startup.sh* script with the following commands:

```
cd geoserver/bin
sh startup.sh
```

Then, in a web browser, navigate to (http://localhost:8080/geoserver) to ensure that the GeoServer was installed successfully. Then, create a workspace with any name and upload a shapefile and associated digital elevation model (DEM) for your study area to your designated workspace. In the following screenshot we created a workspace named *nasaaccess* to illustrate publishing data to GeoServer. The details of the published data in GeoServer will be needed later in setting up the custom settings of the NASAaccess application. The screenshots shown below give the details needed in creating GeoServer workspace named *nasaaccess* and uploaded layers needed (i.e., shapefile and a digital elevation model - DEM) for the **NASAaccess** web-based application.

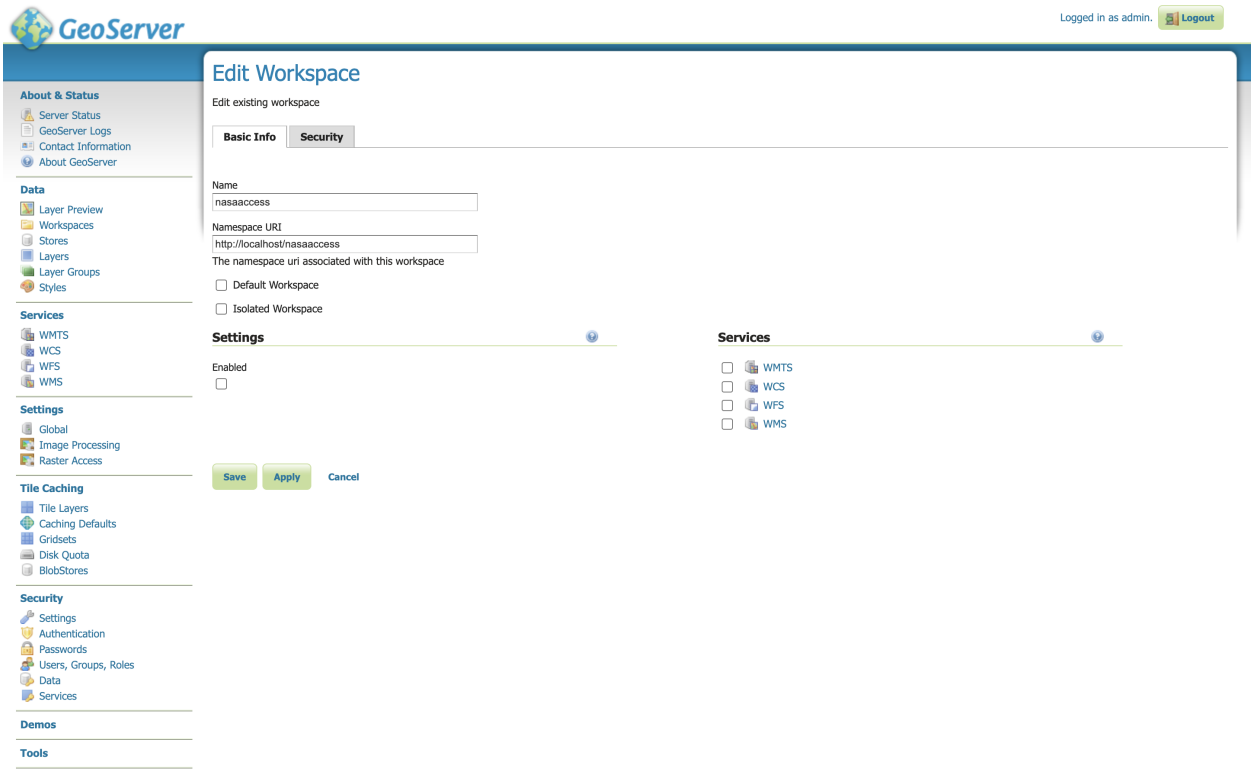


Fig. 6: GeoServer with a workspace name as *nasaaccess* and URI as (http://localhost/nasaaccess).

- **NASAaccess** Application Installation:

  After successful installation of the Tethys Platform and the GeoServer software on your work environment, clone the repository of the **NASAaccess** application available in Github. Next, install the application into the Tethys platform. Once the installation has started, the user will be prompted to select a spatial persistent service and the custom settings related to the application. Finally, start the Tethys development server after the installation has finished. The following commands and steps summarize the process of NASAaccess application installation:

  ```
  conda activate tethys
  ```

(continues on next page)

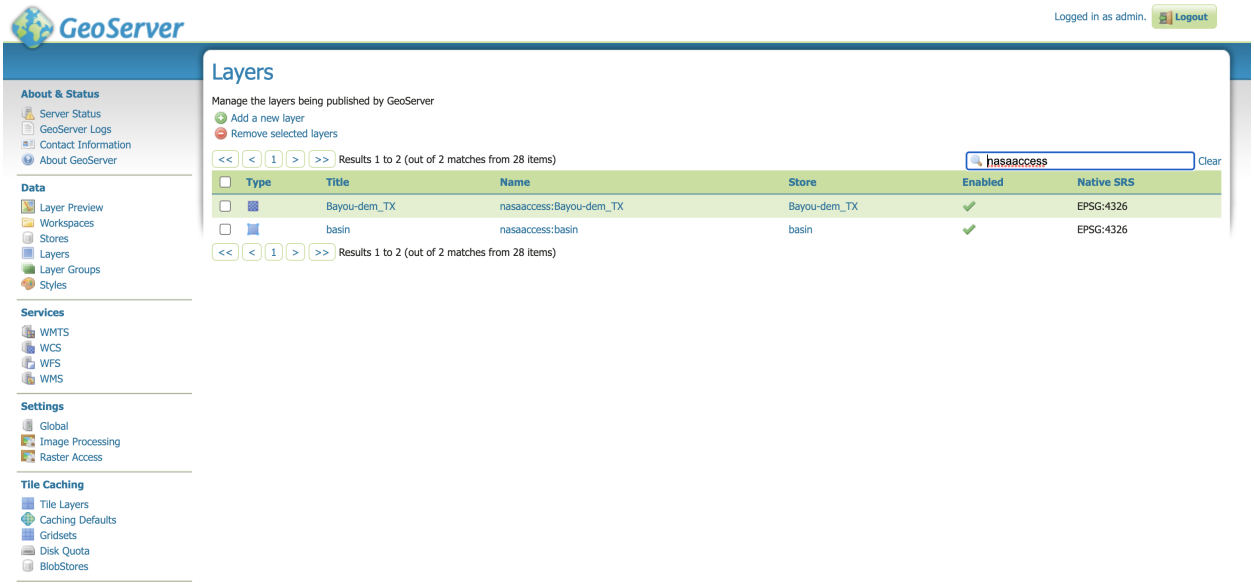


Fig. 7: GeoServer with published shapefile (i.e., basin) and a digital elevation model (i.e., Bayou-dem) stored in *nasaaccess* workspace.

(continued from previous page)

```
git clone https://github.com/imohamme/tethys_nasaaccess.git

cd tethys_nasaaccess
```

Note: make sure the libraries listed in requirements.txt are installed in your tethys environment (i.e., *r-nasaaccess*, *r-remotes*, *r-emayili*, and *r-codetools*)

```
tethys install -d
```

- Select the GeoSpatial persistent service (In this case, the installed GeoServer).
- Enter the value for the custom settings of the NASAaccess application:
  - *data path*: custom setting referring to the path of the data directory for download.
  - *nasaaccess_R*: custom setting referring to the *Rbin* path.
  - *nasaaccess_script*: custom setting referring to the nasaaccess R script containing the logic for data download using the r-nasaaccess conda package.
  - *geoserver workspace*: custom setting referring to the GeoServer workspace name associated with the NASAacces application.
  - *geoserver URI*: custom setting referring to the GeoServer workspace URI associated with the NASAacces application.
  - *geoserver user*: custom setting referring to the GeoServer admin user.
  - *geoserver password*: custom setting referring to the password related to the user of the geoserver user setting.

– Then, starting tethys:

```
tethys manage start
```

It is important to mention here that the custom settings of the **NASAaccess** application can be fixed after installing the application by passing the custom settings step with empty values. After running the Tethys application and navigating to the **NASAaccess** web-based application then these custom settings can be fixed. The following screenshot depicts the custom settings filled with needed information as discussed.

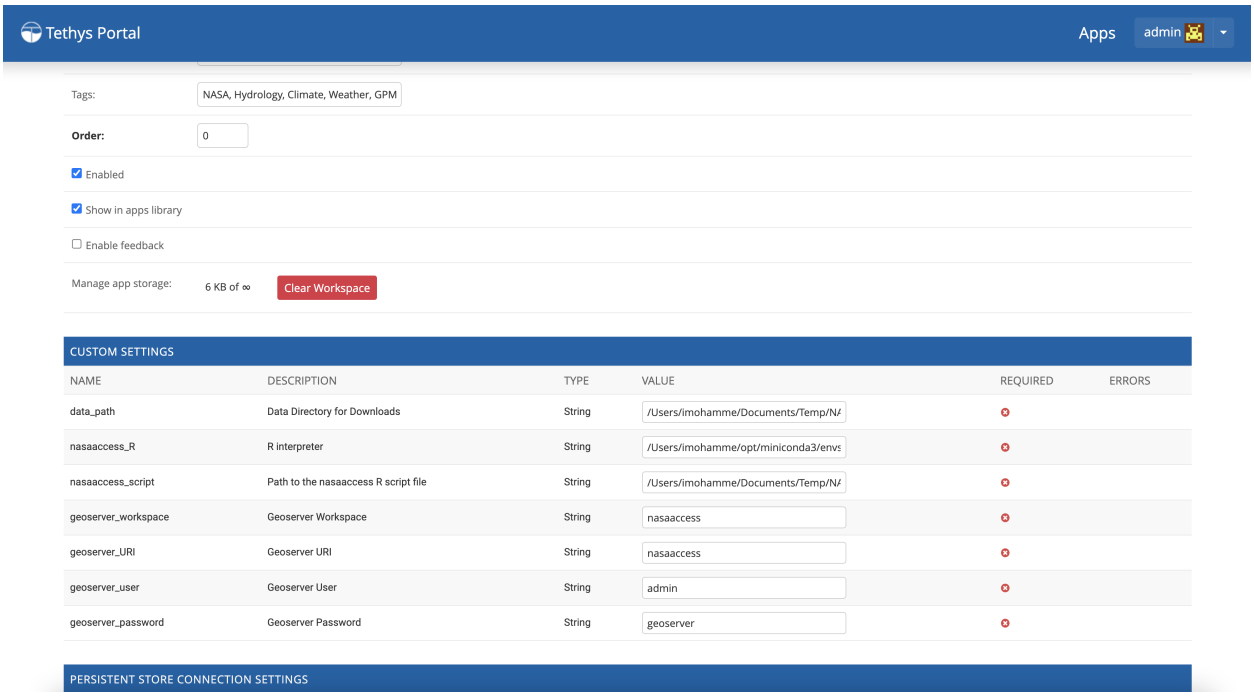


Fig. 8: **NASAaccess** custom settings configuration.
For the installation example shown the following customs settings are used: *data_path* (/pathto/tethys_nasaaccess/nasaaccess_data/), *nasaaccess_R* (/pathto/miniconda3/envs/tethys/bin/Rscript), *nasaaccess_script* (/pathto/tethys_nasaaccess/tethysapp/nasaaccess/scripts/nasaaccess.R), *geoserver_workspace* (nasaaccess), *geoserver_URI* (nasaaccess), *geoserver_user* (admin), and *geoserver_password* (geoserver).
After fixing the custom settings of the **NASAacces** web-based application, the Spatial dataset service needs to be configured manually as shown below. Note here the spatial dataset name is listed as *asaaccess* which is the GeoServer workspace configured previously. The username and password credentials need to match the GeoServer workspace configuration. In this case, the username is *admin* and password is *geoserver*.

After fixing all the needed settings of the NASAacces application, the user should be able to see the application active and ready to work.

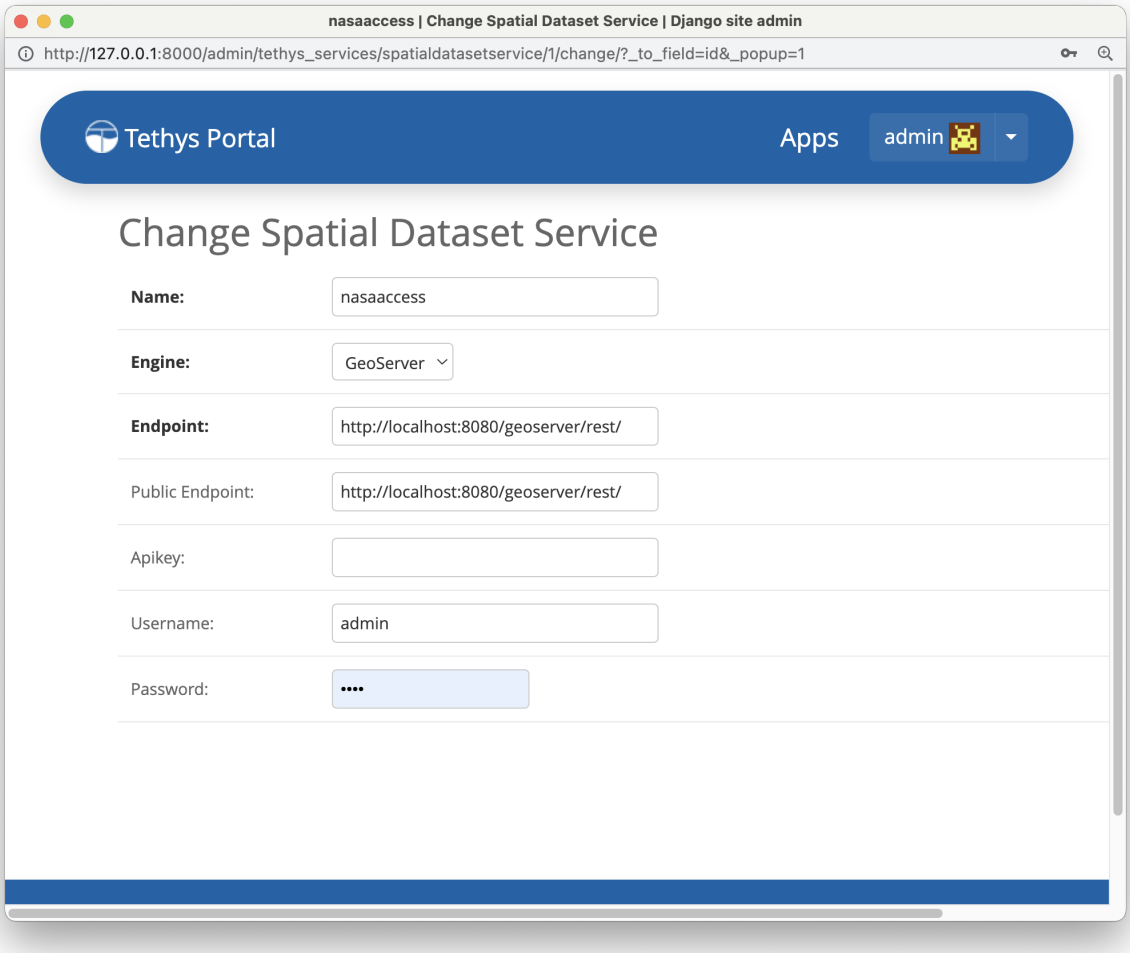


Fig. 9: **NASAaccess** Spatial Dataset Service settings configuration.

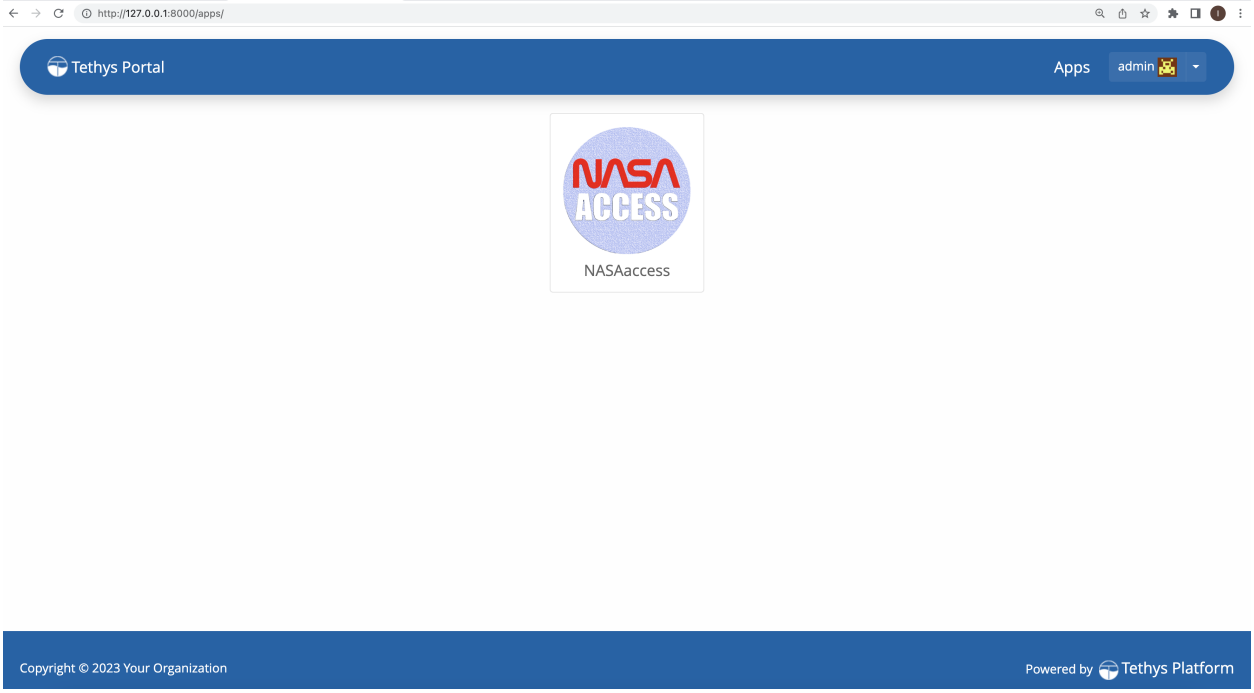


Fig. 10: NASAaccess web-based application after successful configuration.

1.3.5 Source Code & Documentation

The NASAaccess Tethys web-based application source code and documentation are available on Github:

- https://github.com/imohamme/tethys_nasaaccess

CHAPTER

TWO

LICENSE

The NASAaccess software package is an open source software package under NASA Open Source Agreement v1.3

CHAPTER

THREE

SOURCE CODE

The NASAaccess source code is available on Github:

- https://github.com/nasa/NASAaccess
- https://github.com/imohamme/tethys_nasaaccess

CHAPTER

FOUR

CITATION

Mohammed, I. N., 2019, NASAaccess: Downloading and reformatting tool for NASA earth observation data products software. National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, Maryland. https://github.com/nasa/NASAaccess