
Phoenix Documentation

Release 0.8

Birdhouse

Jan 25, 2018

Contents

1	Installation	3
1.1	Run Docker	4
2	Configuration	5
3	User Guide	7
3.1	Login	7
3.2	Dashboard	9
3.3	Processes	9
3.4	Monitor	11
3.5	Wizard	12
3.6	My Account	17
3.7	Settings (admins only)	18
4	Tutorial	25
4.1	Hello World	25
4.2	Run CDO sinfo on data from Thredds Dataservice	28
4.3	Run CDO ensemble operation on CMIP5 data from ESGF	34
4.4	Run CDO ensemble operation on CORDEX data from ESGF using OpenDAP	40
4.5	Creating a timeseries plot	44
4.6	Use the Birdhouse Solr Search in the Wizard	54
5	Troubleshooting	61
5.1	Phoenix does not start	61
5.2	Nginx does not start	62
6	Sphinx AutoAPI Index	63
6.1	ldap	63
6.2	__compat	63
6.3	db	63
6.4	events	63
6.5	layouts	64
6.6	catalog	64
6.7	security	65
6.8	patch	66
6.9	__init__	66
6.10	panels	66

6.11	exceptions	66
6.12	wps	66
6.13	utils	67
6.14	grid	68
6.15	twitcherclient	68
6.16	views	69
6.17	tasks	69
6.18	processes	70
6.19	providers	72
6.20	storage	73
6.21	services	73
6.22	wizard	75
6.23	cart	80
6.24	tests	82
6.25	supervisor	85
6.26	dashboard	85
6.27	settings	86
6.28	account	87
6.29	people	89
6.30	map	91
6.31	solrsearch	91
6.32	geoform	92
6.33	solr	93
6.34	esgf	94
6.35	monitor	97
7	Indices and tables	101
	Python Module Index	103

Phoenix (the bird) *Phoenix is a long-lived bird that is cyclically regenerated or reborn.* (Wikipedia). [..]

Pyramid Phoenix is a web-application build with the Python web-framework [pyramid](#). Phoenix has a user interface to make it easier to interact with [Web Processing Services](#). The user interface gives you the possibility to [register Web Processing Services](#). For these registered WPS services you can see which [Processes](#) they have available. You are provided with a form page to enter the parameters to [execute a process \(job\)](#). You can [monitor the jobs](#) and see the results.

In the climate science community many analyses are using climate data in the [NetCDF](#) format. Phoenix uses the [Malleefowl](#) WPS which provides processes to access NetCDF files from the [ESGF](#) data archive. Malleefowl provides a [workflow](#) process to chain ESGF data retrieval with another WPS process which needs NetCDF data as input. Phoenix has a [Wizard](#) to collect the parameters to run such a workflow with a process of a registered WPS.

Phoenix should help developers of WPS processes to use their processes more conveniently, especially for feeding their processes with different data sources (like ESGF data archive). Phoenix is also used for demonstration of available WPS processes.

Phoenix has a more generic and technical user interface. To use Phoenix successfully you need to have some knowledge about WPS and the existing data archives. So, Phoenix might not become a good choice for scientists users who just want to run a specific analyses job. There are other climate portals available which address these users. But Phoenix should at least become *developer friendly*.

Phoenix is easy to install using the [anaconda](#) python distribution and [buildout](#). So, Phoenix is not only available on production sites where it is close to data archives. You can also install it on your developer machine to make testing of your developed WPS processes easier and to present them to other people.

CHAPTER 1

Installation

This installation works on Linux 64-bit (Ubuntu 14.04, Centos 6, ...). It might still work on MacOSX but packages are updated only from time to time. Most of the dependencies come from [Anaconda](#) Python distribution system. Additional conda packages come from the [Binstar channel Birdhouse](#). The installation is done with [Buildout](#).

Phoenix uses WPS processes provided by Malleefowl. As a requisite you should install a local Malleefowl WPS (this will become part of the Phoenix installer). Alternatively you could configure the WPS URL of a running Malleefowl WPS instance in the Phoenix `custom.cfg`.

To install Malleefowl follow the instructions given in the [Malleefowl documentation](#). In short:

```
$ git clone https://github.com/bird-house/malleefowl.git
$ cd malleefowl
$ make clean install
```

Now start with downloading Phoenix with sources from github:

```
$ git clone https://github.com/bird-house/pyramid-phoenix.git
$ cd pyramid-phoenix
```

For install options run `make help` and read the documentation for the [Makefile](#).

Before installation you *need* to create a password for the local `phoenix` user which is used to login to the Phoenix web application:

```
$ make passwd
Generate Phoenix password ...
Enter a password with at least 8 characters.
Enter password:
Verify password:

Run 'make install restart' to activate this password.
```

Optionally take a look at `custom.cfg` and make additional changes. When you're finished, run `make clean install` to install Phoenix:

```
$ make clean install
```

You always have to rerun `make update` after making changes in `custom.cfg`.

After successful installation you need to start the services. All installed files (config etc ...) are below the conda environment `birdhouse` which is by default in your home directory `~/ .conda/envs/birdhouse`. Now, start the services:

```
$ make start      # starts supervisor services
$ make status     # shows status of supervisor services
```

Phoenix web application is available on `http://localhost:8081`.

Check the log file for errors:

```
$ tail -f ~/birdhouse/var/log/supervisor/phoenix.log
$ tail -f ~/birdhouse/var/log/supervisor/celery.log
```

1.1 Run Docker

Set the `HOSTNAME` environment variable (not `localhost`) and run `docker-compose`:

```
HOSTNAME=phoenix HTTP_PORT=8081 HTTPS_PORT=8443 SUPERVISOR_PORT=9001 docker-compose up
```


CHAPTER 2

Configuration

You can configure Phoenix by editing `custom.cfg` in the Phoenix source folder:

```
$ cd pyramid-phoenix
$ vim custom.cfg
$ cat custom.cfg
```

```
[settings]
hostname = localhost
http-port = 8081
https-port = 8443
log-level = INFO
# run 'make passwd' and to generate password hash
phoenix-password = sha256:#####
esgf-search-url = http://example.org/esg-search
wps-url = http://localhost:8091/wps
# register at github: https://github.com/settings/applications/new
github-consumer-key = #####
github-consumer-secret = #####
```

By default Phoenix runs on localhost. The HTTP port 8081 is redirected to the HTTPS port 8443. If you want to use a different hostname/port then edit the default values in `custom.cfg`:

```
[settings]
hostname = localhost
http-port = 8081
https-port = 8443
```

To be able to login with the phoenix admin user you need to create a password. For this run:

```
$ make passwd
```

To activate the GitHub login for external users you need to configure a GitHub application key for your Phoenix web application:

```
[settings]
# register at github:
github-consumer-key = #####
github-consumer-secret = #####
```

See the [GitHub Settings](#) on how to generate the application key for Phoenix.

If you want to use a different Malleefowl WPS service then change the `wps-url` value:

```
[settings]
wps-url = http://localhost:8091/wps
```

If you want to use a different ESGF index service then change the `esgf-search-url` value:

```
[settings]
esgf-search-url = http://example.org/esg-search
```

After any change to your `custom.cfg` you **need** to run `make update` again and restart the supervisor service:

```
$ make update    # or install
$ make restart
```

The user guide explains how to use the Phoenix web application to interact with Web Processing Services.

- *Login*
- *Dashboard*
- *Processes*
- *Monitor*
- *Wizard*
- *My Account*
- *Settings (admins only)*
 - *Register a WPS or Thredds service*
 - *Activate Users*
 - *Choose Authentication Protocol*
 - *GitHub Support*
 - *LDAP Support*
 - *Solr*


3.1 Login


Press the `Sign in` button in the upper right corner.

Sign In

The login page offers you several options to login to Phoenix.

Sign In

 Sign in with ESGF

 Sign in with GitHub

OR

Admin password *

If you have not configured your password yet then it is likely to be "qwerty"

Sign In

[Register for an account](#)

You can login using your ESGF OpenID or your GitHub account. If you login for the first time your account needs to be activated by an administrator.

If you are Phoenix admin you can also enter the admin password here.

ESGF OpenID

You can use an [ESGF OpenID](#). The ESGF OpenID is used later to access files from [ESGF](#). Make sure, that you have a valid ESGF OpenID of one of the ESGF Providers (for example [DKRZ](#)) and that you are able to download a datafile (you need to register for CMIP5 and CORDEX).

Enter the account name of your ESGF OpenID and choose the according ESGF OpenID provider (by default this is DKRZ).

Sign In

ESGF Provider *

☐ BADC ☒ DKRZ ☐ IPSL ☐ SMHI
☐ PCMDI

Select the Provider of your ESGF OpenID.

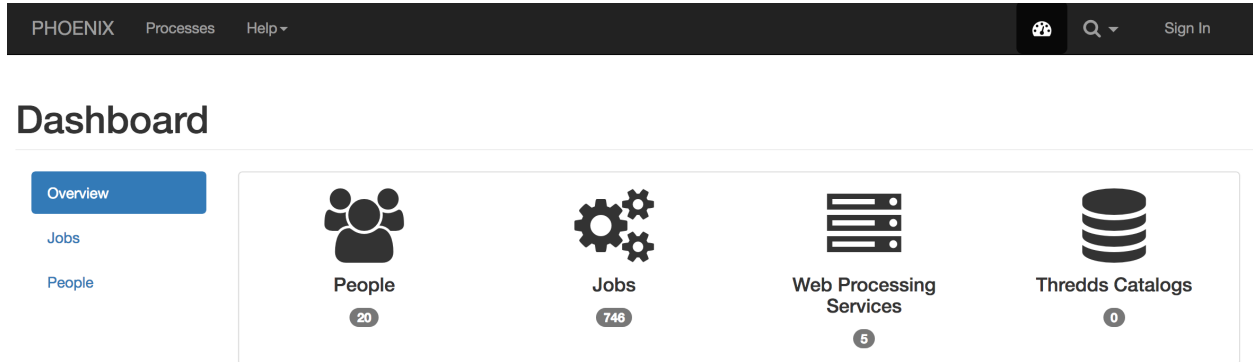
Username *

Your ESGF OpenID Username.

Sign In

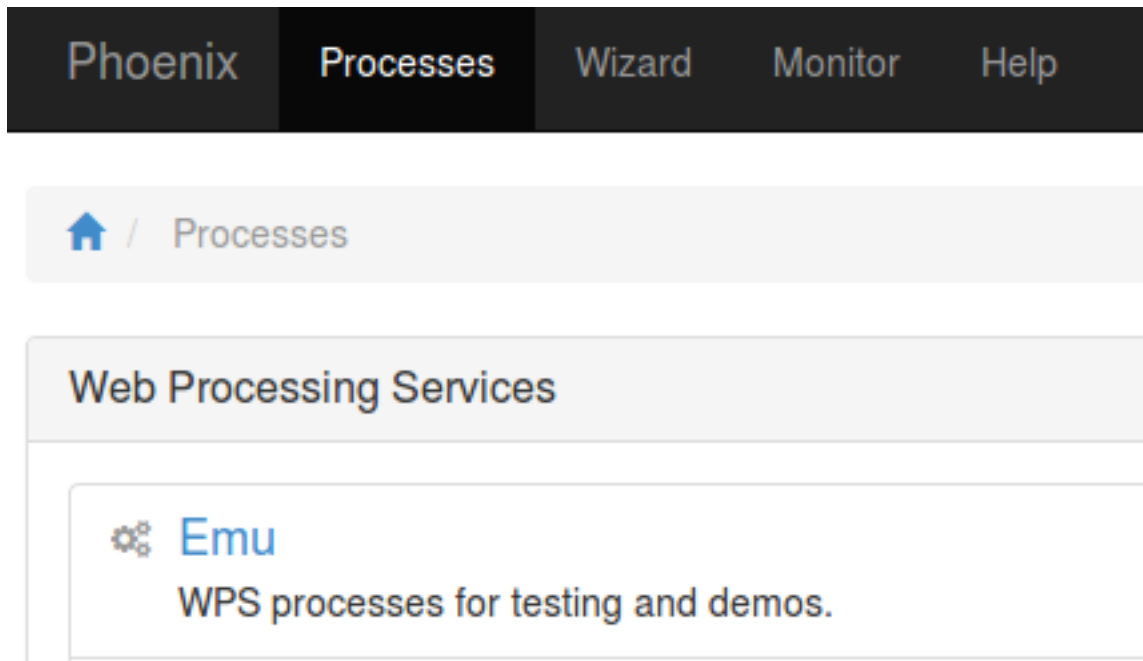
3.2 Dashboard

The dashboard shows some statistics about jobs and users.




3.3 Processes

When you have registered WPS services you can run a process. Go to the `Processes` tab.



Choose one of your registered WPS services. You will get a list of available processes (WPS `GetCapabilities` request).

Phoenix Processes Wizard Monitor Help


 / Processes / Emu


Description

WPS processes for testing and demos.

XML Provider: Birdhouse/Emu

Processes

 **Hello World 1.0**
Welcome user and say hello ...

 **Answer to Life, the Universe and Everything 2.0**
Numerical solution that is the answer to Life, Universe and Everything. T takes 7.5 milion years, but only a few seconds to give a response, with a

Choose one of these processes by using the `Execute` button. In case of Emu you may try the `Hello World` process. You will then be prompted to enter your username:

[Phoenix](#) [Processes](#) [Wizard](#) [Monitor](#) [Help](#)

[Home](#) / [Processes](#) / [Emu](#) / Hello World

Description

Welcome user and say hello ...

Inputs

Your name *

Please enter your name

Press the `Submit` button. When the process is submitted you will be shown your job list in `Monitor`.

3.4 Monitor

In `Monitor` all your running or finished jobs are listed. The list shows the status and progress of your jobs.

Job Monitor

This page shows the status of all your jobs.

My Jobs
Public
Private

All

Delete
Make Public
Make Private
Set Favorite
Unset Favorite

Process Status

Running 1
Finished 41
Matching 42
Sort C

<input type="checkbox"/>	Status	User	Process	Service	Caption	Finished	Duration	Labels	
<input type="checkbox"/>		Carsten Ehbrecht	cloud_taylor	copernicus	???	???	0:00:04	dev single async edit labels	Details Restart
<input type="checkbox"/>		Carsten Ehbrecht	cchecker	hummingbird	???	less than 1 minute ago	0:00:02	dev single async edit labels	Details Restart

When a job has finished with success you can see the results by clicking the `Details` button.

Job Details

This page shows the job details and polls the status of a running job.

☒ cchecker

100%

PyWPS Process IOOS Compliance Checker finished

Delete Job
Restart Job

☐ ???

dev single async

Ran for 0:00:02
3 minutes ago

Runs the IOOS Compliance Checker tool to check datasets against compliance standards. Each compliance standard is executed by a Check Suite, which functions similar to a Python standard Unit Test. A Check Suite runs one or more checks against a dataset, returning a list of Results which are then aggregated into a summary. Development and maintenance for the compliance checker is done by the Integrated Ocean Observing System (IOOS).

Job Log
Inputs
Outputs
View as XML

```

1 0:00:02 0%: PyWPS Process cchecker accepted
2 0:00:02 100%: PyWPS Process IOOS Compliance Checker finished

```

If the result has a document (XML, text, NetCDF, ...) you can view or download this document with the `Download` button.

3.5 Wizard

The wizard is used to chain WPS processes and to collect the input parameters for the processes. Currently the wizard chains a user WPS process with a WPS process to retrieve ESGF data. The chained processes are run with a [workflow management system](#) which is available as WPS process in [Malleefowl](#).

Go to the `Wizard` tab. Enter the appropriate parameters and use `Next` to get to the next wizard page.

Phoenix Dashboard Processes My Jobs **Wizard** My Account

[Home](#) / [Wizard](#) / Start

Start

Choose Favorite or None.

Favorite*

You need to choose a WPS service (e.a. Malleefowl).

[Home](#) / [Wizard](#) / WPS


WPS

Choose Web Processing Service

WPS service* ☒ Malleefowl (Malleefowl Processes (esgf, workflow, publish, security, ...)) [http://localhost:8091/wps]
☐ Hummingbird (WPS processes for general tools used in the climate science community like cdo) [http://localhost:8092/wps]
☐ Flyingpigeon (Processes for climate data, indices and extrem events) [http://localhost:8093/wps]
☐ Emu (WPS processes for testing and demos) [http://localhost:8094/wps]
☐ C3 WPS (C3 WPS processes for testing and demos) [http://localhost:8095/wps]

Select WPS

Choose a process (in case of Malleefowl only Dummy).

 / [Wizard](#) / Choose WPS Process

Choose WPS Process Malleefowl

WPS Process*


- ☐ Logon with ESGF OpenID [esgf_logon]
- ☐ ESGF Search [esgsearch]
- ☐ Download files [wget]
- ☐ Run Dispel Workflow [dispel]
- ☒ Dummy [dummy]
- ☐ Publish [publish]

Previous


Next

Cancel

Select the input parameter of the chosen process (mime-type application/netcdf).

 / [Wizard](#) / Choose Complex Input Parameter

Choose Complex Input Parameter Process Dummy

Input Parameter*  Resource [application/x-netcdf] (0-100)

Previous

Next

Cancel

Select the input source (ESGF).

[Home](#) / [Wizard](#) / Choose Source

Choose Source resource

Source*  ESGF Files

Previous

Next

Cancel

Select an ESGF dataset (select categorie (blue) and values of this category (orange), current selection (green)).

[Home](#) / [Wizard](#) / ESGF Search

ESGF Search*

Datasets found: 4

Options: ☐ All Sites ☐ Including Replicas ☒ Latest Version ☒ Temporal

Query

.

Current Selection

project:CORDEX ✕

time_frequency:mon ✕

institute:MPI-CSC ✕

variable:tasmax ✕

domain:WAS-44 ✕

Search Categories

experiment

experiment_family

Category: domain

WAS-44

Start 2001-01-01T12:00:00Z

End 2010-12-31T12:00:00Z


Previous

Next

Cancel

Please select **only one Dataset!**

You will be prompted for your password of your OpenID if your certificate is not valid anymore.

 / [Wizard](#) / ESGF Credentials

OpenID

OpenID from your ESGF provider

Password

Password for this OpenID

Previous

Next

Cancel

On the final page you can enter some keywords for your process and mark it as favorite (when using a favorite you don't need to enter all parameters again). Press **Done** and the job will be started and shown in your job list **My Jobs**.

[Home](#) / [Wizard](#) / Done

Done

Describe your Job and start Workflow.

Keywords*


test

workflow

dummy

Tags

Save as Favorite



Favorite Name*

dummy





Previous

Done

Cancel

3.6 My Account

In `My Account` you can change your user settings (user name, organisation, openid, ...).

Phoenix Processes Wizard Monitor Help MacPingu    

My Account

Twitcher access token

ESGF access token

Swift access token

Account settings

Your Name

MacPingu

EEmail

None

Organisation

Notes

Update

You can also see your current `Twitcher` access token which you can use to access a registered WPS service directly.

The screenshot shows the 'My Account' page in the Phoenix application. On the left, there is a sidebar with links: 'My Account', 'Twitcher access token' (highlighted in blue), 'ESGF access token', and 'Swift access token'. The main content area displays the 'Twitcher access token' section. It includes a 'Generate token' button in the top right corner. Below this, the 'Twitcher access token' is shown as a long alphanumeric string: '93a454758f65433e8c015e4526ffb7cb'. Underneath the token, the 'Expires' date and time are listed: '2016-04-22 21:27:21 UTC'.

See the [Twitcher Tutorial](#) on how to use the token to access a WPS service.

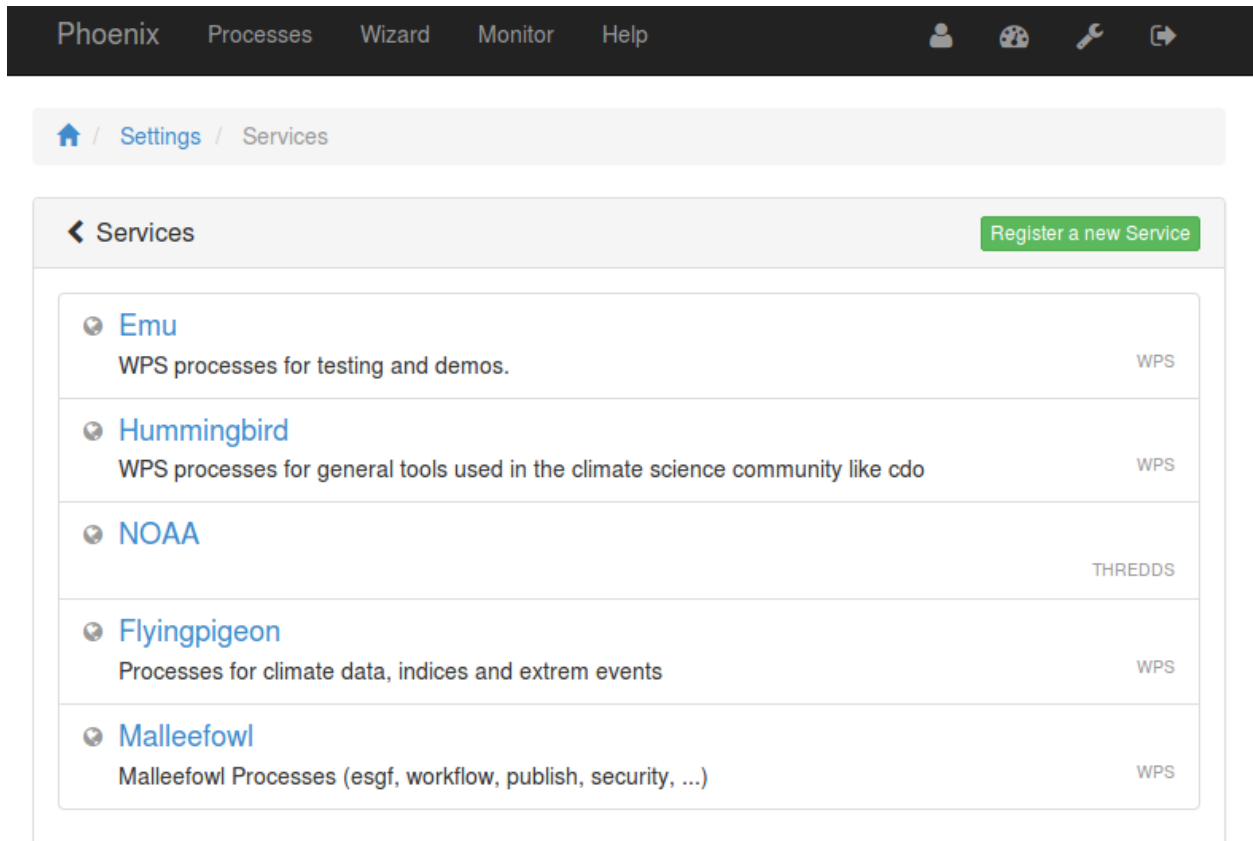
3.7 Settings (admins only)

When you are logged-in as admin user you have the `Settings` page. Here you can make administrative changes and monitor services.

The screenshot shows the 'Settings' page in the Phoenix application. The top navigation bar includes 'Phoenix', 'Processes', 'Wizard', 'Monitor', 'Help', and a user profile 'MacPingu'. Below the navigation bar, there is a breadcrumb trail: 'Home / Settings'. The main content area displays a grid of service tiles. The tiles are arranged in two rows. The first row contains: 'Supervisor' (with a monitor icon), 'Services' (with a bookshelf icon), 'Solr' (with the Solr logo), 'Users' (with a cat mask icon), 'Monitor' (with a '1+1' sign icon), and 'Auth' (with a padlock icon). The second row contains: 'LDAP' (with a server rack icon) and 'GitHub' (with the GitHub Octocat icon).

3.7.1 Register a WPS or Thredds service

Open the `Settings/Services` page. Here you can see which services are registered in the catalog service (we are using `PyCSW`). All these services are known and useable by Phoenix.



To add a new WPS service, press the Register a new Service button and enter the WPS URL in the field Service URL:

- hummingbird: `http://localhost:8092/wps`
- flyingpigeon: `http://localhost:8093/wps`
- emu: `http://localhost:8094/wps`

For example, to register Malleefowl WPS:

`http://localhost:8091/wps`

[Home](#) / [Settings](#) / [Services](#) / Register New Service

Register New Service

Service URL *

Add URL of service (WPS, Thredds, ...). Example: http://localhost:8091/wps, http://localhost/thredds/catalog.xml

Service Name

An optional service name.

Service Type *

☒ Web Processing Service
☐ Thredds Catalog

[Register](#)

To add a new Thredds service press the Register a new Service button again, enter the Thredds URL and choose Thredds Catalog as service type.

[Home](#) / [Settings](#) / [Services](#) / Register New Service

Register New Service

Service URL *

Add URL of service (WPS, Thredds, ...). Example: http://localhost:8091/wps, http://localhost/thredds/catalog.xml

Service Name

An optional service name.

Service Type *

☐ Web Processing Service
☒ Thredds Catalog

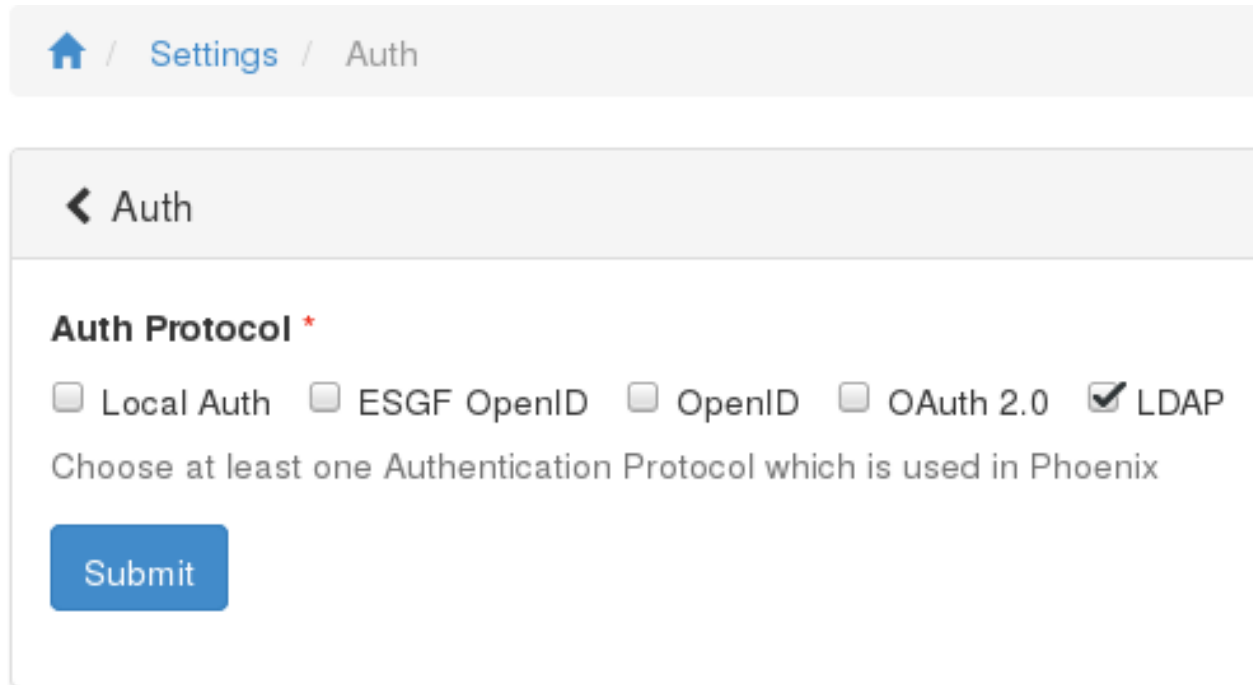
[Register](#)

3.7.2 Activate Users

Open the `Settings/Users` page. Here you activate/deactivate users and also remove them. When a user has registered to the Phoenix web application the user needs to be activated before the user can login.

3.7.3 Choose Authentication Protocol

Open the `Settings/Auth` page. Here you can choose the different authentication protocols (OpenID, LDAP, ...) which users can use on the login page. `Local Auth` enables the local admin account whose password is set in `custom.cfg` in your Phoenix installation.



Home / Settings / Auth

< Auth

Auth Protocol *

☐ Local Auth ☐ ESGF OpenID ☐ OpenID ☐ OAuth 2.0 ☒ LDAP

Choose at least one Authentication Protocol which is used in Phoenix

Submit

3.7.4 GitHub Support

You can use GitHub accounts to login to Phoenix. GitHub uses OAuth2. First you need to register your Phoenix application at [GitHub](#). Then go to `Settings/GitHub` in your Phoenix application and enter the GitHub Consumer Key/Client ID and GitHub Consumer Secret/Client Secret:

[Home](#) / [Settings](#) / [GitHub](#)

[<](#) **GitHub**

GitHub Consumer Key *

Register at GitHub: <https://github.com/settings/applications/new>

GitHub Consumer Secret *

3.7.5 LDAP Support

Basic support for authentication via LDAP has been added recently. To enable LDAP login for your environment, login with your admin account, navigate to `Settings/LDAP` and configure Phoenix to match your LDAP environment.

[Home](#) / [Settings](#) / LDAP**Server ***

URI of LDAP server to connect to, e.g. "ldap://ldap.example.com"

Use TLS *

Activate TLS when connecting

Bind

Bind to use for the LDAP connection, e.g. "CN=admin,DC=example,DC=com". Leave empty for anonymous bind.

Password

Password for the LDAP bind. Leave empty for anonymous bind.

Base DN *

DN where to begin the search for users, e.g. "CN=Users,DC=example,DC=com"

LDAP filter *

Is used to filter the LDAP search. Should always contain the placeholder "%(login)s". Example for OpenLDAP: "(uid=%(login)s)"; Example for MS AD: "(sAMAccountName=%(login)s)". Have a look at <http://pyramid-ldap.readthedocs.org/en/latest/> for more information.

Scope *

Scope to search in

User name attribute

Optional: LDAP attribute to receive user name from query, e.g. "cn"

User e-mail attribute

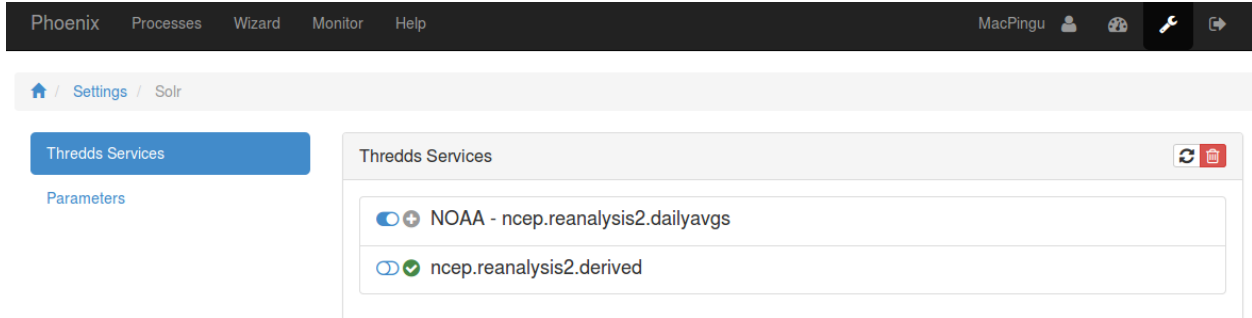
Optional: LDAP attribute to receive user e-mail from query, e.g. "mail"

There is no support for LDAP authorization yet. Use the `Settings/Users` backend to manage the access privileges

for your users. There will be an entry for each user that has been logged in once before.

3.7.6 Solr

You can publish the datasets of a registered Thredds service to a Solr index server. The Solr server is setup with the Phoenix installation.



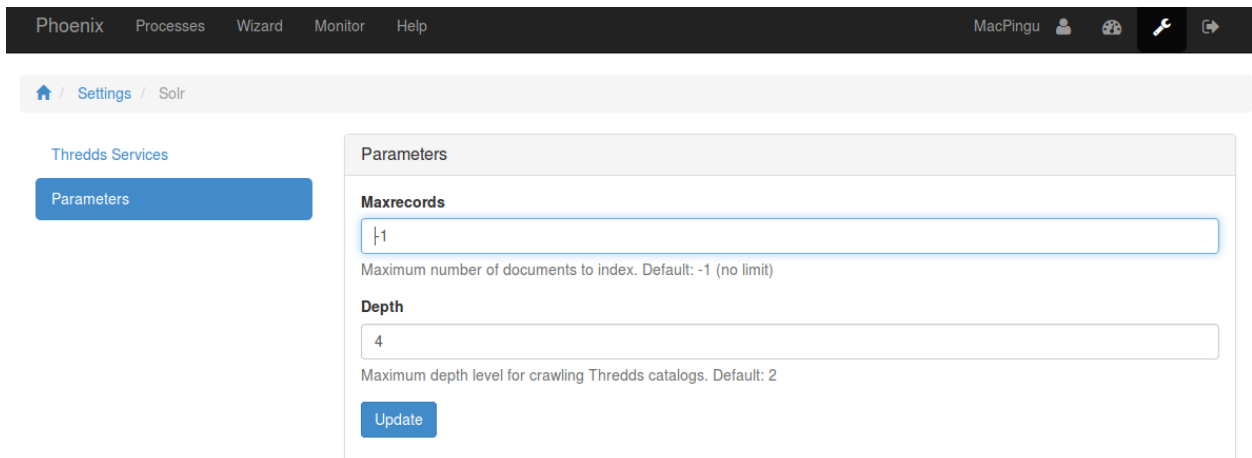
Use the toggle button on the left side of the Thredds service name to activate the publishing. Publishing takes some time. Use the reload button to update the status. The Solr search can then be used in the Wizard to select input files.

To clear the whole Solr index use the trash button.

The publisher has two parameters.

maxrecords Maximum number of datasets that will be published. Use -1 for unlimited.

depth The maximum depth level when crawling Thredds catalogs. Default is 2.



CHAPTER 4

Tutorial

In the following tutorial will guide you though the first steps to get familliar with Phoenix.

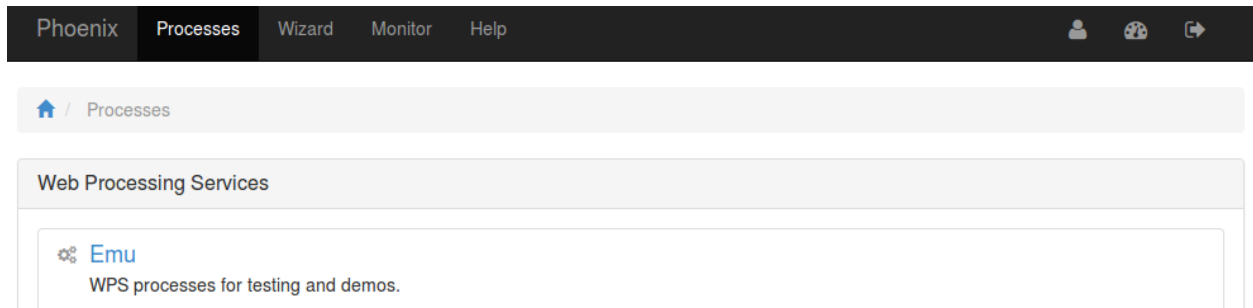
4.1 Hello World

First you need to login. Please follow the login instructions in the [user guide](#).

- *Select Emu WPS Service*
- *Choose Hello World Process*
- *Enter Process Parameters*
- *Monitor running Job*
- *Display the outputs*

4.1.1 Select Emu WPS Service

For this example choose the Emu WPS service which has test processes. For this go to the `Processes` tab.



4.1.2 Choose Hello World Process

With clicking on *Emu* you will get the list of available processes in Emu.

The screenshot shows the Phoenix application interface. At the top is a dark navigation bar with the menu items 'Phoenix', 'Processes', 'Wizard', 'Monitor', and 'Help'. To the right of the menu are three icons: a user profile, a gear, and a share icon. Below the navigation bar is a breadcrumb trail: a home icon followed by 'Processes / Emu'. The main content area is divided into two sections. The first section, titled 'Description', contains the text 'WPS processes for testing and demos.' and two tags: an orange 'XML' tag and a blue 'Provider: Birdhouse/Emu' tag. The second section, titled 'Processes', contains a list of six processes, each with a gear icon, a title, and a description. The processes are: 'Hello World 1.0' (Welcome user and say hello ...), 'Answer to Life, the Universe and Everything 2.0' (Numerical solution that is the answer to Life, Universe and Everything. The process is an improvement to Deep Thought computer (therefore version 2.0) since it no longer takes 7.5 million years, but only a few seconds to give a response, with an update of status every 10 seconds.), 'Chomsky text generator 1.0' (Generates a random chomsky text ...), 'Word Counter 1.0' (Counts words in a given text ...), 'Testing all Data Types 0.2' (Just testing data types like date, datetime etc ...), and 'Multiple Sources 1.0' (Process with multiple different sources ...).

Phoenix Processes Wizard Monitor Help

Home / Processes / Emu

Description

WPS processes for testing and demos.




XML Provider: Birdhouse/Emu

Processes

- Hello World 1.0**
Welcome user and say hello ...
- Answer to Life, the Universe and Everything 2.0**
Numerical solution that is the answer to Life, Universe and Everything. The process is an improvement to Deep Thought computer (therefore version 2.0) since it no longer takes 7.5 million years, but only a few seconds to give a response, with an update of status every 10 seconds.
- Chomsky text generator 1.0**
Generates a random chomsky text ...
- Word Counter 1.0**
Counts words in a given text ...
- Testing all Data Types 0.2**
Just testing data types like date, datetime etc ...
- Multiple Sources 1.0**
Process with multiple different sources ...

4.1.3 Enter Process Parameters

Click on *Hello World* and you will get a form to enter the process parameter:

Phoenix Processes Wizard Monitor Help   

[Home](#) / [Processes](#) / [Emu](#) / Hello World

Description

Welcome user and say hello ...

Inputs

Your name *




Please enter your name

Submit


Enter your name and click submit.

4.1.4 Monitor running Job

The job is now submitted and can be monitored on the *Monitor* page:

Phoenix Processes Wizard Monitor Help   

[Home](#) / [Monitor](#)

Status	Job	Process	Service	Duration	Finished	Progress
	3190b956-1c17-11e5-9569-68f72837e1b4	helloworld	Emu	0:00:01	less than 1 minute ago	<div>100%</div>

4.1.5 Display the outputs

Click on the Job ID link to get to the result of the submitted process.

Phoenix
Processes
Wizard
Monitor
Help

/ Monitor / Details

helloworld
Remove Job

Welcome user and say hello ...

Status
ProcessSucceeded
Duration
0:00:01
Finished
1 minute ago

Progress
100%
Status Location
XML

Status Message
100

Outputs
Log

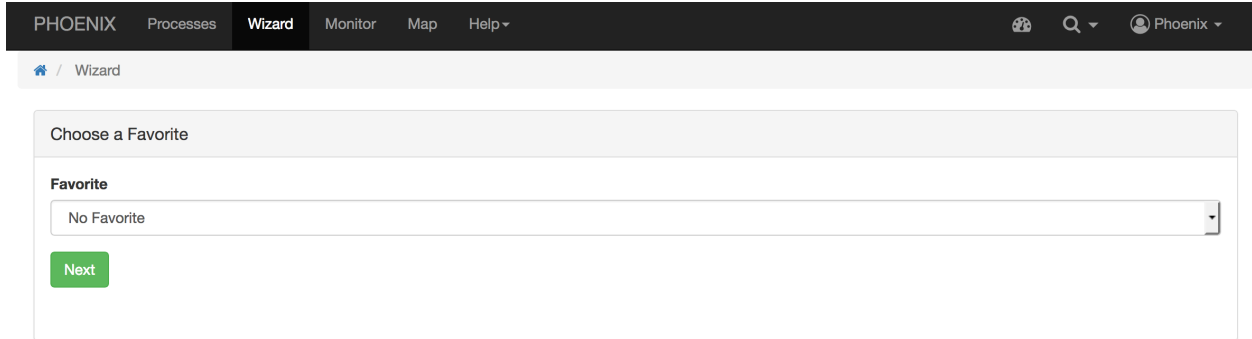
Output	Value
Welcome message	Hello pingu and welcome to WPS :)

4.2 Run CDO sinfo on data from Thredds Dataservice

First you need to login. Please follow the login instructions in the *user guide*.

- *Use the Wizard*
- *Select Hummingbird WPS Service*
- *Choose “CDO sinfo” Process*
- *Choose Input Parameter*
- *Choose Thredds as Source*
- *Choose Thredds Service*
- *Choose Data from Thredds Catalog*
- *Start Process*
- *Monitor running Job*
- *Display the outputs*

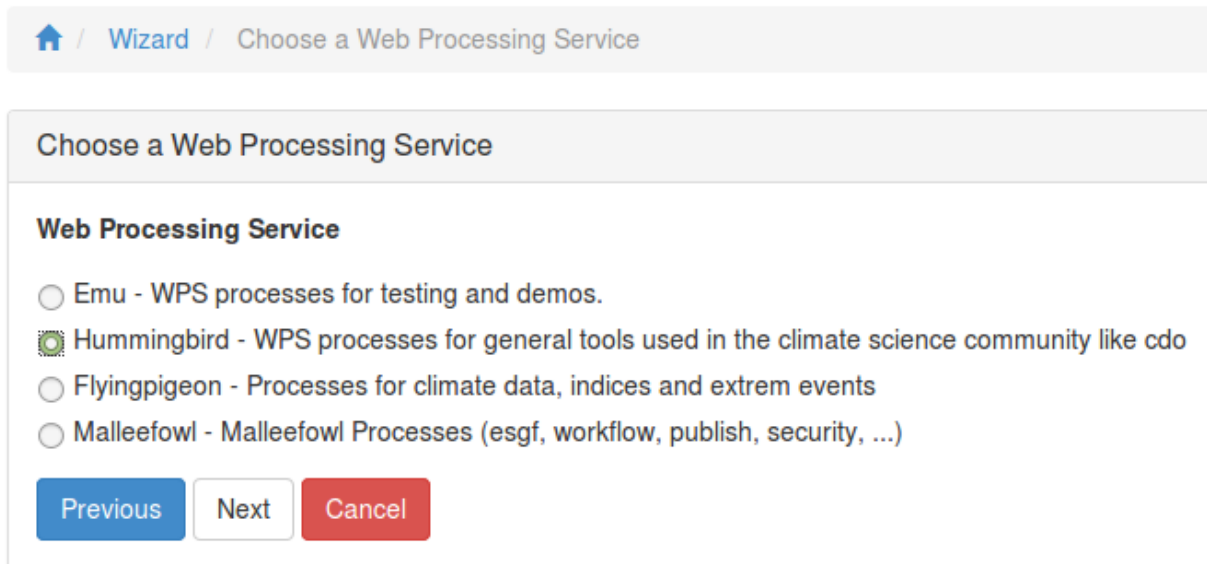
4.2.1 Use the Wizard



The screenshot shows the Phoenix application interface. The top navigation bar includes 'PHOENIX', 'Processes', 'Wizard' (highlighted), 'Monitor', 'Map', and 'Help'. On the right, there are icons for a globe, search, and a user profile labeled 'Phoenix'. Below the navigation bar, a breadcrumb trail shows a home icon followed by 'Wizard'. The main content area is titled 'Choose a Favorite'. It contains a 'Favorite' section with a dropdown menu currently showing 'No Favorite'. Below the dropdown is a green 'Next' button.


4.2.2 Select Hummingbird WPS Service

For this example choose the Hummingbird WPS service which has CDO processes.



The screenshot shows the Phoenix application interface at the 'Choose a Web Processing Service' step. The breadcrumb trail is 'Home / Wizard / Choose a Web Processing Service'. The main content area is titled 'Choose a Web Processing Service'. Below this, the section 'Web Processing Service' lists four options with radio buttons: 'Emu - WPS processes for testing and demos.', 'Hummingbird - WPS processes for general tools used in the climate science community like cdo' (which is selected), 'Flyingpigeon - Processes for climate data, indices and extrem events', and 'Malleefowl - Malleefowl Processes (esgf, workflow, publish, security, ...)'. At the bottom, there are three buttons: 'Previous' (blue), 'Next' (white), and 'Cancel' (red).

4.2.3 Choose “CDO sinfo” Process

 / Wizard / Choose WPS Process

Choose WPS Process of Hummingbird

Process

☐ NetCDF Metadata - Retrieve Metadata of NetCDF File

☒ CDO sinfo - Apply CDO sinfo on NetCDF File.

☐ CDO Operation - Apply CDO Operation like monmax on NetCDF File.


☐ CF Checker - The cfchecker checks NetCDF files for compliance to the CF standard.

Previous

Next

Cancel

4.2.4 Choose Input Parameter

 / Wizard / Choose Input Parameter

Choose Input Parameter of CDO sinfo

Input Parameter


☒ NetCDF File - NetCDF File (application/x-netcdf)

Previous

Next

Cancel

4.2.5 Choose Thredds as Source


 / [Wizard](#) / Choose Data Source

Choose Data Source

Source

☐ Earth System Grid (ESGF)

☐ Swift Cloud

☒  Thredds Catalog Service


[Previous](#) [Next](#) [Cancel](#)

4.2.6 Choose Thredds Service

[Home](#) / [Wizard](#) / [Select Thredds Service](#)

Select Thredds Service

Thredds Service

 NOAA - <http://www.esrl.noaa.gov/psd/thredds/catalog.html>

[Previous](#) [Next](#) [Cancel](#)

4.2.7 Choose Data from Thredds Catalog

[Home](#) / [Wizard](#) / [Thredds browser](#)

[< http://www.esrl.noaa.gov/psd/thredds/catalog/Datasets/ncep.reanalysis2.dailyavgs/surface/catalog.xml](http://www.esrl.noaa.gov/psd/thredds/catalog/Datasets/ncep.reanalysis2.dailyavgs/surface/catalog.xml)

Name	Size	Modified
hgt.sfc.nc	24.0 kB	2010-08-19 21:22:40
mslp.1979.nc	7.7 MB	2011-06-14 00:17:11
mslp.1980.nc	7.7 MB	2011-06-14 00:17:05

4.2.8 Start Process

[Home](#) / [Wizard](#) / Done

Done

Save as Favorite

☐

Favorite Name

[Previous](#) [Done](#) [Cancel](#)

4.2.9 Monitor running Job

The job is now submitted and can be monitored on the *Monitor* page:

[Home](#) / Monitor

Status	Job	Process	Service	Duration	Finished	Progress
✓	89f79b12-1c19-11e5-8de2-68f72837e1b4	cdo_sinfo	Hummingbird	0:00:11	???	<div>100%</div>
✓	3190b956-1c17-11e5-9569-68f72837e1b4	helloworld	Emu	0:00:01	16 minutes ago	<div>100%</div>

4.2.10 Display the outputs

Click on the Job ID link to get to the result of the submitted process.

Job Log

[Home](#) / [Monitor](#) / Details

[← cdo_sinfo](#)
Remove Job

Status ProcessSucceeded Duration 0:00:11 Finished less than 1 minute ago	Progress 100% Status Location XML	Status Message 100
--	---	------------------------------

Outputs

Log

```

1 0%: Process workflow accepted
2 10%: processstarted thredds_download: status_location=http://localhost:8090/wpsoutputs/malleefowl/pywps-8a2240b0-1c19-11e5-8306-68f72837e1b4.xml
3 10%: processstarted thredds_download: Process thredds_download accepted
4 10%: processstarted thredds_download: processstarted start downloading of 109 files
5 16%: processstarted thredds_download: processstarted Downloaded 18/109
6 25%: processstarted thredds_download: processstarted Downloaded 42/109
7 33%: processstarted thredds_download: processstarted Downloaded 65/109
8 50%: processstarted cdo_sinfo: status_location=http://localhost:8090/wpsoutputs/hummingbird/pywps-8dc43ed0-1c19-11e5-83d7-68f72837e1b4.xml
9 50%: processstarted cdo_sinfo: Process cdo_sinfo accepted
10 50%: processstarted cdo_sinfo: processstarted starting cdo sinfo
11 100%: PyWPS Process workflow successfully calculated

```

Job Outputs

[Home](#) / [Monitor](#) / Details

[← cdo_sinfo](#)
Remove Job

Status ProcessSucceeded Duration 0:00:11 Finished 2 minutes ago	Progress 100% Status Location XML	Status Message 100
---	---	------------------------------

Outputs

Log

Output	Value
CDO sinfo result	<div> text/plain Copy View Download Info </div> <div>CDO sinfo result</div>

4.3 Run CDO ensemble operation on CMIP5 data from ESGF

First you need to login. Please follow the login instructions in the [user guide](#).

- *Use the Wizard*
- *Select Hummingbird WPS Service*
- *Choose “CDO Ensembles Operation” Process*
- *Choose CDO ensmean Operator*
- *Choose Input Parameter*
- *Choose ESGF as Source*
- *Update your ESGF credentials if asked*
- *Select ensembles of CMIP5 experiment*
- *Start Process*
- *Monitor running Job*
- *Display the outputs*

4.3.1 Use the Wizard

The screenshot shows the Phoenix application interface with the 'Wizard' tab selected. The breadcrumb trail is 'Phoenix / Wizard'. The main content area is titled 'Choose a Favorite'. Below this, there is a 'Favorite' section with a dropdown menu currently showing 'No Favorite'. A green 'Next' button is located below the dropdown.

4.3.2 Select Hummingbird WPS Service

For this example choose the Hummingbird WPS service which has CDO processes.

The screenshot shows the Phoenix application interface with the 'Wizard' tab selected. The breadcrumb trail is 'Phoenix / Wizard / Choose a Web Processing Service'. The main content area is titled 'Choose a Web Processing Service'. Below this, there is a 'Web Processing Service' section with a list of radio buttons. The 'Hummingbird 0.5_dev' option is selected. Below the list are three buttons: 'Previous' (orange), 'Cancel' (red), and 'Next' (green).

4.3.3 Choose “CDO Ensembles Operation” Process

Choose WPS Process of Hummingbird 0.5_dev

Process *

- ☐ NCDump - Run ncdump to retrieve NetCDF header metadata.
- ☐ Spot Checker - Checks a single uploaded or remote dataset against a variety of compliance standards.
- ☐ IOOS Compliance Checker - Runs the IOOS Compliance Checker tool to check datasets against compliance standards.
- ☐ CF Checker by CEDA - The NetCDF Climate Forecast Conventions compliance checker by CEDA.
- ☐ CMIP6 CMOR Checker - Calls the CMIP6 cmor checker to verify CMIP6 compliance.
- ☐ CF Checker by DKRZ - The NetCDF Climate Forecast Conventions compliance checker by DKRZ.
- ☐ Quality Assurance Checker by DKRZ - The Quality Assurance checker QA-DKRZ checks conformance of meta-data of clir
- ☐ CDO sinfo - Runs CDO to retrieve NetCDF metadata information.
- ☐ CDO Operation - Calls CDO operations like monmax on a NetCDF file.
- ☐ CDO Copy - Calls CDO to copy or concatenate datasets.
- ☐ CDO select lon/lat box - Runs CDO to clip a bounding-box from a NetCDF file.
- ☐ CDO Climate Indices - Calculates climate indices like summer days using CDO.
- ☒ CDO Ensembles Operations - Calling cdo to calculate ensembles operations.
- ☐ CDO Remapping - CDO Remapping of NetCDF File(s) with multiprocessing.

Previous

Cancel

Next

4.3.4 Choose CDO ensmean Operator

Literal inputs of CDO Ensembles Operations

Remote OpenDAP Data URL *

Remote OpenDAP Data URL

Add Remote OpenDAP Data URL

Ensemble command *

ensmean

Choose a CDO Operator

Previous

Cancel

Next

4.3.5 Choose Input Parameter

Choose Input Parameter of CDO Ensembles Operations

Input Parameter *

☒ Dataset - You may provide a URL or upload a NetCDF file. (application/x-netcdf)

Previous

Cancel

Next

4.3.6 Choose ESGF as Source

PHOENIX Processes Wizard

[Home](#) / [Wizard](#) / Choose Data Source

Choose Data Source for Dataset

Source *

☒ Earth System Grid (ESGF)
☐ Thredds Catalog Service

Previous

Cancel

Next

4.3.7 Update your ESGF credentials if asked

Error: You are not allowed to access ESGF data. Please **update** your ESGF credentials.

ESGF Logon

Provider *

☐ CEDA (England) ☒ DKRZ (Hamburg, Germany) ☐ IPSL (Paris, France)

Choose your ESGF provider.

Username *

Type your username for your ESGF account.

Password *

Type your password for your ESGF account.

Submit

ESGF logon was successful.

4.3.8 Select ensembles of CMIP5 experiment

The screenshot shows the Phoenix Wizard interface for selecting ensembles of CMIP5 experiment. The interface is divided into several sections:

- TEXT:** A search bar labeled "Search datasets..." with a magnifying glass icon.
- OPTIONS:** A list of checkboxes for search options:
 - ☐ Distributed Search
 - ☐ Including Replicas
 - ☒ Latest Version
 - ☒ Temporal Extent
 - ☐ BBox Extent
- DATE:** Fields for "Start Year:" (2001) and "End Year:" (2005).
- SELECTION:** A section with a "Clear" button and a list of selected criteria:
 - cmor_table:Amon
 - data_node:esgf1.dkrz.de
 - forcing:GHG
 - Oz
 - SD
 - SI
 - VI
 - LU
 - format:netCDF
 - CF-1.4
 - index_node:esgf-data.dkrz.de
 - product:output1
 - version:20120503
 - project:CMIP5
 - time_frequency:mon
 - realm:atmos
 - experiment:historical
 - institute:MPI-M
 - variable:tas
 - model:MPI-ESM-MR
- CATEGORIES:** A list of category tags: access, cf_standard_name, ensemble, experiment_family, variable, variable_long_name.
- KEYWORDS:** A section labeled "ensemble" with three tags: r11p1, r21p1, r31p1.

At the bottom, there are three buttons: "Previous" (orange), "Cancel" (red), and "Next" (green).

4.3.9 Start Process

The screenshot shows the Phoenix Wizard interface for starting a process. The interface is divided into several sections:

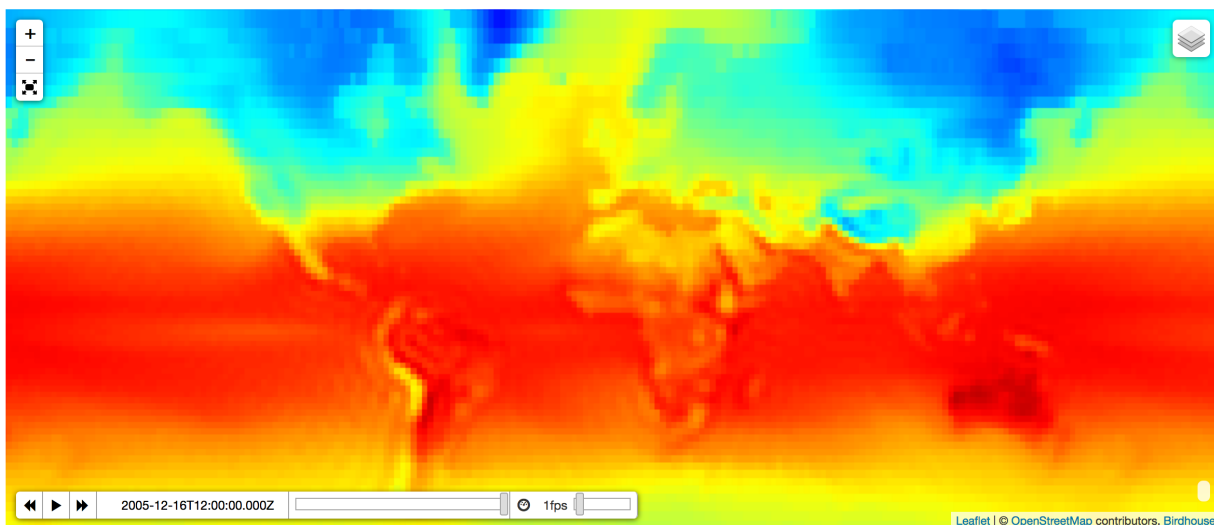
- Done:** A section with a "Caption" label and a text input field containing "ensmean cmip5 tas mpi historical 3 ensembles".
- Add an optional title for this job.** A text input field.

At the bottom, there are three buttons: "Previous" (orange), "Cancel" (red), and "Done" (green).

4.3.10 Monitor running Job

The job is now submitted and can be monitored on the *Monitor* page:

Map cdo_ensmean.nc



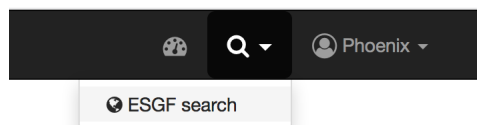
4.4 Run CDO ensemble operation on CORDEX data from ESGF using OpenDAP

First you need to login. Please follow the login instructions in the *user guide*.

- *Search and select CORDEX ensembles*
- *Check the selected files in Cart (optional)*
- *Select Hummingbird WPS Service*
- *Choose “CDO Ensembles Operation” Process*
- *Choose CDO ensmean Operator and OpenDAP datasets*
- *Monitor running Job*
- *Display the outputs*

4.4.1 Search and select CORDEX ensembles

Activate ESGF Search



Update ESGF credentials if asked


Warning: You are not allowed to access ESGF data. Please **update** your ESGF credentials.

Search CORDEX Ensemble

PHOENIX Processes Wizard Monitor Map Help ▾

ESGF Search

▼ TEXT

Search datasets... 

▼ OPTIONS

☐ Distributed Search

☐ Including Replicas

☒ Latest Version

☒ Temporal Extent

☐ BBox Extent

▼ DATE

Start Year:

2001

End Year:

2005

SELECTION **2** Clear

cf_standard_name:air_temperature data_node:esgf1.dkrz.de driving_model:MPI-M-MPI-ESM-LR index_node:esgf-data.dkrz.de

product:output rcm_name:REMO2009 rcm_version:v1 variable_long_name:Near-Surface Air Temperature version:20150609

project:CORDEX × time_frequency:mon × variable:tas × experiment:historical × institute:MPI-CSC × domain:EUR-44 ×

CATEGORIES

access ensemble experiment_family

KEYWORDS: domain

EUR-44

DATASETS


➤ cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r1i1p1.REMO2009.v1.mon.tas 19.8 MB 7


➤ cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r2i1p1.REMO2009.v1.mon.tas 19.4 MB 7


Select Files (OpenDAP)

DATASETS


▼ cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r1i1p1.REMO2009.v1.mon.tas 19.8 MB 7


cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r1i1p1.REMO2009.v1.mon.tas 

 Catalog



cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r1i1p1.REMO2009.v1.mon.tas.tas.20150609.aggregation 

tas air_temperature

 OpenDAP

tas_EUR-44_MPI-M-MPI-ESM-LR_historical_r1i1p1_MPI-CSC-REMO2009_v1_mon_200101-200512.nc 

tas air_temperature MPI-CSC historical EUR-44 mon 1.9 MB

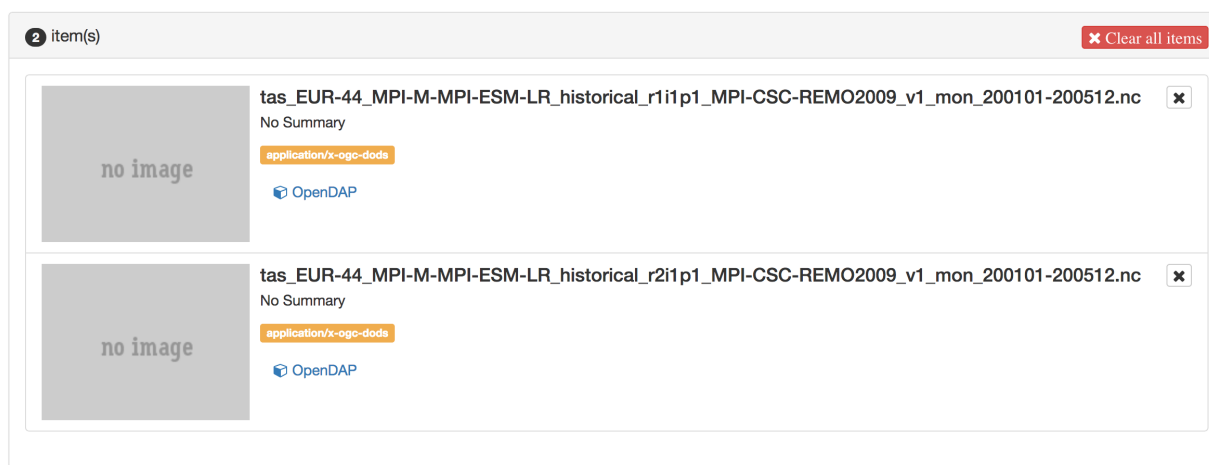
 Download  OpenDAP

➤ cordex.output.EUR-44.MPI-CSC.MPI-M-MPI-ESM-LR.historical.r2i1p1.REMO2009.v1.mon.tas 19.4 MB 7

4.4.2 Check the selected files in Cart (optional)



Cart



4.4.3 Select Hummingbird WPS Service



Choose the Hummingbird WPS service which has CDO processes.



 **Hummingbird 0.5_dev** Please choose one of the processes to submit a job.

WPS processes for general tools used in the climate science community like cdo and compliance checker.

[Capabilities \(XML\)](#) [Hummingbird](#)

-  **NCDump 4.4.1.1** ★ 3
Run ncdump to retrieve NetCDF header metadata.
-  **Spot Checker 0.3.0** ★ 3
Checks a single uploaded or remote dataset against a variety of compliance standards. The dataset is either in the NetCDF format or a remote OpenDAP resource. Available compliance standards are the Climate and Forecast conventions (CF) and project specific rules for CMIP6 and CORDEX.

4.4.4 Choose “CDO Ensembles Operation” Process

PHOENIX Processes Wizard Monitor Map Help

/ Processes / hummingbird / ensembles

⚙️ CDO Ensembles Operations

Please complete the form below and submit a job.

Calling cdo to calculate ensembles operations.

[View as XML](#) [Birdhouse](#) [User Guide](#) [CDO Homepage](#) [CDO Documentation](#)

4.4.5 Choose CDO ensmean Operator and OpenDAP datasets

Remote OpenDAP Data URL *

Remote OpenDAP Data URL

URL

URL

[Add Remote OpenDAP Data URL](#)

Ensemble command *

Choose a CDO Operator

4.4.6 Monitor running Job

The job is now submitted and can be monitored on the *Monitor* page:

Job Monitor This page shows the status of all your jobs.

Process Status

Running **1** Finished **52** Matching **53**

<input type="checkbox"/>	Status	User	Process	Service	Caption	Finished	Duration	Labels
<input type="checkbox"/>	Running	Phoenix	ensembles	hummingbird	???	???	0:00:10	<input type="button" value="dev"/> <input type="button" value="single"/> <input type="button" value="async"/> <input type="button" value="edit labels"/> <input type="button" value="Details"/> <input type="button" value="Restart"/>

4.4.7 Display the outputs

Click on the `Details` button to get to the result of the submitted process.

Outputs

Job Details This page shows the job details and polls the status of a running job.

✓ ensembles

???

dev

single

async

100%

⌚ Ran for 0:00:30

📅 less than 1 minute ago

💬 **ProcessSucceeded**

PyWPS Process CDO Ensembles

Operations finished

Delete Job

Restart Job

Calling cdo to calculate ensembles operations.

[Job Log](#)
[Inputs](#)
[Outputs](#)
[View as XML](#)

no image

NetCDF Output

Parameter `output`, a WPS ComplexType

CDO ensembles result.

cdo_ensmean.nc

application/x-netcdf

[Download](#)
[Share](#)
[Show on Map](#)

Map

Map cdo_ensmean.nc

4.5 Creating a timeseries plot

First you need to login. Please follow the login instructions in the *user guide*.

Once the login procedure is done, processes are operable and data search and download within the ESGF data archive is possible.

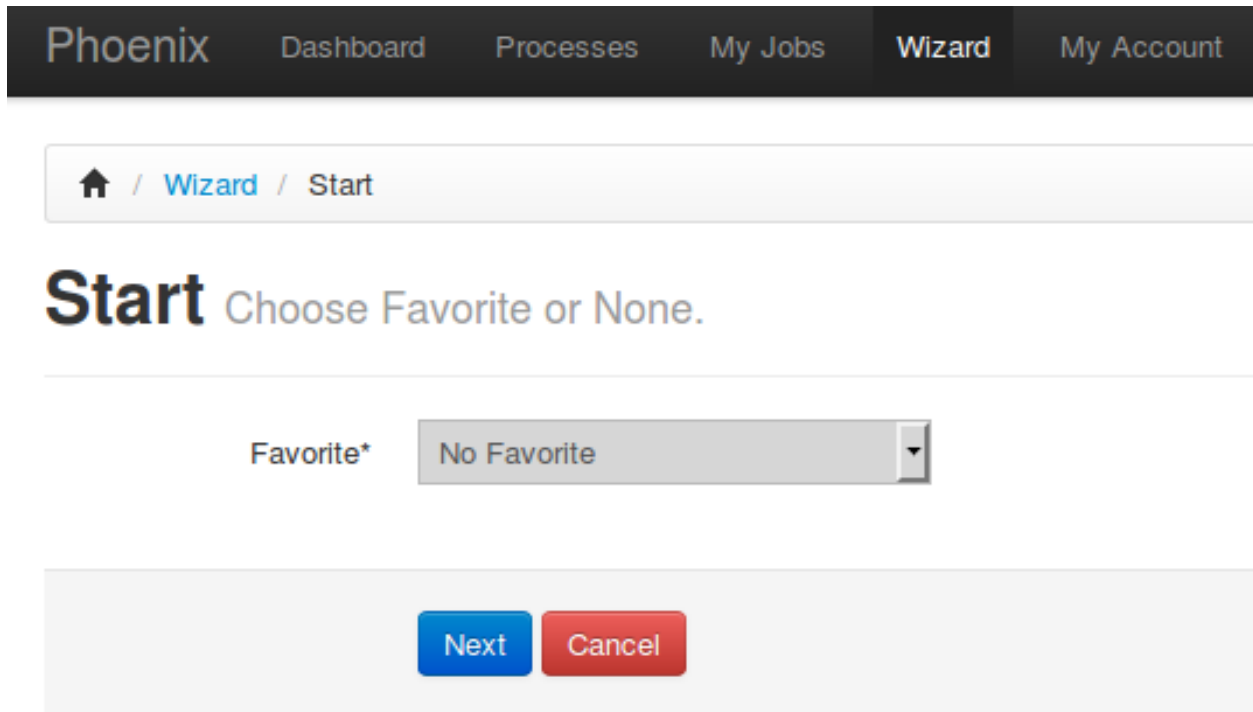
In this timeseries plot example we will use the *Flyingpigeon* WPS. Make sure *Flyingpigeon* is installed and running and check that it is *registered in Phoenix*.

44

Chapter 4. Tutorial

There are two ways to submit a job: either with *Processes* or *Wizard*.

While with *Processes* you can select single operational processes the *Wizard* is guiding you through the necessary steps to submit a job. For getting an idea of the operation procedure choose the *Wizard* tab:



Phoenix Dashboard Processes My Jobs **Wizard** My Account

Home / Wizard / Start

Start Choose Favorite or None.

Favorite* No Favorite

Next Cancel

You could choose a favorite here of a previous run job but in this case please choose *No Favorite* and click *Next*.

The following steps are necessary to run a visualisation job:

- *Select WPS Service*
- *Choose Process*
- *Enter Process Parameters*
- *Select Data Source*
- *Search Input Files*
- *Check your credentials*
- *Start the process*
- *Monitor running Job*
- *Display the outputs*

4.5.1 Select WPS Service

For this example choose the Flyingpigeon WPS service which has processes for the climate impact community.

[Home](#) / [Wizard](#) / WPS

WPS Choose Web Processing Service

- WPS service*
- ☐ Malleefowl (Malleefowl Processes (esgf, workflow, publish, security, ...)) [http://localhost:8091/wps]
 - ☐ Hummingbird (WPS processes for general tools used in the climate science community like cdo) [http://localhost:8092/wps]
 - ☒ Flyingpigeon (Processes for climate data, indices and extrem events) [http://localhost:8093/wps]
 - ☐ Emu (WPS processes for testing and demos) [http://localhost:8094/wps]
 - ☐ C3 WPS (C3 WPS processes for testing and demos) [http://localhost:8095/wps]
 - ☐ ESMValTool (WPS processes for ESMValTool) [http://localhost:8096/wps]
- Select WPS

Previous

Next

Cancel

4.5.2 Choose Process

With clicking on *Next* you'll find the list of available processes. Check the *Visualisation of NetCDF files*.

[Home](#) / [Wizard](#) / Choose WPS Process

Choose WPS Process Flyingpigeon

- WPS Process*
- ☒ Visualisation of netcdf files [visualisation]
 - ☐ Extract Coordinate Points [extractpoints]
 - ☐ Days with analog pressure pattern [analogs]
 - ☐ Ensembles Operations [ensembles]
 - ☐ Climate indices (icclim) [indice]
 - ☐ Vector born diseases [vbd]
 - ☐ Segetal Flora [sflora]
 - ☐ EOBS to CORDEX [eobs_to_cordex]
 - ☐ Calculation of climate indice (simple) [simple_indice]
 - ☐ Calculation of multiple climate indices [multiple_indices]
 - ☐ Calculation of multiple climate indices with clipping [indices_clipping]
 - ☐ Simple Clipping [simple_clipping]

Previous

Next

Cancel

4.5.3 Enter Process Parameters

Click on *Next* which guides you to the process parameter:

[Home](#) / [Wizard](#) / Literal Inputs

Literal Inputs

Process Visualisation of netcdf files

Variable

Variable to be expected in the input files

[Previous](#) [Next](#) [Cancel](#)

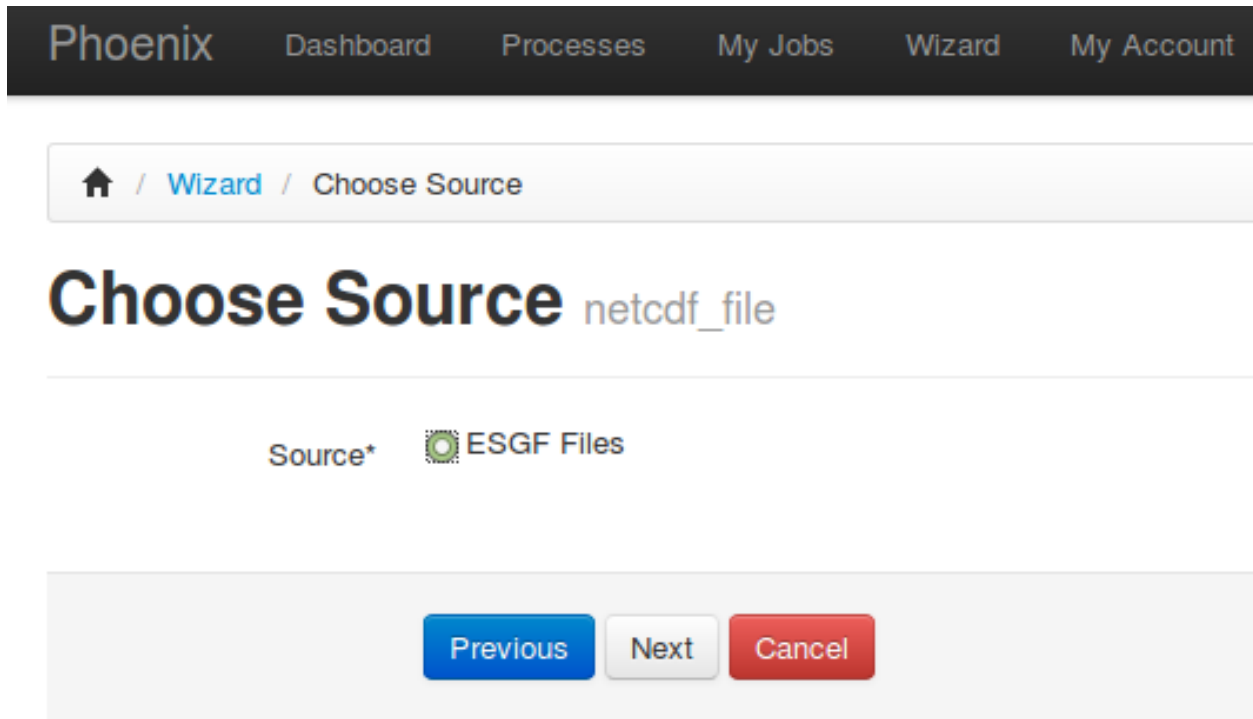
The values in the data files are stored with defined variable names. Here are the most common ones:

- tas – mean air temperature at 2m (in Kelvin)
- tasmin – minimum air temperature at 2m (in Kelvin)
- tasmax – maximum air temperature at 2m (in Kelvin)
- pr – precipitation flux at surface (in kg/second)
- ps – air pressure at surface
- huss – specific humidity (in Kg/Kg)

A list of available variable names used for [CMIP5](#) and [CORDEX](#) experiment can be found in the [Appendix B](#) of the [CORDEX archive design](#).

4.5.4 Select Data Source

In the next step you will choose the data source. Currently there is only the [ESGF](#) data archive:



4.5.5 Search Input Files

This is a search GUI to find appropriate files stored in ESGF data archive. By selecting a *Search Category* (blue buttons), you can choose the appropriate options (in orange).

In this example select the following parameter:

Categorie	Option
project	CORDEX
domain	WAS-44
insitute	MPI-CSC
variable	tas
time_frequency	day

Double selection (like two domains) can be realized with pressing *Ctrl* - tab.

For the visualisation process it is necessary that the selected variable (*tas*) is the same as the variable argument in the *Process Parameters*

And optionally you can set the time bounds:

```
Start: 2005-01-01T12:00:00Z
End:   2010-12-31T12:00:00Z
```

The Selection should look similar to the following screenshot:

Phoenix Dashboard Processes My Jobs Wizard My Account Settings

⌂ / Wizard / ESGF Search

ESGF Search*

Datasets found: 4

Options: ☐ All Sites ☐ Including Replicas ☒ Latest Version ☒ Temporal

Query

Current Selection

project:CORDEX ×

variable:tas ×

time_frequency:day ×

institute:MPI-CSC ×

domain:WAS-44 ×

Search Categories

experiment

experiment_family

Category: experiment

historical

rcp26

rcp45

rcp85

Start

End

Previous

Next

Cancel

4.5.6 Check your credentials

To access ESGF data you need an [x509](#) proxy certificate from ESGF. You can update your certificate in [My Account](#). The x509 proxy certificate is valid only for a few hours. The wizard checks if your certificate is still valid and if not you will be asked to update it on the following wizard page.

Phoenix Dashboard Processes My Jobs Wizard My Account Settings

⬆ / Wizard / ESGF Credentials

OpenID

OpenID from your ESGF provider

Password

Password for this OpenID

Previous Next Cancel

4.5.7 Start the process

On the final page *Done* of the wizard you can give some descriptive keywords for your process. You can also save it as a favorite so that later you can run the same job again.

PHOENIX Processes Wizard Monitor Map Help ▾

⬆ / Wizard / Done

Done

Caption

Add an optional title for this job.

Previous Cancel Done

Press *Done* and the job will start.

4.5.8 Monitor running Job

The job is now submitted and can be monitored on the *My Jobs* page:

The screenshot shows the Phoenix web interface. At the top is a navigation bar with links: Phoenix, Dashboard, Processes, My Jobs (active), Wizard, My Account, and Settings. Below this is a breadcrumb trail: Home / My Jobs / Overview. A red 'Remove all' button is visible. The main content area displays a table of jobs. The first job is titled 'Visualisation of netcdf files' with a description 'Just testing a nice script to visualise some variables'. It has a 'test' tag and a 'workflow' tag. The 'Workflow' column shows 'True'. The 'Status' column shows a blue 'ProcessStarted' button, followed by the text 'processstarted PyWPS Process wget successfully calculated' and an orange 'XML' button. The 'Creation Time' column shows '2015-03-17 10:24:47'. The 'Progress' column shows a progress bar at 5% and 'Show' and 'Remove' buttons.

Title	Workflow	Status	Creation Time	Progress
Visualisation of netcdf files Just testing a nice script to visualise some variables test workflow visualisation	True	ProcessStarted processstarted PyWPS Process wget successfully calculated XML	2015-03-17 10:24:47	5% Show Remove

The job is running ... data will be downloaded and the analyzing of the data starts. In this case, a field mean over the several experiments will be performed and an appropriate timeline drawn.

When the job has finished, the status bar is turning into green:

Status

The screenshot shows the job status after completion. It features a green 'ProcessSucceeded' button, followed by the text 'PyWPS Process dispel successfully calculated' and an orange 'XML' button.

4.5.9 Display the outputs

Click on the *Show* button to get to the result of the submitted process.

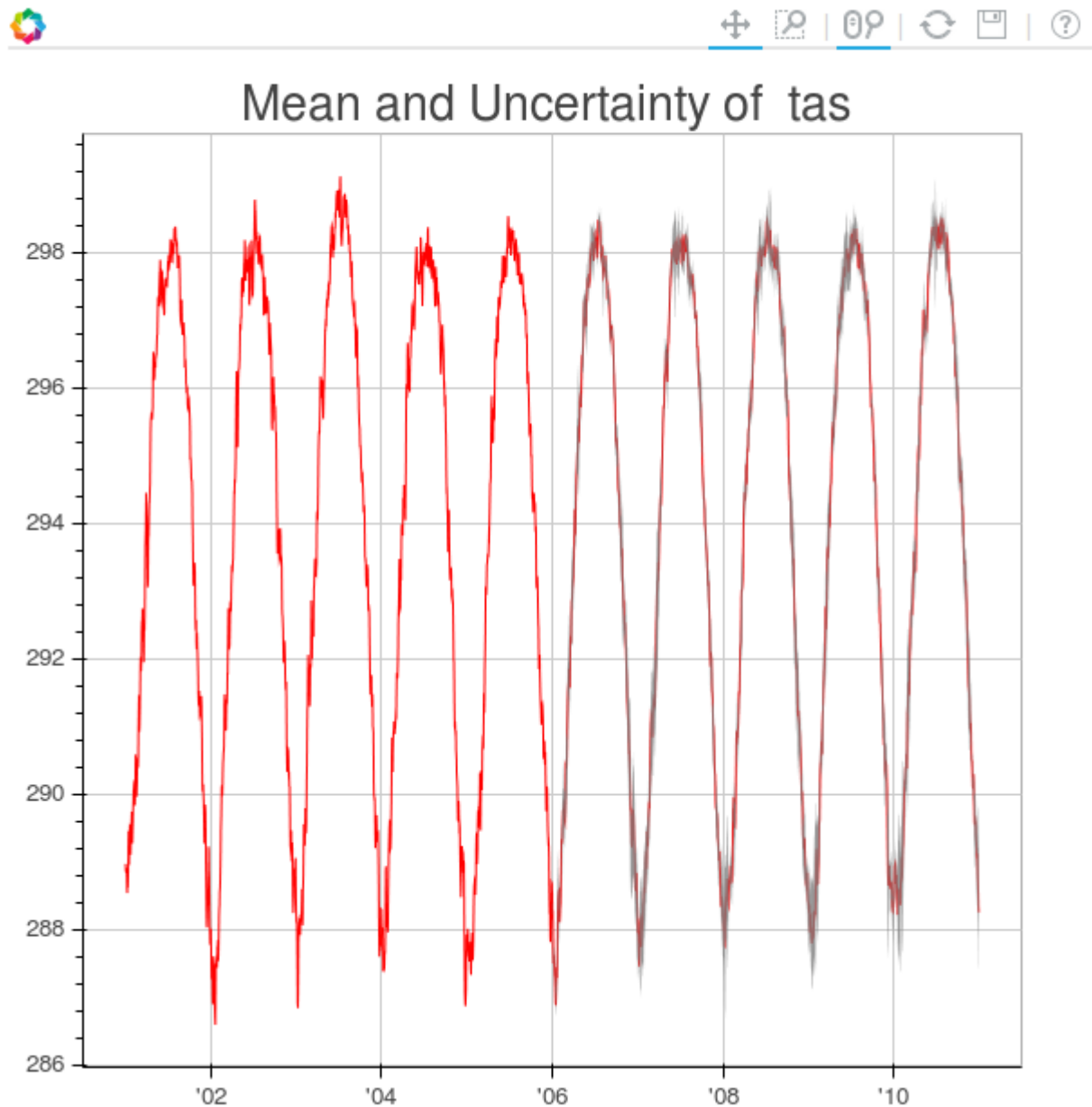
Phoenix Dashboard Processes My Jobs Wizard My Account Settings

Home / My Jobs / Process Outputs

Outputs Inputs Resources Workflow

Output	Identifier	Preview
visualisation Visualisation of single variables <code>application/html</code>	worker1.output	Publish View Show on Map

In this case, it is an URL pointing to a HTML page with an embedded interactive plot using `bokeh`. Opening it in a new browser tab gives the following result:



4.6 Use the Birdhouse Solr Search in the Wizard

First you need to login. Please follow the login instructions in the *user guide*.

- *Prepare Solr Search (Admins only)*
- *Use the Wizard*
- *Select Hummingbird WPS Service*

- Choose “CDO sinfo” Process
- Choose Input Parameter
- Choose Birdhouse Solr as Source
- Choose Data from Solr Search
- Start Process
- Monitor running Job
- Display the outputs

4.6.1 Prepare Solr Search (Admins only)

Register a thredds catalog in Settings/Services. For example use:

```
http://www.esrl.noaa.gov/psd/thredds/catalog/Datasets/ncep.reanalysis2.dailyavgs/
↪ catalog.html
```

Index this Thredds Catalog in Settings/Solr.

4.6.2 Use the Wizard

4.6.3 Select Hummingbird WPS Service

For this example choose the Hummingbird WPS service which has CDO processes.

[Home](#) / [Wizard](#) / Choose a Web Processing Service

Choose a Web Processing Service

Web Processing Service

- ☐ Emu - WPS processes for testing and demos.
- ☒ Hummingbird - WPS processes for general tools used in the climate science community like cdo
- ☐ Flyingpigeon - Processes for climate data, indices and extrem events
- ☐ Malleefowl - Malleefowl Processes (esgf, workflow, publish, security, ...)

[Previous](#) [Next](#) [Cancel](#)

4.6.4 Choose “CDO sinfo” Process

[Home](#) / [Wizard](#) / Choose WPS Process

Choose WPS Process of Hummingbird

Process

- ☐ NetCDF Metadata - Retrieve Metadata of NetCDF File
- ☒ CDO sinfo - Apply CDO sinfo on NetCDF File.
- ☐ CDO Operation - Apply CDO Operation like monmax on NetCDF File.
- ☐ CF Checker - The cfchecker checks NetCDF files for compliance to the CF standard.

[Previous](#) [Next](#) [Cancel](#)

4.6.5 Choose Input Parameter

[Home](#) / [Wizard](#) / Choose Input Parameter

Choose Input Parameter of CDO sinfo

Input Parameter

☒ NetCDF File - NetCDF File (application/x-netcdf)

[Previous](#) [Next](#) [Cancel](#)

4.6.6 Choose Birdhouse Solr as Source

[Home](#) / [Wizard](#) / Choose Data Source

Choose Data Source

Source

☐ Earth System Grid (ESGF)





☐ Swift Cloud

☐ Thredds Catalog Service

☒ Birdhouse Solr Search

[Previous](#) [Next](#) [Cancel](#)

4.6.7 Choose Data from Solr Search

Phoenix Processes **Wizard** Monitor Help MacPingu    

[Home](#) / [Wizard](#) / Solr Search

Solr Search

All

Thredds

Files

All Sources

Tag Cloud


1981

1982

<

>

Showing 1-2 of 2


 tmax.2m.gauss.1982.nc

Datasets/ncep.reanalysis2.dailyavgs/gaussian_grid/tmax.2m.gauss.1982.nc

thredds

application/netcdf

[Download](#) | [Catalog](#) | [OpenDAP](#) | [Create a Map](#)

 tmax.2m.gauss.1981.nc

Datasets/ncep.reanalysis2.dailyavgs/gaussian_grid/tmax.2m.gauss.1981.nc

thredds

application/netcdf

[Download](#) | [Catalog](#) | [OpenDAP](#) | [Create a Map](#)

Previous

Next

Cancel

4.6.8 Start Process

[Home](#) / [Wizard](#) / Done

Done

Save as Favorite

☐

Favorite Name

[Previous](#) [Done](#) [Cancel](#)

4.6.9 Monitor running Job

The job is now submitted and can be monitored on the *Monitor* page:

[Home](#) / Monitor

Status	Job	Process	Service	Duration	Finished	Progress
✓	89f79b12-1c19-11e5-8de2-68f72837e1b4	cdo_sinfo	Hummingbird	0:00:11	???	<div>100%</div>
✓	3190b956-1c17-11e5-9569-68f72837e1b4	helloworld	Emu	0:00:01	16 minutes ago	<div>100%</div>

4.6.10 Display the outputs

Click on the Job ID link to get to the result of the submitted process.

Job Log

Phoenix Processes Wizard **Monitor** Help MacPingu

[Home](#) / [Monitor](#) / Details

[<](#) cdo_sinfo ✔ Remove Job

Status ProcessSucceeded Duration 0:00:16 Finished 15 minutes ago	Progress 100% Status Location XML	Status Message 100
--	--	------------------------------

[Outputs](#) [Log](#)

```

1 0%: Process workflow accepted
2 10%: processstarted download: status_location=http://localhost:8090/wpsoutputs/malleeowl/pywps-e4fdede6-36c6-11e5-8664-68f72837e1b4.xml
3 10%: processstarted download: Process download accepted
4 10%: processstarted download: processstarted start downloading of 2 files
5 50%: processstarted cdo_sinfo: status_location=http://localhost:8090/wpsoutputs/hummingbird/pywps-ed66aa36-36c6-11e5-89f8-68f72837e1b4.xml
6 100%: PyWPS Process workflow successfully calculated
  
```

Job Outputs

Phoenix Processes Wizard **Monitor** Help MacPingu

[Home](#) / [Monitor](#) / Details

[<](#) cdo_sinfo ✔ Remove Job

Status ProcessSucceeded Duration 0:00:16 Finished 16 minutes ago	Progress 100% Status Location XML	Status Message 100
--	--	------------------------------

[Outputs](#) [Log](#)

Output	Value
CDO sinfo result	text/plain 📄 👁 🔍 🔔
CDO sinfo result	

- *Phoenix does not start*
- *Nginx does not start*

5.1 Phoenix does not start

Phoenix needs a running mongodb and pycsw service. Sometimes Phoenix is started when these service are not ready yet. In that case start theses services manually in the order mongodb, pycsw and Phoenix with:

```
$ source activate pyramid-phoenix      # activate conda environment used by phoenix
$ supervisorctl restart mongodb
$ supervisorctl restart pycsw
$ supervisorctl restart phoenix
```

You can also try to restart all services with:

```
$ supervisorctl restart all
```

or:

```
$ make restart
```

Check the log files to see the error messages:

```
$ tail -f ~/birdhouse/var/log/supervisor/phoenix.log
$ tail -f ~/birdhouse/var/log/supervisor/celery.log
```

5.2 Nginx does not start

From a former installation there might be nginx files with false permissions. Remove those files:

```
$ ~/birdhouse/etc/init.d/supervisord stop
$ sudo rm -rf ~/birdhouse/var/run
$ sudo rm -rf ~/birdhouse/var/log
$ ~/birdhouse/etc/init.d/supervisord start
```

This page is the top-level of your generated API documentation. Below is a list of all items that are documented here.

6.1 ldap

6.1.1 Module Contents

`ldap.includeme` (*config*)

6.2 __compat

This python 2.x/3.x compatibility modules is based on the pywps 4.x code.

6.3 db

6.3.1 Module Contents

`db.mongodb` (*registry*)

`db.includeme` (*config*)

6.4 events

6.4.1 Module Contents

`class` `events.JobStarted` (*request*, *task_id*)

```
class events.JobFinished(job)

    succeeded()

class events.SettingsChanged(request, new_settings)

    converted_settings()
```

6.5 layouts

6.5.1 Module Contents

```
class layouts.PageLayout(context, request)

    project_title()
    add_breadcrumb(route_path, title)
    add_heading(name, *args, **kw)
```

6.6 catalog

6.6.1 Module Contents

```
catalog.includeme(config)
catalog.catalog_factory(registry)
catalog._fetch_thredds_metadata(url, title=None)
    Fetch capabilities metadata from thredds catalog service and return record dict.
catalog._fetch_wps_metadata(url, title=None)
    Fetch capabilities metadata from wps service and return record dict.
class catalog.Catalog

    get_record_by_id(identifier)
    delete_record(identifier)
    insert_record(record)
    harvest(url, service_type, service_name=None, service_title=None, public=False, c4i=False)
    get_service_name(record)
        Get service name from twitcher registry for given service (url).
    get_service_by_name(name)
        Get service from twitcher registry by given service name.
    get_service_by_url(url)
        Get service from twitcher registry by given url.
    get_services(service_type=None, maxrecords=100)
```

```

    clear_services ()

class catalog.CatalogService (csw, service_registry)

    get_record_by_id (identifier)
    delete_record (identifier)
    insert_record (record)
    harvest (url, service_type, service_name=None, service_title=None, public=False, c4i=False)
    get_services (service_type=None, maxrecords=100)

catalog.doc2record (document)
    Converts document from mongodb to a Record object.

class catalog.MongodbCatalog (collection, service_registry)
    Implementation of a Catalog with MongoDB.

    get_record_by_id (identifier)
    delete_record (identifier)
    insert_record (record)
    harvest (url, service_type, service_name=None, service_title=None, public=False, c4i=False)
    get_services (service_type=None, maxrecords=100)
    clear_services ()

```

6.7 security

see pyramid security:

- <http://docs.pylonsproject.org/projects/pyramid/en/latest/tutorials/wiki2/authentication.html>

6.7.1 Module Contents

```

security.check_csrf_token (request)
security.has_execute_permission (request, service_name)
security.passwd_check (request, passphrase)
    code taken from IPython.lib.security TODO: maybe import ipython

```

```

>>> passwd_check ('sha1:0e112c3ddfce:a68df677475c2b47b6e86d0467eec97ac5f4b85a',
...               'anotherpassword')
False

```

```

security.groupfinder (userid, request)

class security.Root (request)
security.root_factory (request)
security.authomatic (request)
security.authomatic_config (request)

```

```
class security.MyAuthenticationPolicy
```

```
    authenticated_userid(request)
```

```
security.get_user(request)
```

```
security.includeme(config)
```

6.8 patch

6.8.1 Module Contents

```
patch.patch_myproxy_client()
```

Patch myproxycient to use MESSAGE_DIGEST_TYPE sha256.

6.9 __init__

6.9.1 Package Contents

```
__init__.main(global_config, **settings)
```

This function returns a Pyramid WSGI application.

6.10 panels

6.10.1 Module Contents

```
panels.navbar(context, request)
```

```
panels.messages(context, request)
```

```
panels.breadcrumbs(context, request)
```

```
panels.footer(context, request)
```

```
panels.headings(context, request)
```

6.11 exceptions

all Exceptions defined by Phoenix ...

6.12 wps

6.12.1 Module Contents

```
wps.is_opendap(data_input)
```

`wps.check_status` (*url=None, response=None, sleep_secs=2, verify=False*)

Run owslib.wps check_status with additional exception handling.

Parameters `verify` – Flag to enable SSL verification. Default: False

Returns OWSLib.wps.WPSExecution object.

`wps.appstruct_to_inputs` (*request, appstruct*)

Transforms appstruct to wps inputs.

class `wps.WPSSchema` (*request, hide_complex=False, process=None, use_async=False, user=None, **kw*)

Build a Colander Schema based on the WPS data inputs.

This Schema generator is based on: <http://colanderalchemy.readthedocs.io/en/latest/>

TODO: fix dataType in wps client

`add_async_check` ()

`add_nodes` (*process*)

`literal_data` (*data_input*)

`colander_literal_type` (*data_input*)

`colander_literal_widget` (*node, data_input*)

`bbox_data` (*data_input*)

`complex_data` (*data_input*)

`_url_node_default` (*data_input*)

`bind` (***kw*)

`clone` ()

6.13 utils

6.13.1 Module Contents

class `utils.ActionButton` (*name, title=None, no_children=False, href=None, new_window=False, disabled=False, css_class="btn btn-default", icon=None*)

`url` (*context, request*)

`permitted` (*context, request*)

`utils.pinned_processes` (*request*)

`utils.skip_csrf_token` (*appstruct*)

`utils.headline` (*text, max_length=120*)

`utils.make_tags` (*tags_str*)

`utils.format_tags` (*tags*)

`utils.localize_datetime` (*dt, tz_name="UTC"*)

Provide a timezone-aware object for a given datetime and timezone name

`utils.is_url` (*url*)

Check wheather given text is url or not

```
utils.build_url (url, query)
utils.wps_caps_url (url)
utils.wps_describe_url (url, identifier)
utils.time_ago_in_words (from_time)
utils.root_path (path)
```

6.14 grid

6.14.1 Module Contents

```
grid.get_value (record, attribute, default=None)
class grid.CustomGrid (request, *args, **kwargs)

    checkbox_column_format (column_number, i, record)
    render_td (renderer, **data)
    label_td (attribute, default=None)
    time_ago_td (attribute)
    timestamp_td (attribute)
    size_td (attribute)
    userid_td (attribute)
    user_td (attribute)
    render_title_td (title, abstract=None, keywords=list, data=list, format=None, source="#")
    render_flag_td (flag=False, tooltip="")
    render_format_td (format, source)
    render_preview_td (format, source)
    render_buttongroup_td (buttons=list)
    generate_header_link (column_number, column, label_text)
        Override of the ObjectGrid to customize the headers. This is mostly taken from the example code in
        ObjectGrid itself.
    default_header_column_format (column_number, column_name, header_label)
        Override of the ObjectGrid to use <th> for header columns
    default_header_ordered_column_format (column_number, column_name, header_label)
        Override of the ObjectGrid to use <th> and to add an icon that represents the sort order for the column.
```

6.15 twitcherclient

6.15.1 Module Contents

```
twitcherclient.includeme (config)
```



```
twitcherclient.twitcher_service_factory(registry)
twitcherclient.generate_access_token(registry, userid, valid_in_hours=1)
twitcherclient.is_public(registry, name)
```

6.16 views

6.16.1 Module Contents

```
class views.MyView(request, name, title, description=None)
```

```
    breadcrumbs()
```

```
views.not_found(request)
```

This special view just renders a custom 404 page. We do this so that the 404 page fits nicely into our global layout.

```
views.add_global(event)
```

```
views.unknown_failure(request, exc)
```

```
views.favicon_view(request)
```

```
views.robotstxt_view(request)
```

```
class views.Home(request)
```

```
    view()
```

6.17 tasks

6.17.1 Submodules

```
tasks.esgflogon
```

Module Contents

```
tasks.esgflogon.esgf_login(self, userid, hostname, username, password)
```

```
tasks.execute
```

Module Contents

```
tasks.execute.execute_process(self, url, service_name, identifier, inputs, outputs, async=True,
                               userid=None, caption=None)
```

`tasks.solr`

Module Contents

`tasks.solr.index_thredds` (*self*, *url*, *maxrecords=-1*, *depth=2*)

`tasks.solr.clear_index` (*self*)

`tasks.utils`

Module Contents

`tasks.utils.task_result` (*task_id*)

`tasks.utils.wait_secs` (*run_step=-1*)

`tasks.utils.dump_json` (*obj*)

`tasks.utils.save_log` (*job*, *error=None*)

`tasks.utils.add_job` (*db*, *task_id*, *process_id*, *title=None*, *abstract=None*, *service_name=None*,
service=None, *status_location=None*, *is_workflow=False*, *caption=None*,
userid=None, *async=True*)

`tasks.utils.get_access_token` (*userid*)

`tasks.utils.wps_headers` (*userid*)

`tasks.workflow`

Module Contents

`tasks.workflow.execute_workflow` (*self*, *userid*, *url*, *service_name*, *workflow*, *caption=None*)

6.18 processes

6.18.1 Subpackages

`processes.views`

Submodules

`processes.views.actions`

Module Contents

`processes.views.actions.includeme` (*config*)

class `processes.views.actions.ProcessesActions` (*context*, *request*)
Actions related to processes.

`list_processes` ()

`processes.views.execute`

Module Contents

`class` `processes.views.execute.ExecuteProcess` (*request*)

```
    breadcrumbs ()
    appstruct ()
    generate_form (formid="deform")
    process_form (form)
    execute (appstruct)
    view ()
```

`processes.views.execute_json`

Module Contents

`class` `processes.views.execute_json.ExecuteProcessJson` (*request*)

```
    jsonify (value)
    view ()
```

`processes.views.list`

Module Contents

`class` `processes.views.list.ProcessList` (*request*)

```
    view ()
```

`processes.views.list_json`

Module Contents

`class` `processes.views.list_json.ProcessListJson` (*request*)

```
    view ()
```

`processes.views.overview`

Module Contents

```
class processes.views.overview.Overview(request)
```

```
    wps_services()
    pinned_processes()
    view()
```

`processes.views.overview_json`

Module Contents

```
class processes.views.overview_json.OverviewJson(request)
```

```
    view()
```

6.18.2 Package Contents

`processes.includeme` (*config*)

6.19 providers

6.19.1 Submodules

`providers.esgfoopenid`

Providers which implement the `loopenid|_` protocol based on the `'python-openid'` library. .. warning:

This providers are dependent on the `|pyopenid|_` package.

Module Contents

```
class providers.esgfoopenid.MyFetcher
```

```
    _urlopen()
```

```
class providers.esgfoopenid.ESGFOpenID(*args, **kwargs)
    ESGF OpenID provider with a common provider url template "https://{hostname}/esgf-idp/
    openid/{username}".
```

6.20 storage

6.20.1 Submodules

`storage.views`

Module Contents

`storage.views.download(request)`

`storage.views.delete(request)`

A DELETE request. If found, deletes a file with the corresponding UUID from the servers filesystem.

`storage.views.upload(request)`

`storage.views.handle_delete(request, uuid)`

Handles a filesystem delete based on UUID.

`storage.views.handle_upload(request, attrs)`

Handle a chunked or non-chunked upload.

See example code: <https://github.com/FineUploader/server-examples/blob/master/python/flask-fine-uploader/app.py>

`storage.views.save_chunk(fs, path)`

Save an uploaded chunk.

Chunks are stored in chunks/

`storage.views.combine_chunks(total_parts, source_folder, dest)`

Combine a chunked file into a whole file again. Goes through each part, in order, and appends that part's bytes to another destination file.

Chunks are stored in chunks/

6.20.2 Package Contents

`storage.includeme(config)`

6.21 services

6.21.1 Subpackages

`services.views`

Submodules

`services.views.actions`

Module Contents

`class services.views.actions.ServiceActions(context, request)`

Actions related to service registration.

```
remove_service()
```

```
clear_services()
```

```
services.views.actions.includeme(config)
```

Pyramid includeme hook.

Parameters `config` (`pyramid.config.Configurator`) – app config

`services.views.registerservice`

Module Contents

```
class services.views.registerservice.Schema
```

```
class services.views.registerservice.RegisterService(request)
```

```
    breadcrumbs()
```

```
    generate_form()
```

```
    process_form(form)
```

```
    view()
```

`services.views.services`

Module Contents

```
class services.views.services.Services(request)
```

```
    breadcrumbs()
```

```
    details_view()
```

```
    list_view()
```

6.21.2 Package Contents

```
services.includeme(config)
```

6.22 wizard

6.22.1 Subpackages

`wizard.views`

Submodules

`wizard.views.complexinputs`

Module Contents

```
wizard.views.complexinputs.includeme (config)
wizard.views.complexinputs.deferred_widget (node, kw)
class wizard.views.complexinputs.Schema
class wizard.views.complexinputs.ComplexInputs (request)

    breadcrumbs ()
    schema ()
    success (appstruct)
    next_success (appstruct)
    view ()
    custom_view ()
```

`wizard.views.done`

Module Contents

```
wizard.views.done.includeme (config)
class wizard.views.done.DoneSchema
class wizard.views.done.Done (request)

    breadcrumbs ()
    schema ()
    workflow_description ()
    success (appstruct)
    next_success (appstruct)
    view ()
```

`wizard.views.esgfsearch`

Module Contents

`wizard.views.esgfsearch.includeme` (*config*)

class `wizard.views.esgfsearch.ESGFSearchView` (*request*)

```
    breadcrumbs ()
    schema ()
    appstruct ()
    next_ok ()
    next_success (appstruct)
    custom_view ()
```

`wizard.views.literalinputs`

Module Contents

`wizard.views.literalinputs.includeme` (*config*)

class `wizard.views.literalinputs.LiteralInputs` (*request*)

```
    breadcrumbs ()
    schema ()
    next_success (appstruct)
    view ()
    custom_view ()
```

`wizard.views.solrsearch`

Module Contents

`wizard.views.solrsearch.includeme` (*config*)

class `wizard.views.solrsearch.SolrSearch` (*request*)

```
    breadcrumbs ()
    schema ()
    appstruct ()
    next_success (appstruct)
    view ()
```


wizard.views.source

Module Contents

```
wizard.views.source.includeme (config)
class wizard.views.source.SourceSchemaNode

    after_bind (node, kw)
class wizard.views.source.Schema
class wizard.views.source.ChooseSource (request)

    breadcrumbs ()
    schema ()
    next_success (appstruct)
    view ()
```

wizard.views.start

Module Contents

```
wizard.views.start.includeme (config)
wizard.views.start.job_to_state (request, job_id)
class wizard.views.start.FavoriteSchema

    deferred_favorite_widget (kw)
class wizard.views.start.Start (request)

    schema ()
    appstruct ()
    success (appstruct)
    next_success (appstruct)
    view ()
```

wizard.views.threddsbrowser

Module Contents

```
class wizard.views.threddsbrowser.Schema
class wizard.views.threddsbrowser.ThreddsBrowser (request)

    breadcrumbs ()
```

```
    schema ()
    appstruct ()
    next_success (appstruct)
    custom_view ()
    view ()
class wizard.views.threddsbrowser.Grid (request, *args, **kwargs)

    name_td (col_num, i, item)
```

wizard.views.threddsservice

Module Contents

```
wizard.views.threddsservice.includeme (config)
wizard.views.threddsservice.deferred_widget (node, kw)
class wizard.views.threddsservice.Schema
class wizard.views.threddsservice.ThreddsService (request)

    breadcrumbs ()
    schema ()
    success (appstruct)
    next_success (appstruct)
    view ()
```

wizard.views.wps

Module Contents

```
wizard.views.wps.includeme (config)
class wizard.views.wps.ChooseWPSSchema

    deferred_validator (kw)
    deferred_widget (kw)
class wizard.views.wps.ChooseWPS (request)

    breadcrumbs ()
    schema ()
    next_success (appstruct)
    view ()
```

`wizard.views.wpsprocess`

Module Contents

```
wizard.views.wpsprocess.includeme(config)
wizard.views.wpsprocess.count_literal_inputs(wps, identifier)
class wizard.views.wpsprocess.Schema

    deferred_validator(kw)
    deferred_widget(kw)
class wizard.views.wpsprocess.ChooseWPSPProcess(request)

    breadcrumbs()
    schema()
    next_success(appstruct)
    view()
    custom_view()
```

Package Contents

```
class wizard.views.WizardFavorite(request, session)
    Stores wizard state in session with a name (favorite). TODO: implement as a dict?

    names()
    get(name)
    set(name, state)
    clear()
class wizard.views.WizardState(session, initial_step="wizard", final_step="wizard_done")

    load(state)
    dump()
    current_step()
    is_first()
    is_last()
    next(step)
    previous()
    get(key, default=None)
    set(key, value)
    clear()
```

```
class wizard.views.Wizard(request, name, title, description=None)
```

```
    buttons()
    prev_ok()
    next_ok()
    use_ajax()
    ajax_options()
    success(appstruct)
    appstruct()
    schema()
    previous_success(appstruct)
    previous_failure(validation_failure)
    next_success(appstruct)
    next_failure(validation_failure)
    generate_form(formid="deform")
    process_form(form, action)
    previous()
    next(step, query=None)
    cancel()
    custom_view()
    breadcrumbs()
    resources()
    view()
```

6.22.2 Package Contents

```
wizard.includeme(config)
```

6.23 cart

6.23.1 Submodules

```
cart.actions
```

Module Contents

```
class cart.actions.CartActions(context, request)
```

Actions related to cart.

```
    list_cart()
```

```

add_to_cart ()
remove_from_cart ()
clear_cart ()
remove_item ()

```

`cart.cart`

Module Contents

```
class cart.cart.CartItem (url, title=None, abstract=None, mime_type=None, dataset=None)
```

```

title ()
abstract ()
filename ()
is_service ()
is_opendap ()
is_thredds_catalog ()
to_json ()

```

```
class cart.cart.Cart (request)
```

```

add_item (url, title=None, abstract=None, mime_type=None)
    Add cart item.

remove_item (url)
    Remove cart item with given url.

count ()
    Returns: number of cart items.

has_items ()
    Returns: True if cart items available, otherwise False.

clear ()
    Removes all items of cart and updates session.

save ()
    Store cart items in session.

load ()
    Load cart items from session.

to_json ()
    Returns: json representation of all cart items.

```

`cart.views`

Module Contents

`class cart.views.Cart(request)`

`view()`

6.23.2 Package Contents

`cart.includeme(config)`

6.24 tests

6.24.1 Submodules

`tests.test_cart`

Module Contents

`tests.test_cart.test_cart()`

`tests.test_cart.test_clear_cart()`

`class tests.test_cart.CartTests`

`setUp()`

`tearDown()`

`test_cart_from_request()`

`tests.test_catalog`

Module Contents

`tests.test_catalog.test_doc2record()`

`tests.test_esgf_metadata`

Module Contents

`tests.test_esgf_metadata.test_convert_constraints()`

`tests.test_esgf_search`

Module Contents

```
tests.test_esgf_search.test_build_constraints_dict()
tests.test_esgf_search.test_date_from_filename()
tests.test_esgf_search.test_temporal_filter()
tests.test_esgf_search.test_variable_filter()
class tests.test_esgf_search.ESGFSearchTests

    setUp()
    tearDown()
    test_query_params()
    test_params()
    test_search_datasets()
    test_search_items()
```

`tests.test_form`

Module Contents

```
tests.test_form.invalid_exc(func, *arg, **kw)
class tests.test_form.TestBBoxValidator

    test_default()
    test_minx()
    test_miny()
    test_maxx()
    test_maxy()
class tests.test_form.TestURLValidator

    test_default()
    test_file_scheme()
    test_invalid_path()
    test_invalid_relative_path()
class tests.test_form.TestTextValidator

    test_default()
    test_empty()
    test_restricted_chars()
```

`tests.test_settings`

Module Contents

```
class tests.test_settings.UserSettingsTests
```

```
    setUp()
```

```
    tearDown()
```

```
    test_user_view()
```

```
class tests.test_settings.UserSettingsFunctionalTests
```

```
    setUp()
```

```
    test_user_view()
```

`tests.test_utils`

Module Contents

```
tests.test_utils.test_headline()
```

```
tests.test_utils.test_time_ago_in_words()
```

```
tests.test_utils.test_make_tags()
```

```
tests.test_utils.test_format_tags()
```

`tests.test_wizard`

Module Contents

```
tests.test_wizard.test_convert_states_to_nodes()
```

`tests.test_wps`

Module Contents

```
tests.test_wps.test_check_status()
```


6.25 supervisor

6.25.1 Subpackages

`supervisor.views`

Submodules

`supervisor.views.supervisor`

Module Contents

```
class supervisor.views.supervisor.Supervisor(request)
```

```
    supervisor_process()
```

```
    view()
```

```
class supervisor.views.supervisor.Grid(request, *args, **kwargs)
```

```
    state_td(col_num, i, item)
```

```
    buttongroup_td(col_num, i, item)
```

`supervisor.views.supervisor_log`

Module Contents

```
class supervisor.views.supervisor_log.SupervisorLog(request)
```

```
    view()
```

6.25.2 Package Contents

`supervisor.includeme` (*config*)

6.26 dashboard

6.26.1 Submodules

`dashboard.panels`

Module Contents

```
dashboard.panels.dashboard_overview(context, request)
```

```
dashboard.panels.dashboard_people(context, request)
```

`dashboard.panels.dashboard_jobs` (*context, request*)

`dashboard.views`

Module Contents

class `dashboard.views.Dashboard` (*request*)

`view()`

6.26.2 Package Contents

`dashboard.includeme` (*config*)

6.27 settings

6.27.1 Subpackages

`settings.views`

Submodules

`settings.views.ldap_config`

Module Contents

class `settings.views.ldap_config.Ldap` (*request*)

`breadcrumbs()`

`view()`

`settings.views.overview`

Module Contents

class `settings.views.overview.Overview` (*request*)

`view()`

`settings.views.processes`

Module Contents

class `settings.views.processes.Processes` (*request*)

```
breadcrumbs()  
generate_form()  
process_form(form)  
appstruct()  
view()
```

`settings.views.solr`

Module Contents

```
class settings.views.solr.SolrSettings(request)
```

```
    breadcrumbs()  
    index_service()  
    clear_index()  
    view()
```

6.27.2 Submodules

`settings.schema`

Module Contents

```
settings.schema.deferred_processes_widget(node, kw)  
class settings.schema.ProcessesSchema  
class settings.schema.AuthProtocolSchema  
class settings.schema.LdapSchema
```

6.27.3 Package Contents

```
settings.includeme(config)
```

6.28 account

6.28.1 Submodules

`account.base`

Module Contents

```
account.base.forbidden(request)
```

```
class account.base.Account (request)
```

```
    schema ()
```

```
    generate_form ()
```

```
    process_form (form)
```

```
    _handle_appstruct (appstruct)
```

```
    send_notification (email, subject, message)
```

```
        Sends email notification to admins.
```

```
        Sends email with the pyramid_mailer module. For configuration look at documentation http://pythonhosted.org/pyramid\_mailer/
```

```
    add_user (login_id, email=None)
```

```
    login ()
```

```
    login_success (login_id, email=None, name=None, openid=None, local=False)
```

```
    login_failure (message=None)
```

```
    logout ()
```

```
    register ()
```

```
    authomatic_login ()
```

account.esgf

Module Contents

```
class account.esgf.ESGFSchema
```

```
class account.esgf.ESGFAccount
```

```
    schema ()
```

```
    _handle_appstruct (appstruct)
```

```
    esgf_login ()
```

account.ldap

Module Contents

```
class account.ldap.LDAPSchema
```

```
class account.ldap.LDAPAccount
```

```
    schema ()
```

```
    _handle_appstruct (appstruct)
```

```
        Handle LDAP login.
```

```
    ldap_login ()
```

```
init_ldap()  
    Lazy LDAP connector construction
```

`account.local`

Module Contents

```
class account.local.LocalSchema  
class account.local.LocalAccount
```

```
    schema()  
    _handle_appstruct(appstruct)  
    sign_in()
```

6.28.2 Package Contents

`account.includeme(config)`

6.29 people

6.29.1 Subpackages

`people.views`

Submodules

`people.views.actions`

Module Contents

```
class people.views.actions.Actions(request)  
  
    update_esgf_certs()  
    forget_esgf_certs()  
    generate_twitcher_token()  
    generate_esgf_slcs_token()  
        Update ESGF slcs token.  
    forget_esgf_slcs_token()  
        Forget ESGF slcs token.  
    esgf_oauth_callback()  
        Convert an authorisation grant into an access token.  
    delete_user()
```

```
people.views.actions.includeme(config)
```

Pyramid includeme hook. :param config: app config :type config: pyramid.config.Configurator

people.views.list

Module Contents

```
class people.views.list.People(request)
```

```
    view()
```

```
class people.views.list.PeopleGrid(request, *args, **kwargs)
```

```
    group_td(col_num, i, item)
```

```
    buttongroup_td(col_num, i, item)
```

people.views.profile

Module Contents

```
class people.views.profile.Profile(request)
```

```
    panel_title()
```

```
    appstruct()
```

```
    readonly()
```

```
    schema()
```

```
    generate_form()
```

```
    generate_buttons()
```

```
    process_form(form)
```

```
    view()
```

6.29.2 Submodules

people.schema

Module Contents

```
class people.schema.ProfileSchema
```

```
class people.schema.GroupSchema
```

```
class people.schema.TwitcherSchema
```

```
class people.schema.ESGFSLCSTokenSchema
```

```
class people.schema.ESGFCredentialsSchema
```

6.29.3 Package Contents

```
people.includeme(config)
```

6.30 map

6.30.1 Package Contents

```
class map.Map(request)
```

```
    view()
```

```
map.includeme(config)
```

6.31 solrsearch

6.31.1 Subpackages

```
solrsearch.views
```

Submodules

```
solrsearch.views.actions
```

Module Contents

```
class solrsearch.views.actions.Actions(request)
```

```
solrsearch.views.solrsearch
```

Module Contents

```
class solrsearch.views.solrsearch.SolrSearch(request)
```

```
    view()
```

6.31.2 Submodules

```
solrsearch.panels
```

Module Contents

```
solrsearch.panels.query_path(request)
```

```
solrsearch.panels.solrsearch_script(context, request)
```

```
solrsearch.panels.solrsearch(context, request)
```

`solrsearch.schema`

Module Contents

`class solrsearch.schema.SolrSearchSchema`

`solrsearch.search`

Module Contents

`solrsearch.search.solr_search(url, query, page, category, source, tag)`

6.31.3 Package Contents

`solrsearch.includeme(config)`

6.32 geoform

6.32.1 Submodules

`geoform.form`

Module Contents

`class geoform.form.BBoxValidator`

Bounding-Box validator which succeeds if the bbox value has the format minx,miny,maxx,maxy and values are in range (-180 <= x <=180, -90 <= y <=90).

`class geoform.form.URLValidator(allowed_schemes=None)`

URL validator which can configured with allowed URL schemes.

`class geoform.form.TextValidator(restricted_chars=None)`

`class geoform.form.FileUploadValidator(storage, max_size)`

Runs all validators for file upload checks.

`class geoform.form.FileFormatAllowedValidator(storage)`

File format extension is allowed.

https://pythonhosted.org/pyramid_storage/

`class geoform.form.FileSizeLimitValidator(storage, max_size=2)`

File size limit validator.

You can configure the maximum size by setting the max_size option to the maximum number of megabytes that you want to allow.

`geoform.widget`

Module Contents

class `geoform.widget.ResourceWidget`

Renders an WPS ComplexType input widget with a cart and upload button.

It is based on `deform.widget.TextInputWidget`.

serialize (*field, cstruct, **kw*)

deserialize (*field, pstruct*)

class `geoform.widget.BBoxWidget`

Renders a BoundingBox Widget.

Attributes/Arguments `template`

The template name used to render the input widget. Default: `bbox`.

readonly_template The template name used to render the widget in read-only mode. Default: `readonly/bbox`.

serialize (*field, cstruct, **kw*)

deserialize (*field, pstruct*)

class `geoform.widget.TagsWidget`

serialize (*field, cstruct, **kw*)

deserialize (*field, pstruct*)

6.33 solr

6.33.1 Submodules

`solr.panels`

Module Contents

class `solr.panels.Schema`

class `solr.panels.SolrPanel` (*context, request*)

class `solr.panels.SolrIndexPanel`

panel ()

class `solr.panels.SolrParamsPanel`

appstruct ()

panel ()

6.33.2 Package Contents

```
solr.includeme(config)
```

6.34 esgf

6.34.1 Subpackages

`esgf.views`

Submodules

`esgf.views.esgflogon`

Module Contents

```
class esgf.views.esgflogon.ESGFLogon(request)
```

```
    appstruct()
    generate_form()
    process_form(form)
    check_logon()
    loading()
    callback()
    view()
```

`esgf.views.esgfsearch`

Module Contents

```
class esgf.views.esgfsearch.ESGFSearchActions(request)
```

```
    search_datasets()
    search_items()
```

6.34.2 Submodules

`esgf.logon`

Module Contents

```
esgf.logon.save_credentials(registry, userid, file=None, filename=None)
```

`esgf.logon.logon` (*username=None, password=None, hostname=None, interactive=False, out-dir=None*)
 Logon to MyProxy and fetch proxy certificate.

`esgf.logon.cert_infos` (*filename*)

`esgf.metadata`

Module Contents

`esgf.metadata.process_constraints` (*process*)

`esgf.metadata.convert_constraints` (*url*)
 converts esgf search query to constraints parameter. TODO: constraints parameter should have the same structure as the esgf query.

`esgf.schema`

Module Contents

`class` `esgf.schema.ESGFLogonSchema`

`esgf.schema.esgfsearch_validator` (*node, value*)

`class` `esgf.schema.ESGFSearchSchema`

`esgf.search`

Module Contents

`esgf.search.date_from_filename` (*filename*)

Example cordex: `tas_EUR-44i_ECMWF-ERAINT_evaluation_r1i1p1_HMS-ALADIN52_v1_mon_200101-200812.nc`

`esgf.search.variable_filter` (*constraints, variables*)
 return True if variable fulfills constraints

`esgf.search.temporal_filter` (*filename, start=None, end=None*)
 return True if file is in timerange start/end

`esgf.search.query_params_from_appstruct` (*appstruct, defaults*)

`esgf.search.build_constraint_dict` (*constraints*)

`class` `esgf.search.ESGFSearch` (*request, url=None*)

`_parse_params` ()
 parse search params.

`query_params` ()
 search params as string used for query.

`params` ()
 search params as object.

`search_items` ()
 search files and aggregations with download url and opendap url.

```
_run_search_items (dataset_id, search_type)  
search_datasets ()  
    search datasets according to search parameters.
```

esgf.slcsclient

Module Contents

```
esgf.slcsclient.refresh_token (registry, token, userid)  
esgf.slcsclient.save_token (registry, token, userid)  
    Store the token in the database.  
class esgf.slcsclient.ESGFSLCSClient (request)  
  
    authorize ()  
        Redirect the user to the ESGF SLCS Server for authorisation.  
    callback ()  
        Convert an authorisation grant into an access token.  
    refresh_token ()  
    get_token ()  
    save_token (token)  
        Store the token in the database.  
    delete_token ()  
        Remove token from database.  
    get_certificate ()  
        Generates a new private key and certificate request, submits the request to be signed by the SLCS CA and  
        prints the resulting key/certificate pair.  
        Uses automatic refreshing of tokens if they have expired.
```

esgf.validator

Module Contents

```
esgf.validator.cert_ok (request, valid_hours=3)
```

6.34.3 Package Contents

```
esgf.includeme (config)
```

6.35 monitor

6.35.1 Subpackages

`monitor.panels`

Submodules

`monitor.panels.inputs`

Module Contents

`monitor.panels.inputs.collect_inputs` (*status_location=None, response=None*)

`monitor.panels.inputs.process_inputs` (*request, job_id*)

class `monitor.panels.inputs.Inputs` (*context, request*)

`panel()`

`monitor.panels.outputs`

Module Contents

`monitor.panels.outputs.collect_outputs` (*status_location=None, response=None*)

`monitor.panels.outputs.process_outputs` (*request, job_id*)

class `monitor.panels.outputs.Outputs` (*context, request*)

`panel()`

Package Contents

`monitor.panels.job_details` (*request, job_id*)

`monitor.panels.details` (*context, request*)

`monitor.panels.log` (*context, request*)

`monitor.panels.xml` (*context, request*)

`monitor.views`

Submodules

`monitor.views.actions`

Module Contents

class `monitor.views.actions.NodeActions` (*context, request*)

Actions related to job monitor.

`_selected_children()`

Get the selected children of the given context.

Result List with select children.

Return type *list*

`restart_job()`

`delete_job()`

`delete_jobs()`

Delete selected jobs.

`delete_all_jobs()`

`make_public()`

Make selected jobs public.

`make_private()`

Make selected jobs private.

`set_favorite()`

Set selected jobs as favorite.

`unset_favorite()`

Unset selected jobs as favorite.

`edit_job()`

`active_jobs()`

`monitor_buttons` (*context, request*)

Build the action buttons for the monitor view based on the current state and the permissions of the user.

Result List of ActionButtons.

Return type *list*

`download_wpsoutputs` (*request*)

`includeme` (*config*)

Pyramid includeme hook.

Parameters **`config`** (`pyramid.config.Configurator`) – app config

`monitor.views.details`

Module Contents

`class monitor.views.details.Details(request)`

`view()`

`monitor.views.list`

Module Contents

`class monitor.views.list.CaptionSchema`

This is the form schema to add and edit form for job captions.

`class monitor.views.list.LabelsSchema`

This is the form schema to add and edit form for job tags/labels.

`class monitor.views.list.JobList(request)`

`filter_jobs(page=0, limit=10, tag=None, access=None, status=None, sort="created")`

`generate_caption_form(formid="deform_caption")`

This helper code generates the form that will be used to add and edit job captions based on the schema of the form.

`process_caption_form(form)`

`generate_labels_form(formid="deform_tags")`

This helper code generates the form that will be used to add and edit job tags/labels based on the schema of the form.

`process_labels_form(form)`

`view()`

`class monitor.views.list.JobsGrid(request, *args, **kwargs)`

`status_td(col_num, i, item)`

`duration_td(col_num, i, item)`

`caption_td(col_num, i, item)`

`labels_td(col_num, i, item)`

`buttongroup_td(col_num, i, item)`

`monitor.views.list_json`

Module Contents

`class monitor.views.list_json.JobListJson(request)`

`view()`

`monitor.views.status`

Module Contents

`class monitor.views.status.JobStatus(request)`

`view()`

Package Contents

`monitor.views.notify_job_started(event)`

`monitor.views.notify_job_finished(event)`

6.35.2 Submodules

`monitor.utils`

Module Contents

`monitor.utils.escape_output(output)`

`monitor.utils.output_details(request, output)`

6.35.3 Package Contents

`monitor.includeme(config)`

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

—
__init__, 66
_compat, 63

a

account, 87
account.base, 87
account.esgf, 88
account.ldap, 88
account.local, 89

c

cart, 80
cart.actions, 80
cart.cart, 81
cart.views, 82
catalog, 64

d

dashboard, 85
dashboard.panels, 85
dashboard.views, 86
db, 63

e

esgf, 94
esgf.logon, 94
esgf.metadata, 95
esgf.schema, 95
esgf.search, 95
esgf.slcsclient, 96
esgf.validator, 96
esgf.views, 94
esgf.views.esgflogon, 94
esgf.views.esgfsearch, 94
events, 63
exceptions, 66

g

geoform, 92

geoform.form, 92
geoform.widget, 93
grid, 68

l

layouts, 64
ldap, 63

m

map, 91
monitor, 97
monitor.panels, 97
monitor.panels.inputs, 97
monitor.panels.outputs, 97
monitor.utils, 100
monitor.views, 98
monitor.views.actions, 98
monitor.views.details, 99
monitor.views.list, 99
monitor.views.list_json, 99
monitor.views.status, 100

p

panels, 66
patch, 66
people, 89
people.schema, 90
people.views, 89
people.views.actions, 89
people.views.list, 90
people.views.profile, 90
processes, 70
processes.views, 70
processes.views.actions, 70
processes.views.execute, 71
processes.views.execute_json, 71
processes.views.list, 71
processes.views.list_json, 71
processes.views.overview, 72
processes.views.overview_json, 72

providers, [72](#)
providers.esgfoopenid, [72](#)

S

security, [65](#)
services, [73](#)
services.views, [73](#)
services.views.actions, [73](#)
services.views.registerservice, [74](#)
services.views.services, [74](#)
settings, [86](#)
settings.schema, [87](#)
settings.views, [86](#)
settings.views.ldap_config, [86](#)
settings.views.overview, [86](#)
settings.views.processes, [86](#)
settings.views.solr, [87](#)
solr, [93](#)
solr.panels, [93](#)
solrsearch, [91](#)
solrsearch.panels, [91](#)
solrsearch.schema, [92](#)
solrsearch.search, [92](#)
solrsearch.views, [91](#)
solrsearch.views.actions, [91](#)
solrsearch.views.solrsearch, [91](#)
storage, [73](#)
storage.views, [73](#)
supervisor, [85](#)
supervisor.views, [85](#)
supervisor.views.supervisor, [85](#)
supervisor.views.supervisor_log, [85](#)

t

tasks, [69](#)
tasks.esgflogon, [69](#)
tasks.execute, [69](#)
tasks.solr, [70](#)
tasks.utils, [70](#)
tasks.workflow, [70](#)
tests, [82](#)
tests.test_cart, [82](#)
tests.test_catalog, [82](#)
tests.test_esgf_metadata, [82](#)
tests.test_esgf_search, [83](#)
tests.test_form, [83](#)
tests.test_settings, [84](#)
tests.test_utils, [84](#)
tests.test_wizard, [84](#)
tests.test_wps, [84](#)
twitcherclient, [68](#)

U

utils, [67](#)

V

views, [69](#)

W

wizard, [75](#)
wizard.views, [75](#)
wizard.views.complexinputs, [75](#)
wizard.views.done, [75](#)
wizard.views.esgfsearch, [76](#)
wizard.views.literalinputs, [76](#)
wizard.views.solrsearch, [76](#)
wizard.views.source, [77](#)
wizard.views.start, [77](#)
wizard.views.threddsbrowser, [77](#)
wizard.views.threddsservice, [78](#)
wizard.views.wps, [78](#)
wizard.views.wpsprocess, [79](#)
wps, [66](#)

Symbols

__init__ (module), 66
 _compat (module), 63
 _fetch_thredds_metadata() (in module catalog), 64
 _fetch_wps_metadata() (in module catalog), 64
 _handle_appstruct() (account.base.Account method), 88
 _handle_appstruct() (account.esgf.ESGFAccount method), 88
 _handle_appstruct() (account.ldap.LDAPAccount method), 88
 _handle_appstruct() (account.local.LocalAccount method), 89
 _parse_params() (esgf.search.ESGFSearch method), 95
 _run_search_items() (esgf.search.ESGFSearch method), 96
 _selected_children() (monitor.views.actions.NodeActions method), 98
 _url_node_default() (wps.WPSSchema method), 67
 _urlopen() (providers.esgfopenid.MyFetcher method), 72

A

abstract() (cart.cart.CartItem method), 81
 Account (class in account.base), 87
 account (module), 87
 account.base (module), 87
 account.esgf (module), 88
 account.ldap (module), 88
 account.local (module), 89
 ActionButton (class in utils), 67
 Actions (class in people.views.actions), 89
 Actions (class in solrsearch.views.actions), 91
 active_jobs() (monitor.views.actions.NodeActions method), 98
 add_async_check() (wps.WPSSchema method), 67
 add_breadcrumb() (layouts.PageLayout method), 64
 add_global() (in module views), 69
 add_heading() (layouts.PageLayout method), 64
 add_item() (cart.cart.Cart method), 81
 add_job() (in module tasks.utils), 70

add_nodes() (wps.WPSSchema method), 67
 add_to_cart() (cart.actions.CartActions method), 80
 add_user() (account.base.Account method), 88
 after_bind() (wizard.views.source.SourceSchemaNode method), 77
 ajax_options() (wizard.views.Wizard method), 80
 appstruct() (esgf.views.esgflogon.ESGFLogon method), 94
 appstruct() (people.views.profile.Profile method), 90
 appstruct() (processes.views.execute.ExecuteProcess method), 71
 appstruct() (settings.views.processes.Processes method), 87
 appstruct() (solr.panels.SolrParamsPanel method), 93
 appstruct() (wizard.views.esgfsearch.ESGFSearchView method), 76
 appstruct() (wizard.views.solrsearch.SolrSearch method), 76
 appstruct() (wizard.views.start.Start method), 77
 appstruct() (wizard.views.threddsbrowser.ThreddsBrowser method), 78
 appstruct() (wizard.views.Wizard method), 80
 appstruct_to_inputs() (in module wps), 67
 authenticated_userid() (security.MyAuthenticationPolicy method), 66
 authomatic() (in module security), 65
 authomatic_config() (in module security), 65
 authomatic_login() (account.base.Account method), 88
 authorize() (esgf.slcsclient.ESGFSLCSClient method), 96
 AuthProtocolSchema (class in settings.schema), 87

B

bbox_data() (wps.WPSSchema method), 67
 BBoxValidator (class in geoform.form), 92
 BBoxWidget (class in geoform.widget), 93
 bind() (wps.WPSSchema method), 67
 breadcrumbs() (in module panels), 66
 breadcrumbs() (processes.views.execute.ExecuteProcess method), 71

- `breadcrumbs()` (`services.views.registerservice.RegisterService` method), 74
 - `breadcrumbs()` (`services.views.services.Services` method), 74
 - `breadcrumbs()` (`settings.views.ldap_config.Ldap` method), 86
 - `breadcrumbs()` (`settings.views.processes.Processes` method), 86
 - `breadcrumbs()` (`settings.views.solr.SolrSettings` method), 87
 - `breadcrumbs()` (`views.MyView` method), 69
 - `breadcrumbs()` (`wizard.views.complexinputs.ComplexInputs` method), 75
 - `breadcrumbs()` (`wizard.views.done.Done` method), 75
 - `breadcrumbs()` (`wizard.views.esgfsearch.ESGFSearchView` method), 76
 - `breadcrumbs()` (`wizard.views.literalinputs.LiteralInputs` method), 76
 - `breadcrumbs()` (`wizard.views.solrsearch.SolrSearch` method), 76
 - `breadcrumbs()` (`wizard.views.source.ChooseSource` method), 77
 - `breadcrumbs()` (`wizard.views.threddsbrowser.ThreddsBrowser` method), 77
 - `breadcrumbs()` (`wizard.views.threddsservice.ThreddsService` method), 78
 - `breadcrumbs()` (`wizard.views.Wizard` method), 80
 - `breadcrumbs()` (`wizard.views.wps.ChooseWPS` method), 78
 - `breadcrumbs()` (`wizard.views.wpsprocess.ChooseWPSProcess` method), 79
 - `build_constraint_dict()` (in module `esgf.search`), 95
 - `build_url()` (in module `utils`), 67
 - `buttongroup_td()` (`monitor.views.list.JobsGrid` method), 99
 - `buttongroup_td()` (`people.views.list.PeopleGrid` method), 90
 - `buttongroup_td()` (`supervisor.views.supervisor.Grid` method), 85
 - `buttons()` (`wizard.views.Wizard` method), 80
- ## C
- `callback()` (`esgf.slcsclient.ESGFSLCSClient` method), 96
 - `callback()` (`esgf.views.esgflogon.ESGFLogon` method), 94
 - `cancel()` (`wizard.views.Wizard` method), 80
 - `caption_td()` (`monitor.views.list.JobsGrid` method), 99
 - `CaptionSchema` (class in `monitor.views.list`), 99
 - `Cart` (class in `cart.cart`), 81
 - `Cart` (class in `cart.views`), 82
 - `cart` (module), 80
 - `cart.actions` (module), 80
 - `cart.cart` (module), 81
 - `cart.views` (module), 82
 - `CartActions` (class in `cart.actions`), 80
 - `CartItem` (class in `cart.cart`), 81
 - `CartTests` (class in `tests.test_cart`), 82
 - `Catalog` (class in `catalog`), 64
 - `catalog` (module), 64
 - `catalog_factory()` (in module `catalog`), 64
 - `CatalogService` (class in `catalog`), 65
 - `cert_infos()` (in module `esgf.logon`), 95
 - `cert_ok()` (in module `esgf.validator`), 96
 - `check_csrf_token()` (in module `security`), 65
 - `check_logon()` (`esgf.views.esgflogon.ESGFLogon` method), 94
 - `check_status()` (in module `wps`), 66
 - `checkbox_column_format()` (`grid.CustomGrid` method), 68
 - `ChooseSource` (class in `wizard.views.source`), 77
 - `ChooseWPS` (class in `wizard.views.wps`), 78
 - `ChooseWPSProcess` (class in `wizard.views.wpsprocess`), 79
 - `ChooseWPSSchema` (class in `wizard.views.wps`), 78
 - `clear()` (`cart.cart.Cart` method), 81
 - `clear()` (`wizard.views.WizardFavorite` method), 79
 - `clear()` (`wizard.views.WizardState` method), 79
 - `clear_cart()` (`cart.actions.CartActions` method), 81
 - `clear_index()` (in module `tasks.solr`), 70
 - `clear_index()` (`settings.views.solr.SolrSettings` method), 87
 - `clear_services()` (`catalog.Catalog` method), 64
 - `clear_services()` (`catalog.MongodbCatalog` method), 65
 - `clear_services()` (`services.views.actions.ServiceActions` method), 74
 - `clone()` (`wps.WPSSchema` method), 67
 - `colander_literal_type()` (`wps.WPSSchema` method), 67
 - `colander_literal_widget()` (`wps.WPSSchema` method), 67
 - `collect_inputs()` (in module `monitor.panels.inputs`), 97
 - `collect_outputs()` (in module `monitor.panels.outputs`), 97
 - `combine_chunks()` (in module `storage.views`), 73
 - `complex_data()` (`wps.WPSSchema` method), 67
 - `ComplexInputs` (class in `wizard.views.complexinputs`), 75
 - `convert_constraints()` (in module `esgf.metadata`), 95
 - `converted_settings()` (`events.SettingsChanged` method), 64
 - `count()` (`cart.cart.Cart` method), 81
 - `count_literal_inputs()` (in module `wizard.views.wpsprocess`), 79
 - `current_step()` (`wizard.views.WizardState` method), 79
 - `custom_view()` (`wizard.views.complexinputs.ComplexInputs` method), 75
 - `custom_view()` (`wizard.views.esgfsearch.ESGFSearchView` method), 76
 - `custom_view()` (`wizard.views.literalinputs.LiteralInputs` method), 76
 - `custom_view()` (`wizard.views.threddsbrowser.ThreddsBrowser` method), 78

custom_view() (wizard.views.Wizard method), 80

custom_view() (wizard.views.wpsprocess.ChooseWPSProcess method), 79

CustomGrid (class in grid), 68

D

Dashboard (class in dashboard.views), 86

dashboard (module), 85

dashboard.panels (module), 85

dashboard.views (module), 86

dashboard_jobs() (in module dashboard.panels), 85

dashboard_overview() (in module dashboard.panels), 85

dashboard_people() (in module dashboard.panels), 85

date_from_filename() (in module esgf.search), 95

db (module), 63

default_header_column_format() (grid.CustomGrid method), 68

default_header_ordered_column_format() (grid.CustomGrid method), 68

deferred_favorite_widget() (wizard.views.start.FavoriteSchema method), 77

deferred_processes_widget() (in module settings.schema), 87

deferred_validator() (wizard.views.wps.ChooseWPSSchema method), 78

deferred_validator() (wizard.views.wpsprocess.Schema method), 79

deferred_widget() (in module wizard.views.complexinputs), 75

deferred_widget() (in module wizard.views.threddsservice), 78

deferred_widget() (wizard.views.wps.ChooseWPSSchema method), 78

deferred_widget() (wizard.views.wpsprocess.Schema method), 79

delete() (in module storage.views), 73

delete_all_jobs() (monitor.views.actions.NodeActions method), 98

delete_job() (monitor.views.actions.NodeActions method), 98

delete_jobs() (monitor.views.actions.NodeActions method), 98

delete_record() (catalog.Catalog method), 64

delete_record() (catalog.CatalogService method), 65

delete_record() (catalog.MongodbCatalog method), 65

delete_token() (esgf.slcsclient.ESGFSLCSCClient method), 96

delete_user() (people.views.actions.Actions method), 89

deserialize() (geoform.widget.BBoxWidget method), 93

deserialize() (geoform.widget.ResourceWidget method), 93

deserialize() (geoform.widget.TagsWidget method), 93

Details (class in monitor.views.details), 99

details() (in module monitor.panels), 97

details_view() (services.views.services.Services method), 74

doc2record() (in module catalog), 65

Done (class in wizard.views.done), 75

DoneSchema (class in wizard.views.done), 75

download() (in module storage.views), 73

download_wpsoutputs() (in module monitor.views.actions), 98

dump() (wizard.views.WizardState method), 79

dump_json() (in module tasks.utils), 70

duration_td() (monitor.views.list.JobsGrid method), 99

E

edit_job() (monitor.views.actions.NodeActions method), 98

escape_output() (in module monitor.utils), 100

esgf (module), 94

esgf.logon (module), 94

esgf.metadata (module), 95

esgf.schema (module), 95

esgf.search (module), 95

esgf.slcsclient (module), 96

esgf.validator (module), 96

esgf.views (module), 94

esgf.views.esgflgon (module), 94

esgf.views.esgfsearch (module), 94

esgf_login() (account.esgf.ESGFAccount method), 88

esgf_logon() (in module tasks.esgflgon), 69

esgf_oauth_callback() (people.views.actions.Actions method), 89

ESGFAccount (class in account.esgf), 88

ESGFCredentialsSchema (class in people.schema), 90

ESGFLgon (class in esgf.views.esgflgon), 94

ESGFLgonSchema (class in esgf.schema), 95

ESGFOpenID (class in providers.esgfopenid), 72

ESGFSchema (class in account.esgf), 88

ESGFSearch (class in esgf.search), 95

esgfsearch_validator() (in module esgf.schema), 95

ESGFSearchActions (class in esgf.views.esgfsearch), 94

ESGFSearchSchema (class in esgf.schema), 95

ESGFSearchTests (class in tests.test_esgf_search), 83

ESGFSearchView (class in wizard.views.esgfsearch), 76

ESGFSLCSCClient (class in esgf.slcsclient), 96

ESGFSLCSTokenSchema (class in people.schema), 90

events (module), 63

exceptions (module), 66

execute() (processes.views.execute.ExecuteProcess method), 71

execute_process() (in module tasks.execute), 69

execute_workflow() (in module tasks.workflow), 70

ExecuteProcess (class in processes.views.execute), 71

ExecuteProcessJson (class in processes.views.execute_json), 71

F

favicon_view() (in module views), 69

FavoriteSchema (class in wizard.views.start), 77

FileFormatAllowedValidator (class in geoform.form), 92

filename() (cart.cart.CartItem method), 81

FileSizeLimitValidator (class in geoform.form), 92

FileUploadValidator (class in geoform.form), 92

filter_jobs() (monitor.views.list.JobList method), 99

footer() (in module panels), 66

forbidden() (in module account.base), 87

forget_esgf_certs() (people.views.actions.Actions method), 89

forget_esgf_slcs_token() (people.views.actions.Actions method), 89

format_tags() (in module utils), 67

G

generate_access_token() (in module twitcherclient), 69

generate_buttons() (people.views.profile.Profile method), 90

generate_caption_form() (monitor.views.list.JobList method), 99

generate_esgf_slcs_token() (people.views.actions.Actions method), 89

generate_form() (account.base.Account method), 88

generate_form() (esgf.views.esgflogon.ESGFLogon method), 94

generate_form() (people.views.profile.Profile method), 90

generate_form() (processes.views.execute.ExecuteProcess method), 71

generate_form() (services.views.registerservice.RegisterService method), 74

generate_form() (settings.views.processes.Processes method), 87

generate_form() (wizard.views.Wizard method), 80

generate_header_link() (grid.CustomGrid method), 68

generate_labels_form() (monitor.views.list.JobList method), 99

generate_twitcher_token() (people.views.actions.Actions method), 89

geoform (module), 92

geoform.form (module), 92

geoform.widget (module), 93

get() (wizard.views.WizardFavorite method), 79

get() (wizard.views.WizardState method), 79

get_access_token() (in module tasks.utils), 70

get_certificate() (esgf.slcsclient.ESGFSLCSCClient method), 96

get_record_by_id() (catalog.Catalog method), 64

get_record_by_id() (catalog.CatalogService method), 65

get_record_by_id() (catalog.MongodbCatalog method), 65

get_service_by_name() (catalog.Catalog method), 64

get_service_by_url() (catalog.Catalog method), 64

get_service_name() (catalog.Catalog method), 64

get_services() (catalog.Catalog method), 64

get_services() (catalog.CatalogService method), 65

get_services() (catalog.MongodbCatalog method), 65

get_token() (esgf.slcsclient.ESGFSLCSCClient method), 96

get_user() (in module security), 66

get_value() (in module grid), 68

Grid (class in supervisor.views.supervisor), 85

Grid (class in wizard.views.threddsbrowser), 78

grid (module), 68

group_td() (people.views.list.PeopleGrid method), 90

groupfinder() (in module security), 65

GroupSchema (class in people.schema), 90

H

handle_delete() (in module storage.views), 73

handle_upload() (in module storage.views), 73

harvest() (catalog.Catalog method), 64

harvest() (catalog.CatalogService method), 65

harvest() (catalog.MongodbCatalog method), 65

has_execute_permission() (in module security), 65

has_items() (cart.cart.Cart method), 81

headings() (in module panels), 66

headline() (in module utils), 67

Home (class in views), 69

I

includeme() (in module account), 89

includeme() (in module cart), 82

includeme() (in module catalog), 64

includeme() (in module dashboard), 86

includeme() (in module db), 63

includeme() (in module esgf), 96

includeme() (in module ldap), 63

includeme() (in module map), 91

includeme() (in module monitor), 100

includeme() (in module monitor.views.actions), 98

includeme() (in module people), 91

includeme() (in module people.views.actions), 89

includeme() (in module processes), 72

includeme() (in module processes.views.actions), 70

includeme() (in module security), 66

includeme() (in module services), 74

includeme() (in module services.views.actions), 74

includeme() (in module settings), 87

includeme() (in module solr), 94

includeme() (in module solrsearch), 92

includeme() (in module storage), 73

includeme() (in module supervisor), 85

includeme() (in module twitcherclient), 68
 includeme() (in module wizard), 80
 includeme() (in module wizard.views.complexinputs), 75
 includeme() (in module wizard.views.done), 75
 includeme() (in module wizard.views.esgfsearch), 76
 includeme() (in module wizard.views.literalinputs), 76
 includeme() (in module wizard.views.solrsearch), 76
 includeme() (in module wizard.views.source), 77
 includeme() (in module wizard.views.start), 77
 includeme() (in module wizard.views.threddsservice), 78
 includeme() (in module wizard.views.wps), 78
 includeme() (in module wizard.views.wpsprocess), 79
 index_service() (settings.views.solr.SolrSettings method), 87
 index_thredds() (in module tasks.solr), 70
 init_ldap() (account.Ldap.LDAPAccount method), 88
 Inputs (class in monitor.panels.inputs), 97
 insert_record() (catalog.Catalog method), 64
 insert_record() (catalog.CatalogService method), 65
 insert_record() (catalog.MongodbCatalog method), 65
 invalid_exc() (in module tests.test_form), 83
 is_first() (wizard.views.WizardState method), 79
 is_last() (wizard.views.WizardState method), 79
 is_opendap() (cart.cart.CartItem method), 81
 is_opendap() (in module wps), 66
 is_public() (in module twitcherclient), 69
 is_service() (cart.cart.CartItem method), 81
 is_thredds_catalog() (cart.cart.CartItem method), 81
 is_url() (in module utils), 67

J

job_details() (in module monitor.panels), 97
 job_to_state() (in module wizard.views.start), 77
 JobFinished (class in events), 63
 JobList (class in monitor.views.list), 99
 JobListJson (class in monitor.views.list_json), 99
 JobsGrid (class in monitor.views.list), 99
 JobStarted (class in events), 63
 JobStatus (class in monitor.views.status), 100
 jsonify() (processes.views.execute_json.ExecuteProcessJson method), 71

L

label_td() (grid.CustomGrid method), 68
 labels_td() (monitor.views.list.JobsGrid method), 99
 LabelsSchema (class in monitor.views.list), 99
 layouts (module), 64
 Ldap (class in settings.views.ldap_config), 86
 ldap (module), 63
 ldap_login() (account.Ldap.LDAPAccount method), 88
 LDAPAccount (class in account.ldap), 88
 LDAPSchemas (class in account.ldap), 88
 LdapSchema (class in settings.schema), 87
 list_cart() (cart.actions.CartActions method), 80

list_processes() (processes.views.actions.ProcessesActions method), 70
 list_view() (services.views.services.Services method), 74
 literal_data() (wps.WPSSchema method), 67
 LiteralInputs (class in wizard.views.literalinputs), 76
 load() (cart.cart.Cart method), 81
 load() (wizard.views.WizardState method), 79
 loading() (esgf.views.esgflogon.ESGFLogon method), 94
 LocalAccount (class in account.local), 89
 localize_datetime() (in module utils), 67
 LocalSchema (class in account.local), 89
 log() (in module monitor.panels), 97
 login() (account.base.Account method), 88
 login_failure() (account.base.Account method), 88
 login_success() (account.base.Account method), 88
 logon() (in module esgf.logon), 94
 logout() (account.base.Account method), 88

M

main() (in module __init__), 66
 make_private() (monitor.views.actions.NodeActions method), 98
 make_public() (monitor.views.actions.NodeActions method), 98
 make_tags() (in module utils), 67
 Map (class in map), 91
 map (module), 91
 messages() (in module panels), 66
 mongodb() (in module db), 63
 MongodbCatalog (class in catalog), 65
 monitor (module), 97
 monitor.panels (module), 97
 monitor.panels.inputs (module), 97
 monitor.panels.outputs (module), 97
 monitor.utils (module), 100
 monitor.views (module), 98
 monitor.views.actions (module), 98
 monitor.views.details (module), 99
 monitor.views.list (module), 99
 monitor.views.list_json (module), 99
 monitor.views.status (module), 100
 monitor_buttons() (in module monitor.views.actions), 98
 MyAuthenticationPolicy (class in security), 65
 MyFetcher (class in providers.esgfopenid), 72
 MyView (class in views), 69

N

name_td() (wizard.views.threddsbrowser.Grid method), 78
 names() (wizard.views.WizardFavorite method), 79
 navbar() (in module panels), 66
 next() (wizard.views.Wizard method), 80
 next() (wizard.views.WizardState method), 79
 next_failure() (wizard.views.Wizard method), 80

- `next_ok()` (wizard.views.esgfsearch.ESGFSearchView method), 76
- `next_ok()` (wizard.views.Wizard method), 80
- `next_success()` (wizard.views.complexinputs.ComplexInputs method), 75
- `next_success()` (wizard.views.done.Done method), 75
- `next_success()` (wizard.views.esgfsearch.ESGFSearchView method), 76
- `next_success()` (wizard.views.literalinputs.LiteralInputs method), 76
- `next_success()` (wizard.views.solrsearch.SolrSearch method), 76
- `next_success()` (wizard.views.source.ChooseSource method), 77
- `next_success()` (wizard.views.start.Start method), 77
- `next_success()` (wizard.views.threddsbrowser.ThreddsBrowser method), 78
- `next_success()` (wizard.views.threddsservice.ThreddsService method), 78
- `next_success()` (wizard.views.Wizard method), 80
- `next_success()` (wizard.views.wps.ChooseWPS method), 78
- `next_success()` (wizard.views.wpsprocess.ChooseWPSProcess method), 79
- `NodeActions` (class in monitor.views.actions), 98
- `notfound()` (in module views), 69
- `notify_job_finished()` (in module monitor.views), 100
- `notify_job_started()` (in module monitor.views), 100
- O**
- `output_details()` (in module monitor.utils), 100
- `Outputs` (class in monitor.panels.outputs), 97
- `Overview` (class in processes.views.overview), 72
- `Overview` (class in settings.views.overview), 86
- `OverviewJson` (class in processes.views.overview_json), 72
- P**
- `PageLayout` (class in layouts), 64
- `panel()` (monitor.panels.inputs.Inputs method), 97
- `panel()` (monitor.panels.outputs.Outputs method), 97
- `panel()` (solr.panels.SolrIndexPanel method), 93
- `panel()` (solr.panels.SolrParamsPanel method), 93
- `panel_title()` (people.views.profile.Profile method), 90
- `panels` (module), 66
- `params()` (esgf.search.ESGFSearch method), 95
- `passwd_check()` (in module security), 65
- `patch` (module), 66
- `patch_myproxy_client()` (in module patch), 66
- `People` (class in people.views.list), 90
- `people` (module), 89
- `people.schema` (module), 90
- `people.views` (module), 89
- `people.views.actions` (module), 89
- `people.views.list` (module), 90
- `people.views.profile` (module), 90
- `PeopleGrid` (class in people.views.list), 90
- `permitted()` (utils.ActionButton method), 67
- `pinned_processes()` (in module utils), 67
- `pinned_processes()` (processes.views.overview.Overview method), 72
- `prev_ok()` (wizard.views.Wizard method), 80
- `previous()` (wizard.views.Wizard method), 80
- `previous()` (wizard.views.WizardState method), 79
- `previous_failure()` (wizard.views.Wizard method), 80
- `previous_success()` (wizard.views.Wizard method), 80
- `process_caption_form()` (monitor.views.list.JobList method), 99
- `process_constraints()` (in module esgf.metadata), 95
- `process_form()` (account.base.Account method), 88
- `process_form()` (esgf.views.esgflogon.ESGFLogon method), 94
- `process_form()` (people.views.profile.Profile method), 90
- `process_form()` (processes.views.execute.ExecuteProcess method), 71
- `process_form()` (services.views.registerservice.RegisterService method), 74
- `process_form()` (settings.views.processes.Processes method), 87
- `process_form()` (wizard.views.Wizard method), 80
- `process_inputs()` (in module monitor.panels.inputs), 97
- `process_labels_form()` (monitor.views.list.JobList method), 99
- `process_outputs()` (in module monitor.panels.outputs), 97
- `Processes` (class in settings.views.processes), 86
- `processes` (module), 70
- `processes.views` (module), 70
- `processes.views.actions` (module), 70
- `processes.views.execute` (module), 71
- `processes.views.execute_json` (module), 71
- `processes.views.list` (module), 71
- `processes.views.list_json` (module), 71
- `processes.views.overview` (module), 72
- `processes.views.overview_json` (module), 72
- `ProcessesActions` (class in processes.views.actions), 70
- `ProcessesSchema` (class in settings.schema), 87
- `ProcessList` (class in processes.views.list), 71
- `ProcessListJson` (class in processes.views.list_json), 71
- `Profile` (class in people.views.profile), 90
- `ProfileSchema` (class in people.schema), 90
- `project_title()` (layouts.PageLayout method), 64
- `providers` (module), 72
- `providers.esgfopenid` (module), 72
- Q**
- `query_params()` (esgf.search.ESGFSearch method), 95
- `query_params_from_appstruct()` (in module esgf.search), 95

query_path() (in module solrsearch.panels), 91

R

readonly() (people.views.profile.Profile method), 90

refresh_token() (esgf.slcsclient.ESGFSLCSCClient method), 96

refresh_token() (in module esgf.slcsclient), 96

register() (account.base.Account method), 88

RegisterService (class in services.views.registerservice), 74

remove_from_cart() (cart.actions.CartActions method), 81

remove_item() (cart.actions.CartActions method), 81

remove_item() (cart.cart.Cart method), 81

remove_service() (services.views.actions.ServiceActions method), 73

render_buttongroup_td() (grid.CustomGrid method), 68

render_flag_td() (grid.CustomGrid method), 68

render_format_td() (grid.CustomGrid method), 68

render_preview_td() (grid.CustomGrid method), 68

render_td() (grid.CustomGrid method), 68

render_title_td() (grid.CustomGrid method), 68

resources() (wizard.views.Wizard method), 80

ResourceWidget (class in geoform.widget), 93

restart_job() (monitor.views.actions.NodeActions method), 98

robotstxt_view() (in module views), 69

Root (class in security), 65

root_factory() (in module security), 65

root_path() (in module utils), 68

S

save() (cart.cart.Cart method), 81

save_chunk() (in module storage.views), 73

save_credentials() (in module esgf.logon), 94

save_log() (in module tasks.utils), 70

save_token() (esgf.slcsclient.ESGFSLCSCClient method), 96

save_token() (in module esgf.slcsclient), 96

Schema (class in services.views.registerservice), 74

Schema (class in solr.panels), 93

Schema (class in wizard.views.complexinputs), 75

Schema (class in wizard.views.source), 77

Schema (class in wizard.views.threddsbrowser), 77

Schema (class in wizard.views.threddsservice), 78

Schema (class in wizard.views.wpsprocess), 79

schema() (account.base.Account method), 88

schema() (account.esgf.ESGFAccount method), 88

schema() (account ldap.LDAPAccount method), 88

schema() (account.local.LocalAccount method), 89

schema() (people.views.profile.Profile method), 90

schema() (wizard.views.complexinputs.ComplexInputs method), 75

schema() (wizard.views.done.Done method), 75

schema() (wizard.views.esgfsearch.ESGFSearchView method), 76

schema() (wizard.views.literalinputs.LiteralInputs method), 76

schema() (wizard.views.solrsearch.SolrSearch method), 76

schema() (wizard.views.source.ChooseSource method), 77

schema() (wizard.views.start.Start method), 77

schema() (wizard.views.threddsbrowser.ThreddsBrowser method), 77

schema() (wizard.views.threddsservice.ThreddsService method), 78

schema() (wizard.views.Wizard method), 80

schema() (wizard.views.wps.ChooseWPS method), 78

schema() (wizard.views.wpsprocess.ChooseWPSProcess method), 79

search_datasets() (esgf.search.ESGFSearch method), 96

search_datasets() (esgf.views.esgfsearch.ESGFSearchActions method), 94

search_items() (esgf.search.ESGFSearch method), 95

search_items() (esgf.views.esgfsearch.ESGFSearchActions method), 94

security (module), 65

send_notification() (account.base.Account method), 88

serialize() (geoform.widget.BBoxWidget method), 93

serialize() (geoform.widget.ResourceWidget method), 93

serialize() (geoform.widget.TagsWidget method), 93

ServiceActions (class in services.views.actions), 73

Services (class in services.views.services), 74

services (module), 73

services.views (module), 73

services.views.actions (module), 73

services.views.registerservice (module), 74

services.views.services (module), 74

set() (wizard.views.WizardFavorite method), 79

set() (wizard.views.WizardState method), 79

set_favorite() (monitor.views.actions.NodeActions method), 98

settings (module), 86

settings.schema (module), 87

settings.views (module), 86

settings.views.ldap_config (module), 86

settings.views.overview (module), 86

settings.views.processes (module), 86

settings.views.solr (module), 87

SettingsChanged (class in events), 64

setUp() (tests.test_cart.CartTests method), 82

setUp() (tests.test_esgf_search.ESGFSearchTests method), 83

setUp() (tests.test_settings.UserSettingsFunctionalTests method), 84

setUp() (tests.test_settings.UserSettingsTests method), 84

sign_in() (account.local.LocalAccount method), 89

size_td() (grid.CustomGrid method), 68
skip_csrf_token() (in module utils), 67
solr (module), 93
solr.panels (module), 93
solr_search() (in module solrsearch.search), 92
SolrIndexPanel (class in solr.panels), 93
SolrPanel (class in solr.panels), 93
SolrParamsPanel (class in solr.panels), 93
SolrSearch (class in solrsearch.views.solrsearch), 91
SolrSearch (class in wizard.views.solrsearch), 76
solrsearch (module), 91
solrsearch() (in module solrsearch.panels), 91
solrsearch.panels (module), 91
solrsearch.schema (module), 92
solrsearch.search (module), 92
solrsearch.views (module), 91
solrsearch.views.actions (module), 91
solrsearch.views.solrsearch (module), 91
solrsearch_script() (in module solrsearch.panels), 91
SolrSearchSchema (class in solrsearch.schema), 92
SolrSettings (class in settings.views.solr), 87
SourceSchemaNode (class in wizard.views.source), 77
Start (class in wizard.views.start), 77
state_td() (supervisor.views.supervisor.Grid method), 85
status_td() (monitor.views.list.JobsGrid method), 99
storage (module), 73
storage.views (module), 73
succeeded() (events.JobFinished method), 64
success() (wizard.views.complexinputs.ComplexInputs method), 75
success() (wizard.views.done.Done method), 75
success() (wizard.views.start.Start method), 77
success() (wizard.views.threddsservice.ThreddsService method), 78
success() (wizard.views.Wizard method), 80
Supervisor (class in supervisor.views.supervisor), 85
supervisor (module), 85
supervisor.views (module), 85
supervisor.views.supervisor (module), 85
supervisor.views.supervisor_log (module), 85
supervisor_process() (supervisor.views.supervisor.Supervisor method), 85
SupervisorLog (class in supervisor.views.supervisor_log), 85

T

TagsWidget (class in geoform.widget), 93
task_result() (in module tasks.utils), 70
tasks (module), 69
tasks.esgflogon (module), 69
tasks.execute (module), 69
tasks.solr (module), 70
tasks.utils (module), 70

tasks.workflow (module), 70
tearDown() (tests.test_cart.CartTests method), 82
tearDown() (tests.test_esgf_search.ESGFSearchTests method), 83
tearDown() (tests.test_settings.UserSettingsTests method), 84
temporal_filter() (in module esgf.search), 95
test_build_constraints_dict() (in module tests.test_esgf_search), 83
test_cart() (in module tests.test_cart), 82
test_cart_from_request() (tests.test_cart.CartTests method), 82
test_check_status() (in module tests.test_wps), 84
test_clear_cart() (in module tests.test_cart), 82
test_convert_constraints() (in module tests.test_esgf_metadata), 82
test_convert_states_to_nodes() (in module tests.test_wizard), 84
test_date_from_filename() (in module tests.test_esgf_search), 83
test_default() (tests.test_form.TestBBoxValidator method), 83
test_default() (tests.test_form.TestTextValidator method), 83
test_default() (tests.test_form.TestURLValidator method), 83
test_doc2record() (in module tests.test_catalog), 82
test_empty() (tests.test_form.TestTextValidator method), 83
test_file_scheme() (tests.test_form.TestURLValidator method), 83
test_format_tags() (in module tests.test_utils), 84
test_headline() (in module tests.test_utils), 84
test_invalid_path() (tests.test_form.TestURLValidator method), 83
test_invalid_relative_path() (tests.test_form.TestURLValidator method), 83
test_make_tags() (in module tests.test_utils), 84
test_maxx() (tests.test_form.TestBBoxValidator method), 83
test_maxy() (tests.test_form.TestBBoxValidator method), 83
test_minx() (tests.test_form.TestBBoxValidator method), 83
test_miny() (tests.test_form.TestBBoxValidator method), 83
test_params() (tests.test_esgf_search.ESGFSearchTests method), 83
test_query_params() (tests.test_esgf_search.ESGFSearchTests method), 83
test_restricted_chars() (tests.test_form.TestTextValidator method), 83
test_search_datasets() (tests.test_esgf_search.ESGFSearchTests method), 83

test_search_items() (tests.test_esgf_search.ESGFSearchTests method), 83

test_temporal_filter() (in module tests.test_esgf_search), 83

test_time_ago_in_words() (in module tests.test_utils), 84

test_user_view() (tests.test_settings.UserSettingsFunctionalTests method), 84

test_user_view() (tests.test_settings.UserSettingsTests method), 84

test_variable_filter() (in module tests.test_esgf_search), 83

TestBBoxValidator (class in tests.test_form), 83

tests (module), 82

tests.test_cart (module), 82

tests.test_catalog (module), 82

tests.test_esgf_metadata (module), 82

tests.test_esgf_search (module), 83

tests.test_form (module), 83

tests.test_settings (module), 84

tests.test_utils (module), 84

tests.test_wizard (module), 84

tests.test_wps (module), 84

TestTextValidator (class in tests.test_form), 83

TestURLValidator (class in tests.test_form), 83

TextValidator (class in geoform.form), 92

ThreddsBrowser (class in wizard.views.threddsbrowser), 77

ThreddsService (class in wizard.views.threddsservice), 78

time_ago_in_words() (in module utils), 68

time_ago_td() (grid.CustomGrid method), 68

timestamp_td() (grid.CustomGrid method), 68

title() (cart.cart.CartItem method), 81

to_json() (cart.cart.Cart method), 81

to_json() (cart.cart.CartItem method), 81

twitcher_service_factory() (in module twitcherclient), 68

twitcherclient (module), 68

TwitcherSchema (class in people.schema), 90

U

unknown_failure() (in module views), 69

unset_favorite() (monitor.views.actions.NodeActions method), 98

update_esgf_certs() (people.views.actions.Actions method), 89

upload() (in module storage.views), 73

url() (utils.ActionButton method), 67

URLValidator (class in geoform.form), 92

use_ajax() (wizard.views.Wizard method), 80

user_td() (grid.CustomGrid method), 68

userid_td() (grid.CustomGrid method), 68

UserSettingsFunctionalTests (class in tests.test_settings), 84

UserSettingsTests (class in tests.test_settings), 84

V

variable_filter() (in module esgf.search), 95

view() (cart.views.Cart method), 82

view() (dashboard.views.Dashboard method), 86

view() (esgf.views.esgflogon.ESGFLogon method), 94

view() (map.Map method), 91

view() (monitor.views.details.Details method), 99

view() (monitor.views.list.JobList method), 99

view() (monitor.views.list_json.JobListJson method), 99

view() (monitor.views.status.JobStatus method), 100

view() (people.views.list.People method), 90

view() (people.views.profile.Profile method), 90

view() (processes.views.execute.ExecuteProcess method), 71

view() (processes.views.execute_json.ExecuteProcessJson method), 71

view() (processes.views.list.ProcessList method), 71

view() (processes.views.list_json.ProcessListJson method), 71

view() (processes.views.overview.Overview method), 72

view() (processes.views.overview_json.OverviewJson method), 72

view() (services.views.registerservice.RegisterService method), 74

view() (settings.views.ldap_config.Ldap method), 86

view() (settings.views.overview.Overview method), 86

view() (settings.views.processes.Processes method), 87

view() (settings.views.solr.SolrSettings method), 87

view() (solrsearch.views.solrsearch.SolrSearch method), 91

view() (supervisor.views.supervisor.Supervisor method), 85

view() (supervisor.views.supervisor_log.SupervisorLog method), 85

view() (views.Home method), 69

view() (wizard.views.complexinputs.ComplexInputs method), 75

view() (wizard.views.done.Done method), 75

view() (wizard.views.literalinputs.LiteralInputs method), 76

view() (wizard.views.solrsearch.SolrSearch method), 76

view() (wizard.views.source.ChooseSource method), 77

view() (wizard.views.start.Start method), 77

view() (wizard.views.threddsbrowser.ThreddsBrowser method), 78

view() (wizard.views.threddsservice.ThreddsService method), 78

view() (wizard.views.Wizard method), 80

view() (wizard.views.wps.ChooseWPS method), 78

view() (wizard.views.wpsprocess.ChooseWPSProcess method), 79

views (module), 69

W

`wait_secs()` (in module `tasks.utils`), 70
`Wizard` (class in `wizard.views`), 79
`wizard` (module), 75
`wizard.views` (module), 75
`wizard.views.complexinputs` (module), 75
`wizard.views.done` (module), 75
`wizard.views.esgfsearch` (module), 76
`wizard.views.literalinputs` (module), 76
`wizard.views.solrsearch` (module), 76
`wizard.views.source` (module), 77
`wizard.views.start` (module), 77
`wizard.views.threddsbrowser` (module), 77
`wizard.views.threddsservice` (module), 78
`wizard.views.wps` (module), 78
`wizard.views.wpsprocess` (module), 79
`WizardFavorite` (class in `wizard.views`), 79
`WizardState` (class in `wizard.views`), 79
`workflow_description()` (`wizard.views.done.Done` method), 75
`wps` (module), 66
`wps_caps_url()` (in module `utils`), 68
`wps_describe_url()` (in module `utils`), 68
`wps_headers()` (in module `tasks.utils`), 70
`wps_services()` (`processes.views.overview.Overview` method), 72
`WPSSchema` (class in `wps`), 67

X

`xml()` (in module `monitor.panels`), 97