ECAS WPS Demo Documentation

Release 0.1.0

Carsten Ehbrecht

Nov 01, 2018

Contents:

A Web Processing Service for the ECASLab to provide a climate analytic service using Ophidia.

- Free software: Apache Software License 2.0
- Documentation: https://ecas-wps-demo.readthedocs.io.

CHAPTER 1

Credits

This package was created with Cookiecutter and the bird-house/cookiecutter-birdhouse project template.

1.1 Installation

1.1.1 Install from Anaconda

1.1.2 Install from GitHub

Check out code from the ECAS WPS Demo GitHub repo and start the installation:

```
$ git clone https://github.com/cehbrecht/ecas-wps-demo.git
$ cd ecaswps
$ conda env create -f environment.yml
$ source activate ecaswps
$ python setup.py develop
```

. . . or do it the lazy way

The previous installation instructions assume you have Anaconda installed. We provide also a `Makefile` to run this installation without additional steps:

```
$ git clone https://github.com/cehbrecht/ecas-wps-demo.git
$ cd ecaswps
$ make clean    # cleans up a previous Conda environment
$ make install  # installs Conda if necessary and runs the above installation steps
```

1.1.3 Start ECAS WPS Demo PyWPS service

After successful installation you can start the service using the `ecaswps` command-line.

```
$ ecaswps --help # show help
$ ecaswps start  # start service with default configuration

OR

$ ecaswps start --daemon # start service as daemon
loading configuration
forked process id: 42
```

The deployed WPS service is by default available on:

http://localhost:5000/wps?service=WPS&version=1.0.0&request=GetCapabilities.

Note: Remember the process ID (PID) so you can stop the service with `kill PID`.

You can find which process uses a given port using the following command (here for port 5000):

```
$ netstat -nlp | grep :5000
```

Check the log files for errors:

```
$ tail -f  pywps.log
```

... or do it the lazy way

You can also use the `Makefile` to start and stop the service:

```
$ make start
$ make status
$ tail -f pywps.log
$ make stop
```

1.1.4 Run ECAS WPS Demo as Docker container

You can also run ECAS WPS Demo as a Docker container, see the Tutorial.

1.1.5 Use Ansible to deploy ECAS WPS Demo on your System

Use the Ansible playbook for PyWPS to deploy ECAS WPS Demo on your system. Follow the example for ECAS WPS Demo given in the playbook.

1.1.6 Building the docs

First install dependencies for the documentation:

```
$ make bootstrap_dev
$ make docs
```

1.2 Configuration

1.2.1 Command-line options

You can overwrite the default PyWPS configuration by using command-line options. See the ECAS WPS Demo help which options are available:

```
$ ecaswps start --help
--hostname HOSTNAME        hostname in PyWPS configuration.
--port PORT                port in PyWPS configuration.
```

Start service with different hostname and port:

```
$ ecaswps start --hostname localhost --port 5001
```

1.2.2 Use a custom configuration file

You can overwrite the default PyWPS configuration by providing your own PyWPS configuration file (just modifiy the options you want to change). Use one of the existing `sample-*.cfg` files as example and copy them to `etc/custom.cfg`.

For example change the hostname (*demo.org*) and logging level:

```
$ cd ecaswps
$ vim etc/custom.cfg
$ cat etc/custom.cfg
[server]
url = http://demo.org:5000/wps
outputurl = http://demo.org:5000/outputs

[logging]
level = DEBUG
```

Start the service with your custom configuration:

```
# start the service with this configuration
$ ecaswps start -c etc/custom.cfg
```

1.3 Processes

1.3.1 Sleep

class ecaswps.processes.wps_sleep.**Sleep**
sleep Sleep Process (v1.0)

Testing a long running process, in the sleep.This process will sleep for a given delay or 10 seconds if not a valid value.

Parameters **delay** (*float*) – Delay between every update

Returns **sleep_output** – Sleep Output

Return type string

References

- PyWPS Demo

1.3.2 TropicalNights

class ecaswps.processes.wps_index_tn.**TropicalNights**

tropical_nights Tropical Nights (v1.0)

Computes the Tropical Nights index: starting from the daily minimum temperature (1980-1990) TN, the Tropical Nights index is the number of days where TN > T (T is a reference temperature, e.g. 20 degree celsius)

Parameters **dataset** (*string*) – Dataset

Returns **output** – Map of Trophical Nights

Return type *image/png*

References

- ECASLab
- ECASLab Notebooks
- Documentation
- Media

1.4 Changes

1.4.1 0.1.0 (2018-08-06)

- First release.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Index