rooki Documentation

Release 0.5.0

Carsten Ehbrecht

Feb 08, 2023

CONTENTS:

1 Online Demo 3

2 Credits 5

Rooki is a client for roocs climate data operations service (rook).

The rooki python package is a lightweight wrapper around the birdy client library for WPS. It provides the *rooki* python object that has methods that can be called to query and invoke the rook WPS.

A Jupyter Notebook is provided to demonstrate the basic use of rooki.

Full documentation is on ReadTheDocs.

CHAPTER

ONE

ONLINE DEMO

You can try Rooki online using Binder (just click on the binder link below), or view the notebooks on NBViewer.

CHAPTER

TWO

CREDITS

This package was created with Cookiecutter and the cookiecutter-pypackage project template.

2.1 Installation

2.1.1 Install from Anaconda

- TODO

2.1.2 Install from PyPi

Create a conda environment with birdy and install with pip:

```
$ conda create -n rooki -c conda-forge python=3.8 birdy
$ conda activate rooki
$ pip install rooki
```

2.1.3 Install from GitHub

Check out code from the rooki GitHub repo and start the installation:

```
$ git clone https://github.com/roocs/rooki.git
$ cd rooki
$ conda env create -f environment.yml
$ pip install -e .
```

2.2 Usage

```
# Optional: set ROOK_URL ... or use default
import os
os.environ['ROOK_URL'] = http://localhost:5000/wps
# import rooki
from rooki import rooki
# run subset on c3s-cmip5 dataset with time selection
response = rooki.subset(
```

(continues on next page)

(continued from previous page)

```
  collection='c3s-cmip5.output1.ICHEC.EC-EARTH.historical.day.atmos.day.r1i1p1.tas.latest
↪',
  time='1860-01-01/1900-12-30')
# successful?
response.ok
# show links to result files
response.download_urls()
```

2.3 Development Guide

2.3.1 Get Started!

Check out code from the rooki GitHub repo and start the installation:

```
$ git clone https://github.com/roocs/rooki.git
$ cd rooki
$ conda env create -f environment.yml
$ conda activate rooki
$ pip install -e .
```

Install additional dependencies:

```
$ pip install -r requirements_dev.txt
```

When you're done making changes, check that your changes pass *black*, *flake8* and the tests:

```
$ black rooki tests
$ flake8 rooki tests
$ pytest tests
```

Or use the Makefile:

```
$ make lint
$ make test
```

2.3.2 Add pre-commit hooks

Before committing your changes, we ask that you install *pre-commit* in your environment. *Pre-commit* runs git hooks that ensure that your code resembles that of the project and catches and corrects any small errors or inconsistencies when you *git commit*:

```
$ conda install -c conda-forge pre_commit
$ pre-commit install
```

2.3.3 Write Documentation

You can find the documentation in the *docs/source* folder. To generate the Sphinx documentation locally you can use the *Makefile*:

```
$ make docs
```

2.3.4 Bump a new version

Make a new version of rooki in the following steps:

- Make sure everything is commit to GitHub.
- Update `HISTORY.rst` with the next version.
- Dry Run: `bumpversion --dry-run --verbose patch # --new-version 0.2.1`
- Do it: `bumpversion --new-version patch`
- ... or: `bumpversion --new-version minor # --new-version 0.3.0`
- Push it: `git push --tags`

See the bumpversion documentation for details.

2.4 Notebooks

These notebooks demonstrate the use of rooki.

2.4.1 Use HTTP requests for WPS rook

- rook wps: https://github.com/roocs/rook
- wps documentation: http://geoprocessing.info/wpsdoc/

[1]:
```
import requests

url = 'http://rook.dkrz.de/wps'
```

GetCapabilities

[2]:
```
req_url = f"{url}?service=WPS&request=GetCapabilities"
req_url
```

[2]:
```
'http://rook.dkrz.de/wps?service=WPS&request=GetCapabilities'
```

[3]:
```
resp = requests.get(req_url)
resp.ok
```

[3]:
```
True
```

[4]:
```
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- PyWPS 4.4.0 -->
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US" xmlns:xlink="http://www.
↪w3.org/1999/xlink" xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.
↪opengis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
↪schemaLocation="http://www.opengis.net/wps/1.0.0 ../wpsGetCapabilities_response.xsd"␣
↪updateSequence="1">
    <ows:ServiceIdentification>
        <ows:Title>rook</ows:Title>
        <ows:Abstract>A WPS service for roocs.</ows:Abstract>
        <ows:Keywords>
        <ows:Keyword>PyWPS</ows:Keyword>
        <ows:Keyword> WPS</ows:Keyword>
        <ows:Keyword> OGC</ows:Keyword>
        <ows:Keyword> processing</ows:Keyword>
        <ows:Keyword> birdhouse</ows:Keyword>
        <ows:Keyword> roocs</ows:Keyword>
        <ows:Keyword> demo</ows:Keyword>
        <ows:Keyword> cp4cds</ows:Keyword>
        <ows:Keyword> copernicus</ows:Keyword>
        <ows:Keyword> ecmwf</ows:Keyword>
            <ows:Type codeSpace="ISOTC211/19115">theme</ows:Type>
        </ows:Keywords>
        <ows:ServiceType>WPS</ows:ServiceType>
        <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
        <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
        <ows:Fees></ows:Fees>
        <ows:AccessConstraints>
        open access
        </ows:AccessConstraints>
    </ows:ServiceIdentification>
    <ows:ServiceProvider>
        <ows:ProviderName>rook4.cloud.dkrz.de</ows:ProviderName>
        <ows:ProviderSite xlink:href="https://roocs.github.io/"/>
        <ows:ServiceContact>
            <ows:IndividualName>DKRZ</ows:IndividualName>
            <ows:PositionName>Position Title</ows:PositionName>
            <ows:ContactInfo>
                <ows:Phone>
                    <ows:Voice>+xx-xxx-xxx-xxxx</ows:Voice>
                    <ows:Facsimile></ows:Facsimile>
                </ows:Phone>
                <ows:Address>
                    <ows:DeliveryPoint></ows:DeliveryPoint>
                    <ows:City>Hamburg</ows:City>
                    <ows:AdministrativeArea></ows:AdministrativeArea>
                    <ows:PostalCode>Zip or Postal Code</ows:PostalCode>
                    <ows:Country>Germany</ows:Country>
                    <ows:ElectronicMailAddress>Email Address</ows:ElectronicMailAddress>
                </ows:Address>
            </ows:ContactInfo>
        </ows:ServiceContact>
    </ows:ServiceProvider>
```

(continues on next page)

(continued from previous page)

```
    <ows:OperationsMetadata>
        <ows:Operation name="GetCapabilities">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://rook4.cloud.dkrz.de:80/wps"/>
                </ows:HTTP>
            </ows:DCP>
        </ows:Operation>
        <ows:Operation name="DescribeProcess">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://rook4.cloud.dkrz.de:80/wps"/>
                    <ows:Post xlink:href="http://rook4.cloud.dkrz.de:80/wps"/>
                </ows:HTTP>
            </ows:DCP>
        </ows:Operation>
        <ows:Operation name="Execute">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://rook4.cloud.dkrz.de:80/wps"/>
                    <ows:Post xlink:href="http://rook4.cloud.dkrz.de:80/wps"/>
                </ows:HTTP>
            </ows:DCP>
        </ows:Operation>
    </ows:OperationsMetadata>
    <wps:ProcessOfferings>
        <wps:Process wps:processVersion="1.0">
            <ows:Identifier>subset</ows:Identifier>
            <ows:Title>Subset</ows:Title>
            <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</
↪ows:Abstract>
            <ows:Metadata xlink:title="DAOPS" xlink:type="simple"
              xlink:href="https://github.com/roocs/daops"
            />
        </wps:Process>
        <wps:Process wps:processVersion="1.0">
            <ows:Identifier>average</ows:Identifier>
            <ows:Title>Average</ows:Title>
            <ows:Abstract>Run averaging on climate model data. Calls daops operators.</
↪ows:Abstract>
            <ows:Metadata xlink:title="DAOPS" xlink:type="simple"
              xlink:href="https://github.com/roocs/daops"
            />
        </wps:Process>
        <wps:Process wps:processVersion="1.0">
            <ows:Identifier>orchestrate</ows:Identifier>
            <ows:Title>Orchestrate</ows:Title>
            <ows:Abstract>Run a workflow with combined operations. A workflow can be␣
↪build using the rooki client.</ows:Abstract>
            <ows:Metadata xlink:title="Rooki" xlink:type="simple"
              xlink:href="https://github.com/roocs/rooki"
            />
```

(continues on next page)

(continued from previous page)

```
        </wps:Process>
    </wps:ProcessOfferings>
    <wps:Languages>
        <wps:Default>
            <ows:Language>en-US</ows:Language>
        </wps:Default>
        <wps:Supported>
            <ows:Language>en-US</ows:Language>
        </wps:Supported>
    </wps:Languages>
</wps:Capabilities>
```

DescribeProcess subset

[5]:
```
req_url = f"{url}?service=WPS&version=1.0.0&request=DescribeProcess&identifier=subset"
req_url
```

[5]:
```
'http://rook.dkrz.de/wps?service=WPS&version=1.0.0&request=DescribeProcess&
↪identifier=subset'
```

[6]:
```
resp = requests.get(req_url)
resp.ok
```

[6]:
```
True
```

[7]:
```
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- PyWPS 4.4.0 -->
<wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://
↪www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://
↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.
↪0 ../wpsDescribeProcess_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US">
    <ProcessDescription wps:processVersion="1.0" storeSupported="true" statusSupported=
↪"true">
        <ows:Identifier>subset</ows:Identifier>
        <ows:Title>Subset</ows:Title>
        <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</ows:
↪Abstract>
        <ows:Metadata xlink:title="DAOPS" xlink:type="simple"
            xlink:href="https://github.com/roocs/daops"
        />
        <DataInputs>
            <Input minOccurs="1" maxOccurs="1">
                <ows:Identifier>collection</ows:Identifier>
                <ows:Title>Collection</ows:Title>
                <ows:Abstract>A dataset identifier or list of comma separated␣
↪identifiersExample: c3s-cmip5.output1.ICHEC.EC-EARTH.historical.day.atmos.day.r1i1p1.
↪tas.latest</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
```

(continues on next page)

(continued from previous page)

```
↪string</ows:DataType>
                </LiteralData>
            </Input>
            <Input minOccurs="0" maxOccurs="1">
                <ows:Identifier>time</ows:Identifier>
                <ows:Title>Time Period</ows:Title>
                <ows:Abstract>The time period to subset over separated by /Example: 1860-
↪01-01/1900-12-30</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
↪string</ows:DataType>
                </LiteralData>
            </Input>
            <Input minOccurs="0" maxOccurs="1">
                <ows:Identifier>area</ows:Identifier>
                <ows:Title>Area</ows:Title>
                <ows:Abstract>The area to subset over as 4 comma separated values.
↪Example: 0.,49.,10.,65</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
↪string</ows:DataType>
                </LiteralData>
            </Input>
            <Input minOccurs="0" maxOccurs="1">
                <ows:Identifier>level</ows:Identifier>
                <ows:Title>Level</ows:Title>
                <ows:Abstract>The level range to subset over separated by a /Example: 0/
↪1000</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#string">
↪string</ows:DataType>
                </LiteralData>
            </Input>
            <Input minOccurs="1" maxOccurs="1">
                <ows:Identifier>pre_checked</ows:Identifier>
                <ows:Title>Pre-Checked</ows:Title>
                <ows:Abstract>Use checked data only.</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#boolean">
↪boolean</ows:DataType>
                <DefaultValue>False</DefaultValue>
                </LiteralData>
            </Input>
            <Input minOccurs="1" maxOccurs="1">
                <ows:Identifier>apply_fixes</ows:Identifier>
                <ows:Title>Apply Fixes</ows:Title>
                <ows:Abstract>Apply fixes to datasets.</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#boolean">
↪boolean</ows:DataType>
                <DefaultValue>False</DefaultValue>
                </LiteralData>
```

(continues on next page)

(continued from previous page)

```
            </Input>
            <Input minOccurs="1" maxOccurs="1">
                <ows:Identifier>original_files</ows:Identifier>
                <ows:Title>Original Files</ows:Title>
                <ows:Abstract>Return original files only.</ows:Abstract>
                <LiteralData>
                <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#boolean">
↪boolean</ows:DataType>
                <DefaultValue>False</DefaultValue>
                </LiteralData>
            </Input>
        </DataInputs>
        <ProcessOutputs>
            <Output>
                <ows:Identifier>output</ows:Identifier>
                <ows:Title>METALINK v4 output</ows:Title>
                <ows:Abstract>Metalink v4 document with references to NetCDF files.</ows:
↪Abstract>
                <ComplexOutput>
                    <Default>
                        <Format>
                            <MimeType>application/metalink+xml; version=4.0</MimeType>
                            <Schema>metalink/4.0/metalink4.xsd</Schema>
                        </Format>
                    </Default>
                    <Supported>
                        <Format>
                            <MimeType>application/metalink+xml; version=4.0</MimeType>
                            <Schema>metalink/4.0/metalink4.xsd</Schema>
                        </Format>
                    </Supported>
                </ComplexOutput>
            </Output>
            <Output>
                <ows:Identifier>prov</ows:Identifier>
                <ows:Title>Provenance</ows:Title>
                <ows:Abstract>Provenance document using W3C standard.</ows:Abstract>
                <ComplexOutput>
                    <Default>
                        <Format>
                            <MimeType>application/json</MimeType>
                        </Format>
                    </Default>
                    <Supported>
                        <Format>
                            <MimeType>application/json</MimeType>
                        </Format>
                    </Supported>
                </ComplexOutput>
            </Output>
            <Output>
                <ows:Identifier>prov_plot</ows:Identifier>
```

(continues on next page)

(continued from previous page)

```
                <ows:Title>Provenance Diagram</ows:Title>
                <ows:Abstract>Provenance document as diagram.</ows:Abstract>
                <ComplexOutput>
                    <Default>
                        <Format>
                            <MimeType>image/png</MimeType>
                            <Encoding>base64</Encoding>
                        </Format>
                    </Default>
                    <Supported>
                        <Format>
                            <MimeType>image/png</MimeType>
                            <Encoding>base64</Encoding>
                        </Format>
                    </Supported>
                </ComplexOutput>
            </Output>
        </ProcessOutputs>
    </ProcessDescription>
</wps:ProcessDescriptions>
```

Execute subset (sync mode)

Edit data inputs

[8]:
```
collection = "CMIP6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.v20180803"
time = "1985-01-01/2014-12-30"
```

[9]:
```
datainputs = f"DataInputs=collection={collection};time={time}"
req_url = f"{url}?service=WPS&version=1.0.0&request=Execute&identifier=subset&
↪{datainputs}"
req_url
```

[9]:
```
'http://rook.dkrz.de/wps?service=WPS&version=1.0.0&request=Execute&identifier=subset&
↪DataInputs=collection=CMIP6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803;time=1985-01-01/2014-12-30'
```

[10]:
```
resp = requests.get(req_url)
resp.ok
```

[10]:
```
True
```

[11]:
```
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.
↪opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.
↪w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ..
↪/wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US"␣
↪serviceInstance="http://rook4.cloud.dkrz.de:80/wps?request=GetCapabilities&amp;amp;
↪service=WPS" statusLocation="">
```

(continues on next page)

(continued from previous page)

```
    <wps:Process wps:processVersion="1.0">
        <ows:Identifier>subset</ows:Identifier>
        <ows:Title>Subset</ows:Title>
        <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</ows:
↪Abstract>
        </wps:Process>
    <wps:Status creationTime="2021-03-18T15:08:21Z">
        <wps:ProcessSucceeded>PyWPS Process Subset finished</wps:ProcessSucceeded>
        </wps:Status>
        <wps:ProcessOutputs>
                <wps:Output>
            <ows:Identifier>output</ows:Identifier>
            <ows:Title>METALINK v4 output</ows:Title>
            <ows:Abstract>Metalink v4 document with references to NetCDF files.</ows:
↪Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/63eb9d28-
↪87f3-11eb-b8ed-fa163e1098db/input.meta4" mimeType="application/metalink+xml; version=4.
↪0" encoding="" schema="metalink/4.0/metalink4.xsd"/>
                </wps:Output>
                <wps:Output>
            <ows:Identifier>prov</ows:Identifier>
            <ows:Title>Provenance</ows:Title>
            <ows:Abstract>Provenance document using W3C standard.</ows:Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/63eb9d28-
↪87f3-11eb-b8ed-fa163e1098db/provenance.json" mimeType="application/json" encoding=""␣
↪schema=""/>
                </wps:Output>
                <wps:Output>
            <ows:Identifier>prov_plot</ows:Identifier>
            <ows:Title>Provenance Diagram</ows:Title>
            <ows:Abstract>Provenance document as diagram.</ows:Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/63eb9d28-
↪87f3-11eb-b8ed-fa163e1098db/provenance.png" mimeType="image/png" encoding="base64"␣
↪schema=""/>
                </wps:Output>
        </wps:ProcessOutputs>
</wps:ExecuteResponse>
```

Load metalink result document

Replace the metalink output URL.

```
metalink_url = ''
```

[12]:
```
metalink_url = 'http://rook4.cloud.dkrz.de/outputs/rook/8c23a070-87f2-11eb-bc89-
↪fa163e1098db/provenance.json'
```

[13]:
```
print(requests.get(metalink_url).text)
```

```
{"prefix": {"provone": "http://purl.dataone.org/provone/2015/01/15/ontology#", "dcterms":
↪ "http://purl.org/dc/terms/", "default": "http://purl.org/roocs/prov#"}, "agent": {
↪"copernicus_CDS": {"prov:type": "prov:Organization", "dcterms:title": "Copernicus␣
```

(continues on next page)

(continued from previous page)

```
↪Climate Data Store"}, "rook": {"prov:type": "prov:SoftwareAgent", "dcterms:source":
↪"https://github.com/roocs/rook/releases/tag/v0.4.0"}, "daops": {"prov:type": "prov:
↪SoftwareAgent", "dcterms:source": "https://github.com/roocs/daops/releases/tag/v0.5.0"}
↪}, "wasAttributedTo": {"_:id1": {"prov:entity": "rook", "prov:agent": "copernicus_CDS"}
↪}, "activity": {"subset": {"time": "1985-01-01/2014-12-30", "apply_fixes": false}},
↪"entity": {"CMIP6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.v20180803":
↪{}, "rlds_Amon_IPSL-CM6A-LR_historical_r1i1p1f1_gr_19850116-20141216.nc": {}},
↪"wasStartedBy": {"_:id2": {"prov:activity": "subset", "prov:trigger": "rook", "prov:
↪starter": "daops"}}, "wasDerivedFrom": {"_:id3": {"prov:generatedEntity": "rlds_Amon_
↪IPSL-CM6A-LR_historical_r1i1p1f1_gr_19850116-20141216.nc", "prov:usedEntity": "CMIP6.
↪CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.v20180803", "prov:activity":
↪"subset"}}}
```

Execute subset (async mode)

[14]:
```
req_url = f"{url}?service=WPS&version=1.0.0&request=Execute&identifier=subset&
↪{datainputs}"
req_url += "&status=true&storeExecuteResponse=true"
req_url
```

[14]:
```
'http://rook.dkrz.de/wps?service=WPS&version=1.0.0&request=Execute&identifier=subset&
↪DataInputs=collection=CMIP6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803;time=1985-01-01/2014-12-30&status=true&storeExecuteResponse=true'
```

[15]:
```
resp = requests.get(req_url)
resp.ok
```

[15]:
```
True
```

[16]:
```
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.
↪opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.
↪w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ..
↪/wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US"␣
↪serviceInstance="http://rook4.cloud.dkrz.de:80/wps?request=GetCapabilities&amp;amp;
↪service=WPS" statusLocation="http://rook4.cloud.dkrz.de:80/outputs/rook/65b4dfc0-87f3-
↪11eb-a430-fa163e1098db.xml">
    <wps:Process wps:processVersion="1.0">
        <ows:Identifier>subset</ows:Identifier>
        <ows:Title>Subset</ows:Title>
        <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</ows:
↪Abstract>
        </wps:Process>
    <wps:Status creationTime="2021-03-18T15:08:22Z">
        <wps:ProcessAccepted percentCompleted="0">PyWPS Process subset accepted</wps:
↪ProcessAccepted>
        </wps:Status>
</wps:ExecuteResponse>
```

Poll status location

Replace the statusLocation URL.

```
statusLocation = ''
```

[17]:
```
# statusLocation = ''
statusLocation = 'http://rook4.cloud.dkrz.de/outputs/rook/bc97d460-87f2-11eb-b8ed-
↪fa163e1098db.xml'
```

[18]:
```
resp = requests.get(statusLocation)
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.
↪opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.
↪w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ..
↪/wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US"␣
↪serviceInstance="http://rook4.cloud.dkrz.de:80/wps?request=GetCapabilities&amp;amp;
↪service=WPS" statusLocation="http://rook4.cloud.dkrz.de:80/outputs/rook/bc97d460-87f2-
↪11eb-b8ed-fa163e1098db.xml">
    <wps:Process wps:processVersion="1.0">
        <ows:Identifier>subset</ows:Identifier>
        <ows:Title>Subset</ows:Title>
        <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</ows:
↪Abstract>
        </wps:Process>
    <wps:Status creationTime="2021-03-18T15:03:44Z">
        <wps:ProcessSucceeded>PyWPS Process Subset finished</wps:ProcessSucceeded>
        </wps:Status>
        <wps:ProcessOutputs>
                <wps:Output>
            <ows:Identifier>output</ows:Identifier>
            <ows:Title>METALINK v4 output</ows:Title>
            <ows:Abstract>Metalink v4 document with references to NetCDF files.</ows:
↪Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/bc97d460-
↪87f2-11eb-b8ed-fa163e1098db/input.meta4" mimeType="application/metalink+xml; version=4.
↪0" encoding="" schema="metalink/4.0/metalink4.xsd"/>
                </wps:Output>
                <wps:Output>
            <ows:Identifier>prov</ows:Identifier>
            <ows:Title>Provenance</ows:Title>
            <ows:Abstract>Provenance document using W3C standard.</ows:Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/bc97d460-
↪87f2-11eb-b8ed-fa163e1098db/provenance.json" mimeType="application/json" encoding=""␣
↪schema=""/>
                </wps:Output>
                <wps:Output>
            <ows:Identifier>prov_plot</ows:Identifier>
            <ows:Title>Provenance Diagram</ows:Title>
            <ows:Abstract>Provenance document as diagram.</ows:Abstract>
            <wps:Reference href="http://rook4.cloud.dkrz.de:80/outputs/rook/bc97d460-
↪87f2-11eb-b8ed-fa163e1098db/provenance.png" mimeType="image/png" encoding="base64"␣
```

(continues on next page)

(continued from previous page)

```
↪schema=""/>
                </wps:Output>
        </wps:ProcessOutputs>
</wps:ExecuteResponse>
```

Load metalink document

Replace the metalink output URL.

```
metalink_url = ''
```

[19]:
```
metalink_url = 'http://rook4.cloud.dkrz.de:80/outputs/rook/bc97d460-87f2-11eb-b8ed-
↪fa163e1098db/input.meta4'
```

[20]:
```
print(requests.get(metalink_url).text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<metalink xmlns="urn:ietf:params:xml:ns:metalink">
    <published>2021-03-18T15:03:43Z</published>
    <generator>PyWPS/4.4.0</generator>

    <file name="rlds_Amon_IPSL-CM6A-LR_historical_r1i1p1f1_gr_19850116-20141216.nc">
        <identity>NetCDF file</identity>
        <size>20313784</size>
        <metaurl mediatype="application/x-netcdf">http://rook4.cloud.dkrz.de:80/outputs/
↪rook/bfe9ffee-87f2-11eb-a863-fa163e1098db/rlds_Amon_IPSL-CM6A-LR_historical_r1i1p1f1_
↪gr_19850116-20141216.nc</metaurl>
        <publisher name="None" url="http://rook4.cloud.dkrz.de:80/wps"/>
    </file>

</metalink>
```

Download netCDF output

Replace the download URL.

```
download_url = ''
```

[21]:
```
download_url = 'http://rook4.cloud.dkrz.de:80/outputs/rook/bfe9ffee-87f2-11eb-a863-
↪fa163e1098db/rlds_Amon_IPSL-CM6A-LR_historical_r1i1p1f1_gr_19850116-20141216.nc</
↪metaurl'
```

[22]:
```
print(download_url)
```

```
http://rook4.cloud.dkrz.de:80/outputs/rook/bfe9ffee-87f2-11eb-a863-fa163e1098db/rlds_
↪Amon_IPSL-CM6A-LR_historical_r1i1p1f1_gr_19850116-20141216.nc</metaurl
```

Execute subset (POST, sync)

See WPS examples: http://schemas.opengis.net/wps/1.0.0/examples/

[23]:
```
xml = """<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"␣
↪xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"␣
↪xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
↪opengis.net/wps/1.0.0
../wpsExecute_request.xsd">
    <ows:Identifier>subset</ows:Identifier>
    <wps:DataInputs>
            <wps:Input>
                    <ows:Identifier>collection</ows:Identifier>
                    <wps:Data>
                            <wps:LiteralData>c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.
↪r1i1p1f1.Amon.rlds.gr.v20180803</wps:LiteralData>
                    </wps:Data>
        </wps:Input>
        <wps:Input>
                    <ows:Identifier>time</ows:Identifier>
                    <wps:Data>
                            <wps:LiteralData>1860-01-01/1900-12-30</wps:LiteralData>
                    </wps:Data>
            </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
            <wps:ResponseDocument storeExecuteResponse="false" status="false">
                    <wps:Output asReference="true">
                            <ows:Identifier>output</ows:Identifier>
                    </wps:Output>
            </wps:ResponseDocument>
    </wps:ResponseForm>
</wps:Execute>
"""
```

[24]:
```
resp = requests.post(url, data=xml)
resp.ok
```

[24]:
```
True
```

[25]:
```
print(resp.text)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.
↪opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.
↪w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 ..
↪/wpsExecute_response.xsd" service="WPS" version="1.0.0" xml:lang="en-US"␣
↪serviceInstance="http://rook4.cloud.dkrz.de:80/wps?request=GetCapabilities&amp;amp;
↪service=WPS" statusLocation="">
    <wps:Process wps:processVersion="1.0">
        <ows:Identifier>subset</ows:Identifier>
        <ows:Title>Subset</ows:Title>
        <ows:Abstract>Run subsetting on climate model data. Calls daops operators.</ows:
```

(continues on next page)

(continued from previous page)

```
↪Abstract>
        </wps:Process>
    <wps:Status creationTime="2021-03-18T15:08:42Z">
        <wps:ProcessFailed>
            <wps:ExceptionReport>
                    <ows:Exception exceptionCode="NoApplicableCode" locator="None">
                            <ows:ExceptionText>Process error: Some or all of the␣
↪requested collection are not in the list of available data.</ows:ExceptionText>
                    </ows:Exception>
            </wps:ExceptionReport>
        </wps:ProcessFailed>
        </wps:Status>
</wps:ExecuteResponse>
```

2.4.2 Run subset by area operation

Rooki calls climate data operations on the **rook** processing service.

[ ]:
```
import os
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'

from rooki import rooki
```

parameters of subset operation

[ ]:
```
rooki.subset?
```

run subset by area

[ ]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1860-01-01/1980-12-30',
    area='0.,49.,10.,65'
)
resp.ok
```

show metalink output

[ ]:
```
resp.url
```

[ ]:
```
print(resp.xml)
```

Size in MBytes

[ ]:
```
resp.size_in_mb
```

URLs in metalink document ...

[ ]:
```
resp.download_urls()
```

download files ...

```
[ ]: resp.download()
```

... and open with xarray

```
[ ]: dsets = resp.datasets()
```

```
[ ]: ds = dsets[0]
ds
```

```
[ ]: ds.attrs
```

2.4.3 Run subset by time operation

Rooki calls climate data operations on the **rook** processing service.

```
[ ]: import os
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'

from rooki import rooki
```

parameters of subset operation

```
[ ]: rooki.subset?
```

data inventory

https://github.com/cp4cds/c3s_34g_manifests/tree/master/inventories

using: https://github.com/cp4cds/c3s_34g_manifests/blob/master/inventories/c3s-cmip6/c3s-cmip6_v20210126.yml

run subset

```
[ ]: resp = rooki.subset(
    collection='c3s-cmip6.ScenarioMIP.INM.INM-CM5-0.ssp245.r1i1p1f1.day.tas.gr1.v20190619
↪',
    time='2016-01-01/2016-12-30',
)
resp.ok
```

show metalink output

```
[ ]: resp.url
```

```
[ ]: print(resp.xml)
```

Size in MBytes

```
[ ]: resp.size_in_mb
```

URLs in metalink document ...

```
[ ]: resp.download_urls()
```

download files ...

[ ]:
```
resp.download()
```

... and open with xarray

[ ]:
```
dsets = resp.datasets()
```

[ ]:
```
ds = dsets[0]
ds
```

[ ]:
```
ds.attrs
```

provenance

[ ]:
```
prov_plot_url = resp.provenance_image()
prov_plot_url
```

[ ]:
```
from IPython.display import Image
Image(prov_plot_url)
```

2.4.4 Advanced Rooki Usage

Use enviroment to change rooki config

[ ]:
```
import os
from rooki import rooki, reinit
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'
# os.environ['ROOK_URL'] = 'http://localhost:5000/wps'
# mode: sync or async
# os.environ['ROOK_MODE'] = 'async'
```

[ ]:
```
# change default download folder
os.environ['ROOKI_OUTPUT_DIR'] = '/tmp/rooki'
```

[ ]:
```
# HINT: re-init rooki!
reinit()
rooki.url
```

[ ]:
```
rooki.output_dir
```

[ ]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1985-01-01/2014-12-30')
resp.ok
```

[ ]:
```
# number of files to download
resp.num_files
```

[ ]:
```
# total size of all files in bytes
resp.size
```

[ ]:
```
resp.size_in_mb
```

[ ]:
```
resp.download_urls()
```

[ ]:
```
files = resp.download()
```

Use Rooki client

[ ]:
```
from rooki.client import Rooki
url='http://rook.dkrz.de/wps'
# url='http://localhost:5000/wps'

rooki = Rooki(url, mode='async', output_dir='/tmp/rooki')
rooki.url
```

[ ]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1985-01-01/2014-12-30')
resp.ok
```

[ ]:
```
# total size
resp.size_in_mb
```

[ ]:
```
# download files
files = resp.download()
```

[ ]:
```
files[0]
```

[ ]:
```
# open as xarray dataset
dsets = resp.datasets()
```

[ ]:
```
ds = dsets[0]
ds
```

2.4.5 Show exceptions

[1]:
```
import os
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'

from rooki import rooki
```

check that subset operator is working

[2]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1985-01-01/2014-12-30',
)
resp.ok
```

[2]:
```
True
```

Error: missing collection parameter

[3]:
```
try:
    resp = rooki.subset()
except TypeError as e:
    print(f"{e}")
```

```
subset() missing 1 required positional argument: 'collection'
```

Check which time range is available

[4]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
)
resp.ok
```

[4]:
```
True
```

[5]:
```
resp.download_urls()
```

[5]:
```
['https://data.mips.copernicus-climate.eu/thredds/fileServer/esg_c3s-cmip6/CMIP/IPSL/
↪IPSL-CM6A-LR/historical/r1i1p1f1/Amon/rlds/gr/v20180803/rlds_Amon_IPSL-CM6A-LR_
↪historical_r1i1p1f1_gr_185001-201412.nc']
```

Error: not available time range

[6]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='2100-01-01/2200-12-30',
)
resp.ok
```

```
 owslib.wps.WPSException : {'code': 'NoApplicableCode', 'locator': 'None', 'text':
↪'Process error: No files found in given time range for c3s-cmip6.CMIP.IPSL.IPSL-CM6A-
↪LR.historical.r1i1p1f1.Amon.rlds.gr.v20180803'}
```

[6]:
```
False
```

[7]:
```
resp.status
```

[7]:
```
'Process error: No files found in given time range for c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.
↪historical.r1i1p1f1.Amon.rlds.gr.v20180803'
```

Error: invalid time parameter

[8]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1900-01-01',
)
resp
```

```
 owslib.wps.WPSException : {'code': 'NoApplicableCode', 'locator': 'None', 'text':
↪'Process error: TimeParameter should be passed in as a range separated by /'}
```

[8]:
```
Process error: TimeParameter should be passed in as a range separated by /
```

[9]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='today',
)
resp
```

```
 owslib.wps.WPSException : {'code': 'NoApplicableCode', 'locator': 'None', 'text':
↪'Process error: TimeParameter should be passed in as a range separated by /'}
```

[9]:
```
Process error: TimeParameter should be passed in as a range separated by /
```

Error: not available collection … c3s-cmip7

[10]:
```
resp = rooki.subset(
    collection='c3s-cmip7.output1.MOHC.HadGEM2-ES.rcp85.mon.atmos.Amon.r1i1p1.latest.tas
↪',
    time='2085-01-01/2120-12-30',
)
resp
```

```
 owslib.wps.WPSException : {'code': 'NoApplicableCode', 'locator': 'None', 'text':
↪'Process error: The project could not be identified and force was set to false'}
```

[10]:
```
Process error: The project could not be identified and force was set to false
```

Error: invalid collection parameter

[11]:
```
resp = rooki.subset(
    collection='c3s-cmip5/tas',
    time='2085-01-01/2120-12-30',
)
resp
```

```
 owslib.wps.WPSException : {'code': 'NoApplicableCode', 'locator': 'None', 'text':
↪'Process error: The project could not be identified and force was set to false'}
```

[11]:
```
Process error: The project could not be identified and force was set to false
```

Error: operation failed ...0 meridian not supported

Update: this is solved!

Issue: https://github.com/roocs/clisops/issues/35

[12]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.IPSL.IPSL-CM6A-LR.historical.r1i1p1f1.Amon.rlds.gr.
↪v20180803',
    time='1901-01-01/1921-12-30',
    area='-20, 40, 20, 70',
)
resp
```

[12]:
```
Metalink URL: http://rook1.cloud.dkrz.de:80/outputs/rook/a7b09ae6-76b3-11eb-94aa-
↪fa163eac7aff/input.meta4, num files: 1
```

2.4.6 Test subset Operation

[ ]:
```
import os
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'

from rooki import rooki
```

[ ]:
```
resp = rooki.subset(
    collection='c3s-cmip6.CMIP.INM.INM-CM5-0.historical.r1i1p1f1.Amon.rlds.gr1.v20190610
↪',
    time='1900-01-01/1900-12-30',
)
assert resp.ok
```

[ ]:
```
assert 'rlds_Amon_INM-CM5-0_historical_r1i1p1f1_gr1_19000116-19001216.nc' in resp.
↪download_urls()[0]
```

2.4.7 Test rook workflow with subset chain

[ ]:
```
import os
os.environ['ROOK_URL'] = 'http://rook.dkrz.de/wps'

from rooki import operators as ops
```

[ ]:
```
wf = ops.Subset(
        ops.Subset(
            ops.Input(
                'rlds', ['c3s-cmip6.CMIP.INM.INM-CM5-0.historical.r1i1p1f1.Amon.rlds.gr1.
↪v20190610']
            ),
            time="1890-01-01/1920-12-30",
```

(continues on next page)

(continued from previous page)

```
        ),
        time="1900-01-01/1900-12-30",
)
```

[ ]:
```
resp = wf.orchestrate()
assert resp.ok
```

[ ]:
```
assert 'rlds_Amon_INM-CM5-0_historical_r1i1p1f1_gr1_19000116-19001216.nc' in resp.
↪download_urls()[0]
```

2.5 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

2.5.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/roocs/rooki/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

rooki could always use more documentation, whether as part of the official rooki docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/roocs/rooki/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Read the *Development Guide* to set up `rooki` for local development.

2.6 Version History

2.6.1 v0.5.0 (2022-09-28)

New Features

- Added operator and notebook for *Concat* demo (#104).

Changes

- Updated notebooks (subset, cmip6-decadal, intake).

2.6.2 v0.4.0 (2022-04-14)

New Features

- Added operator for *AverageByTime* (#93, #96).

Changes

- Added notebooks for CMIP6 decadal (#89, #91).
- Added notebooks for "subset by point" (#87).
- Updated notebooks for c4i demo (#86, #94, #95).
- Updated notebooks for "average" operator (#93, #96).

2.6.3 v0.3.3 (2021-08-12)

New Features

- Use reinit internally to update config from environment variables … e.g. update access token (#81).
- Added wps lineage option (#80).
- Using environment variable ACCESS_TOKEN for OAuth access token (#80).

Changes

- Updated notebooks for c4i and dashboard demo.

2.6.4 v0.3.2 (2021-03-21)

Changes

Notebooks:

- Added tests (#55, #58, #59)
- Added c4i demo (#54).
- Added intake example (#56).

Bug Fixes

- Quick fix for missing cancel function (#57).
- Allow metalink download from unverified https end-point (#52).

2.6.5 v0.3.1 (2021-02-24)

Changes

- Updated notebooks (#45, #46, #47, #48).
- Updated requirements (birdy>=0.7.0).

2.6.6 v0.3.0 (2020-12-18)

New Features

- Configure output folder for metalink downloads (#41).
- Access provenance document (#38).
- Added provenance notebook (#39).
- Added test notebook with execution time measure (#40).

2.6.7 v0.2.3 (2020-11-05)

New Features

- Allow Python 3.6 (#36)
- Run travis tests on multiple Python versions >= 3.6.
- Run doc build test on travis.

2.6.8 v0.2.2 (2020-11-02)

Bug Fixes

- Using pymetalink package from pypi (#34).

2.6.9 v0.2.1 (2020-10-28)

Bug Fixes

- Fixed pymetalink requirement (#33).

2.6.10 v0.2.0 (2020-10-26)

New Features

- Lightweight wrapper for birdy WPS client.
- Operators to build workflow.
- Configuration to overwrite default settings.
- Result object to access MetaLink outputs.
- Notebooks with usage examples.

2.6.11 v0.1.0 (2020-03-19)

- First release.