



A COMMON, OPEN SOURCE INTERFACE
BETWEEN EARTH OBSERVATION DATA
INFRASTRUCTURES AND FRONT-END
APPLICATIONS

Deliverable 04

Version 1.0 from 2018/03/26

openEO core API prototype including Proof of Concept



Change history

| Issue | Date | Author(s) | Description |
|-------|------------|---|--|
| 0.1 | 2018/03/09 | Edzer Pebesma, Matthias Mohr, WWU | First draft |
| 0.2 | 2018/03/20 | Jeroen Dries, VITO | Internal review |
| 1.0 | 2018/03/26 | Matthias Schramm, TU Wien | Final review and creation of final version |

For any clarifications please contact openEO@list.tuwien.ac.at.

Number of pages: **8**

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776242. Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

© openEO Consortium, 2018

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.



Table of Contents

- 1 Executive summary** **6**
- 2 Links to the demonstrator** **6**
 - 2.1 Link to the API specification, documentation, and proof-of-concept 6
 - 2.2 Selected use-cases 6
 - 2.3 Links to blog post and videos 7
 - 2.4 Videos demonstrating client-back-end interactions 7
- 3 Succeeded use case/client/back-end combination** **7**
- 4 Discussion** **8**
- 5 References** **8**

List of Tables

3.1 Combinations of clients, back-ends and use cases that were successfully tested
(numbers refer to use case 1-3) 8

List of Acronyms

API Application Programming Interface

EO Earth Observation

NDVI Normalized Difference Vegetation Index



1 Executive summary

The openEO EU project [1] develops an open API to connect R, python and JavaScript clients to big Earth observation cloud back-ends in a simple and unified way. This deliverable (of type “demonstrator”) demonstrates the existence of this API and its functioning for a selected number of use cases, by pointing to the API documentation and to functioning software implementation where an R, Python and JavaScript client connect to a variety of back-ends. The purpose of the proof-of-concept is to inform the Earth Observation community about the progress made in openEO by ways of the proof of concept, and this document contains the links to the on-line, public documentation created (API, descriptions, blog post with demonstration videos); it does not copy those.

The API description, the description of the use cases, the client and back-end drivers developed and the demonstrations showing that they actually work well together all indicate that the openEO API is a feasible concept.

2 Links to the demonstrator

2.1 Link to the API specification, documentation, and proof-of-concept

The core concepts and API specification for openEO are found at <https://open-eo.github.io/openeo-api/index.html>.

The proof-of-concept is described at: <https://open-eo.github.io/openeo-api/poc/index.html>.

2.2 Selected use-cases

To demonstrate the feasibility of the openEO concept, a prototype has been developed that focuses on using clients for analysing Earth Observation imagery in cloud-based back-ends. Features like user authentication or accounting have, on purpose, not been addressed in this proof of concept.

During the openEO kick-off meeting in Oct 2017, three use cases were selected to demonstrate the openEO proof-of-concept. They include

1. Deriving minimum NDVI measurements over pixel time series of Sentinel 2 imagery
2. Creating a monthly aggregated Sentinel 1 product from a custom Python script
3. Computing time series of zonal (regional) statistics of Sentinel 2 imagery over user-uploaded polygons

All case studies follow the pattern:

- Check whether data is available at the back-end
- Check that needed processes are available
- Create a job at the back-end
- Define output



- Retrieve output
- Stop the job (and the service)

In the documentation, found at <https://open-eo.github.io/openeo-api/poc/index.html>, the example API calls with (example) requests and responses are given.

2.3 Links to blog post and videos

The following blog post on the <http://openeo.org/> project home page describes the proof of concept use cases to a wide audience:

<http://openeo.org/openeo/news/2018/03/17/poc.html>

2.4 Videos demonstrating client-back-end interactions

The following videos were created to demonstrate successful client-server interactions (also linked from the blog post):

1. GRASS GIS (mundialis) accessed with curl (UC 1,2,3): [view](#)¹
2. WCPS (EURAC) accessed with R client (UC1): [view](#)²
3. R back-end (WWU) accessed with R client (UC1): [view](#)³
4. R back-end (WWU) accessed with R client (UC3): [view](#)⁴
5. Sentinel-Hub (Sinergise), OpenShift (EODC) and WCPS (EURAC) accessed with the Web Editor, based on the JavaScript client (UC1): [view](#)⁵
6. R back-end (WWU) accessed with the Web Editor, based on the JavaScript client (UC3): [view](#)⁶
7. GeoPySpark (VITO) accessed with the Python client (UC1): [view](#)⁷

3 Succeeded use case/client/back-end combination

The combinations of clients, back-ends and use cases that were successfully executed are listed in table 3.1

¹<https://www.youtube.com/watch?v=NgF1WgCtSiM>

²<https://www.youtube.com/watch?v=NoD0nVGM3ww>

³https://www.youtube.com/watch?v=Yb_QfIO-uIE

⁴https://www.youtube.com/watch?v=LYnad6KC_CU

⁵<https://youtu.be/zDaQkw0NhpY>

⁶<https://youtu.be/XsPbKypUuIE>

⁷<https://www.youtube.com/watch?v=qtItp9OC0qHY>

| back-end | Python client | R client | JavaScript client |
|-------------------------|---------------|----------|-------------------|
| GRASS GIS, mundialis | | 1,2,3 | |
| OpenShift, EODC | | 1 | 1 |
| GeoPySpark, VITO | 1,2,3 | | |
| Sentinel Hub, Sinergise | | | 1 |
| WCPS, EURAC | 1 | 1 | 1 |
| R back-end, WWU | | 1,2,3 | 1,3 |

Table 3.1: Combinations of clients, back-ends and use cases that were successfully tested (numbers refer to use case 1-3)

4 Discussion

The API description, the description of the use cases, the client and back-end drivers developed and the demonstrations showing that they actually work well together all indicate that the openEO API is a feasible concept. The amount of working dedicated openEO clients (3) and back-ends (6) exceeds by far the tentative goals that were set in the project proposal for this early stage of the project.

By intention, the use-cases were picked ambitiously: they address variation of multiple attributes (bands) over space and time, user-defined functions in R and Python that need to be evaluated on the back-end, and integration of imagery over polygons, and delivering results in various forms (file formats, web services). It was clear from the start that not all back-ends would be able (or will ever be able) to cope with all of them. This has led to good discussions about the differences between back-ends, and has anticipated upcoming decisions on what can become compulsory and what will have to become optional parts of the openEO API.

5 References

- [1] E. Pebesma, W. Wagner, M. Schramm, A. Von Beringe, C. Paulik, M. Neteler, J. Reiche, J. Verbesselt, J. Dries, E. Goor, and et al., “openEO – a common, open source interface between earth observation data infrastructures and front-end applications,” *Zenodo*, Nov 2017. [Online]. Available: <https://zenodo.org/record/1065474>

