5 OpenLandMap STAC

On this page

5 OpenLandMap STAC

5.1 Listing layers

5.2 Spatial overlay

> You are reading the work-in-progress OpenLandMap.org Open Land Data Services. This chapter is currently draft version, a peer-review publication is pending. You can find the polished first edition at https://openlandmap.github.io/book/.

5.1 Listing layers

Thanks to the STAC functionality and `rstac` package, it is possible to query directly which collections are available on the stac.OpenLandMap.org (Note: some layers that are available in STAC, might not be available in the front-end / web-GIS):

```
library(rstac)
olm <- stac_read("http://s3.eu-central-1.wasabisys.com/stac/openlandmap/catalog.json")
olm
#> ###Catalog
#> - id: openlandmap
#> - description:
#> Spatio-Temporal Asset Catalog for global layers provided by [OpenLandMap](https://openlandmap.org) and maintaned by
[OpenGeoHub Foundation](https://opengeohub.org)
#> - field(s): type, id, stac_version, description, links, title
```

To enumerate all available collections in the OpenLandMap catalog, we can scrutinize the links entry:

```
links(olm)
#> ###Links
#> - links (74 entries(s)):
#>  1 OpenLandMap STAC (./catalog.json)
#>  2 Sentinel-5P monthly tropospheric nitrogen dioxide density
#> (./no2_s5p.l3.trop.tmwm/collection.json)
#>  3 Sentinel-5P long-term tropospheric nitrogen dioxide density
#> (./no2_s5p.l3.trop.tmwm.ltm/collection.json)
#>  4 OpenLandMap annual soil organic carbon
#> (./log.oc_iso.10694/collection.json)
#>  5 MOD13Q1 long-term Enhanced Vegetation Index (EVI - trend analysis)
#> (./evi_mod13q1.stl.trend.logit.ols.beta/collection.json)
#>  6 ESA CCI annual land cover
#> (./land.cover_esacci.lc.l4/collection.json)
#>  7 MOD13Q1 bi-monthly Enhanced Vegetation Index Index (EVI)
#> (./evi_mod13q1.tmwm.inpaint/collection.json)
#>  8 OpenLandMap ensemble digital terrain model
#> (./dtm.bareearth_ensemble/collection.json)
#>  9
#> Monthly fraction of absorbed photosynthetically active radiation (FAPAR)
#> (./fapar_essd.lstm/collection.json)
#> 10
#> Long-term fraction of absorbed photosynthetically active radiation (FAPAR - trend analysis)
#> (./fapar_essd.lstm.p95.beta/collection.json)
#>   ... with 64 more link(s).
```

For instance, to compile a list of layers with the `title` containing the text `"GLC"` for land cover annual time-series, we can use:

```
links(olm, grepl("GLC", title))
#> ###Links
#> - links (3 entries(s)):
#> 1 UMD GLAD annual land cover and land use (GLCLUC)
#> (./lc_glad.glcluc/collection.json)
#> 2 UMD GLAD annual land cover and land use change (GLCLUC)
#> (./lc_glad.glcluc.change/collection.json)
#> 3 GLC_FCS30D annual land land-cover dynamic monitoring product
#> (./lc_glc.fcs30d/collection.json)
```

Let's explore the third link, referencing the `GLC_FCS30D` annual land-cover dynamic monitoring product:

```r
glc_link <- links(olm, grepl("GLC", title))[[3]]
glc <- link_open(glc_link)
glc
#> ###Collection
#> - id: lc_glc.fcs30d
#> - title: GLC_FCS30D annual land land-cover dynamic monitoring product
#> - description:
#> GLC_FCS30D is the first global fine land cover dynamic product at a 30-meter resolution that adopts continuous change detection. It utilizes a refined classification system containing 35 land-cover categories and covers the time span from 1985 to 2022. Before the year 2000, the update cycle was every 5 years, while after 2000, it is updated annually. In specific, it developed by combining the continuous change detection method, local adaptive updating models and the spatiotemporal optimization algorithm from dense time-series Landsat imagery, and was validated to achieve an overall accuracy of 80.88% (±0.27%) for the basic classification system 10 major land-cover types) and 73.24% (±0.30%) for the LCCS level-1 validation system (17 LCCS land-cover types).
#> - field(s):
#> type, id, stac_version, description, links, stac_extensions, Theme, version, doi, layer_unit, contact_name, contact_email, date_offset, sld_url, qml_url, title, extent, license, keywords, providers
```

Now, let's list its available items by filtering links with the `rel == "item"` attribute:

```r
links(glc, rel == "item")
#> ###Links
#> - links (26 entries(s)):
#>  1 [item]
#> (./lc_glc.fcs30d_19850101_19851231/lc_glc.fcs30d_19850101_19851231.json)
#>  2 [item]
#> (./lc_glc.fcs30d_19900101_19901231/lc_glc.fcs30d_19900101_19901231.json)
#>  3 [item]
#> (./lc_glc.fcs30d_19950101_19951231/lc_glc.fcs30d_19950101_19951231.json)
#>  4 [item]
#> (./lc_glc.fcs30d_20000101_20001231/lc_glc.fcs30d_20000101_20001231.json)
#>  5 [item]
#> (./lc_glc.fcs30d_20010101_20011231/lc_glc.fcs30d_20010101_20011231.json)
#>  6 [item]
#> (./lc_glc.fcs30d_20020101_20021231/lc_glc.fcs30d_20020101_20021231.json)
#>  7 [item]
#> (./lc_glc.fcs30d_20030101_20031231/lc_glc.fcs30d_20030101_20031231.json)
#>  8 [item]
#> (./lc_glc.fcs30d_20040101_20041231/lc_glc.fcs30d_20040101_20041231.json)
#>  9 [item]
#> (./lc_glc.fcs30d_20050101_20051231/lc_glc.fcs30d_20050101_20051231.json)
#> 10 [item]
#> (./lc_glc.fcs30d_20060101_20061231/lc_glc.fcs30d_20060101_20061231.json)
#>   ... with 16 more link(s).
```

To be able to filter items based on spatial and temporal attributes, we need to open them:

```r
glc_items <- read_items(glc, progress = FALSE)
glc_items
#> ###Items
#> - features (26 item(s)):
#>   - lc_glc.fcs30d_19850101_19851231
#>   - lc_glc.fcs30d_19900101_19901231
#>   - lc_glc.fcs30d_19950101_19951231
#>   - lc_glc.fcs30d_20000101_20001231
#>   - lc_glc.fcs30d_20010101_20011231
#>   - lc_glc.fcs30d_20020101_20021231
#>   - lc_glc.fcs30d_20030101_20031231
#>   - lc_glc.fcs30d_20040101_20041231
#>   - lc_glc.fcs30d_20050101_20051231
#>   - lc_glc.fcs30d_20060101_20061231
#>   - ... with 16 more feature(s).
#> - assets: lc_glc.fcs30d_c_30m_s, qml, sld, thumbnail
#> - item's fields:
#> assets, bbox, collection, geometry, id, links, properties, stac_extensions, stac_version, type
```

We have items for all dates:

```r
items_datetime(glc_items)
#>  [1] "1985-01-01T00:00:00Z" "1990-01-01T00:00:00Z" "1995-01-01T00:00:00Z"
#>  [4] "2000-01-01T00:00:00Z" "2001-01-01T00:00:00Z" "2002-01-01T00:00:00Z"
#>  [7] "2003-01-01T00:00:00Z" "2004-01-01T00:00:00Z" "2005-01-01T00:00:00Z"
#> [10] "2006-01-01T00:00:00Z" "2007-01-01T00:00:00Z" "2008-01-01T00:00:00Z"
#> [13] "2009-01-01T00:00:00Z" "2010-01-01T00:00:00Z" "2011-01-01T00:00:00Z"
#> [16] "2012-01-01T00:00:00Z" "2013-01-01T00:00:00Z" "2014-01-01T00:00:00Z"
#> [19] "2015-01-01T00:00:00Z" "2016-01-01T00:00:00Z" "2017-01-01T00:00:00Z"
#> [22] "2018-01-01T00:00:00Z" "2019-01-01T00:00:00Z" "2020-01-01T00:00:00Z"
#> [25] "2021-01-01T00:00:00Z" "2022-01-01T00:00:00Z"
```

To enumerate all available assets across these items, we can run:

```r
items_assets(glc_items)
#> [1] "lc_glc.fcs30d_c_30m_s" "qml"                   "sld"
#> [4] "thumbnail"
```

5.2 Spatial overlay

For overlaying multiple new points with COGs, we can leverage the `rstac` function `assets_url()` to retrieve the URLs of all COG files. These URLs can then be passed to an extraction function:

```
urls <- assets_url(glc_items, asset_names = "lc_glc.fcs30d_c_30m_s", append_gdalvsi = TRUE)
urls[1:3]
#> [1] "/vsicurl/https://s3.openlandmap.org/arco/lc_glc.fcs30d_c_30m_s_19850101_19851231_go_epsg.4326_v20231026.tif"
#> [2] "/vsicurl/https://s3.openlandmap.org/arco/lc_glc.fcs30d_c_30m_s_19900101_19901231_go_epsg.4326_v20231026.tif"
#> [3] "/vsicurl/https://s3.openlandmap.org/arco/lc_glc.fcs30d_c_30m_s_19950101_19951231_go_epsg.4326_v20231026.tif"
```

Let's define an extracting function. This function can extract the values in parallel:

```
extract_xy = function(lon, lat, cogs, mc.cores = 10) {
  values = parallel::mclapply(cogs, function(i) {
    point <- terra::vect(matrix(c(lon, lat), ncol = 2), crs = "EPSG:4326")
    value <- terra::extract(terra::rast(i), point)
    #dplyr::as_tibble(value)[,2]
    value[,2]
  }, mc.cores = mc.cores)
  values = dplyr::tibble(glc = unlist(values))
  return(values)
}
```

This only needs URL address of the COGs on some S3 storage and then longitude and latitude of the query points (in the WGS84 system). This is an example of query of all land cover classes from 1985 to 2022:

```
values <- extract_xy(-35.5, -9.0, urls)
# add date column
values$date <- as.Date(items_datetime(glc_items))
values
#> # A tibble: 26 × 2
#>     glc date
#>   <int> <date>
#> 1   130 1985-01-01
#> 2   130 1990-01-01
#> 3   130 1995-01-01
#> 4    10 2000-01-01
#> 5    10 2001-01-01
#> 6    10 2002-01-01
#> # ℹ 20 more rows
```

« 4 OpenLandMap layers

6 References »