

API Reference

# RS Native API

API Version: 1.12.0-rc1

The Native API of the Copernicus Reference System (COPRS) that can be used to query and download product data.

# INDEX

<b>1. STAC INTERFACE</b>	<b>3</b>
1.1 GET /stac	3
1.2 GET /stac/{missionId}	3
1.3 GET /stac/{missionId}/collections	4
1.4 GET /stac/{missionId}/collections/{productType}	4
1.5 GET /stac/search	5

# API

## 1. STAC INTERFACE

STAC interface to query items available on the RS

### 1.1 GET /stac

Landing page for the STAC interface

#### REQUEST

No request parameters

#### RESPONSE

STATUS CODE - 200: Returns landing page of the STAC interface

RESPONSE MODEL - application/json

```
{
  stac_version      string
  collections       [string]
  description       string
  links [{
    Array of object:
      rel    string
      href   string
      type   string
      title  string
  }]
  id          string
  conformsTo  [string]
  type        string
  title       string
  stac_extensions [string]
}
```

### 1.2 GET /stac/{missionId}

Retrieve SubCatalog for specific mission

#### REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*missionId	enum ALLOWED: S1, S2, S3	Mission identifier to specify which Sentinel mission shall be queried.

#### RESPONSE

STATUS CODE - 200: Returns STAC conform catalog containing further links

RESPONSE MODEL - application/json

```
{
  stac_version      string
  collections       [string]
  description       string
  links [{
    Array of object:
      rel    string
      href   string
      type   string
      title  string
    }]
  id           string
  type         string
  title        string
  stac_extensions [string]
}
```

1.3 GET /stac/{missionId}/collections

Retrieve list of collections for mission

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*missionId	enum ALLOWED: S1, S2, S3	Mission identifier to specify which Sentinel mission shall be queried.

RESPONSE

STATUS CODE - 200: Returns list of collections available for this mission. Collections map to product type.

RESPONSE MODEL - application/json

```
{
  stac_version      string
  collections       [string]
  description       string
  links [{
    Array of object:
      rel    string
      href   string
      type   string
      title  string
    }]
  id           string
  type         string
  title        string
  stac_extensions [string]
}
```

1.4 GET /stac/{missionId}/collections/{productType}

## Retrieve information about a specific collection

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*missionId	enum ALLOWED: S1, S2, S3	Mission identifier to specify which Sentinel mission shall be queried.
*productType	string	Product type to get the items of.

### RESPONSE

STATUS CODE - 200: Collection object for a specific collection

#### RESPONSE MODEL - application/json

```
{
  extent          undefined
  stac_version     string
  keywords         [string]
  description      string
  type             string
  title            string
  license          string
  assets {
  }
  links [{
    Array of object:
      rel    string
      href   string
      type   string
      title  string
  }]
  id         string
  stac_extensions [string]
  providers  [string]
  summaries {
  }
}
```

## 1.5 GET /stac/search

### Search STAC items with simple filtering.

Retrieve Items matching filters. Intended as a shorthand API for simple queries.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
ids	string	UUID value for a specific item. Example: * "808dc636-5bd6-43d9-ad52-3b2b589c2d80"

NAME	TYPE	DESCRIPTION
bbox	string	4 points (longitude, latitude) describing a polygon. Longitude and latitude have to be separated by an URL encoded space ("%20"), points have to be separated by a comma. Items with intersecting footprints will be returned. Example: * "76.036377%20-75.654331,76.57106%20-75.654331,76.57196%20-75.594357,76.036377%20-75.594357"
point	string	1 point (logitude, latitude). Longitude and latitude have to be separated by an URL encoded space ("%20"). Items with a footprint containing the point will be returned. Example: * 76.036377%20-75.654331
line	string	2 points (logitude, latitude) describing a line. Longitude and latitude have to be separated by an URL encoded space ("%20"), points have to be separated by a comma. Items with a footprint intersecting the line will be returned. Example: * 76.036377%20-75.654331,76.57106%20-75.654331
productname	string	Part of the productname that should be included in the retrieved items. Example: * S1A_EW_RAW__ONDH
collections	string	Exact name of the product type which items shall be queried for. Example: * OL_1_EFR__
cloudcover	string	Percentage value interval in which the items are filtered for. Has to be provided in the form min/max where empty values may be left out or replaced by "..". Examples: * Closed interval: "5.0/10.0" * Open interval (minimum open): "../10.0" * Open interval (maximum open): 90.0/..
datetime	string	Interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots.  Examples: * A closed interval: "2018-02-12T00:00:00Z/2018-03-18T12:31:12Z" * Open intervals: "2018-02-12T00:00:00Z/.." or "../2018-03-18T12:31:12Z"  Only features that have a temporal property of `ContentDate` that intersects the value of `datetime` are selected.
publicationdate	string	Interval, open or closed. Date and time expressions adhere to RFC 3339. Open intervals are expressed using double-dots.  Examples: * A closed interval: "2018-02-12T00:00:00Z/2018-03-18T12:31:12Z" * Open intervals: "2018-02-12T00:00:00Z/.." or "../2018-03-18T12:31:12Z"  Only features that have a temporal property of `CreationDate` that intersects the value of `datetime` are selected.
polarisation	string	Polarisation to filter for. Example: * DV
page	integer	Number of page used for pagination. First page is page number 1.
limit	integer	Maximum number of results that shall be retrieved. Too big values might be cut down to maximum value allowed by API.

## RESPONSE

**STATUS CODE - 200:** A feature collection.

### RESPONSE MODEL - application/geo+json

```
{
  A GeoJSON FeatureCollection augmented with foreign members that contain values relevant to a STAC entity
  type*          enum          ALLOWED:FeatureCollection
  features* [{
    Array of object: A GeoJSON Feature augmented with foreign members that contain values relevant to a STAC entity
    stac_version*      string
    stac_extensions
    ANY OF
    prop0
    string
    prop1
    string
    id*                string                Provider identifier, a unique ID.
    bbox*              [number]
    geometry*
    ONE OF
    OPTION 1{
      type*          enum          ALLOWED:Point
      coordinates*   [number]
    }
    OPTION 2{
      type*          enum          ALLOWED:MultiPoint
      coordinates*   [{
        Array of object:
      }]
    }
    OPTION 3{
      type*          enum          ALLOWED:LineString
      coordinates*   [{
        Array of object:
      }]
    }
    OPTION 4{
      type*          enum          ALLOWED:MultiLineString
      coordinates*   [{
        Array of object:
          ::props [number]
      }]
    }
    OPTION 5{
      type*          enum          ALLOWED:Polygon
      coordinates*   [{
        Array of object:
          ::props [number]
      }]
    }
    OPTION 6{
      type*          enum          ALLOWED:MultiPolygon
      coordinates*   [{
        Array of object:
          ::props [{
            Array of object:
```

```

    }}
  }}
}
OPTION 7{
  type*          enum          ALLOWED:GeometryCollection
  geometries*    [{recursive}] geometryGeoJSON
}
type*            enum          ALLOWED:Feature
                                   The GeoJSON type

links* [{
  Array of object:
    href* string  The location of the resource
    rel*  string  Relation type of the link
    type  string  The media type of the resource
    title string  Title of the resource
}]
properties* {
  provides the core metadata fields plus extensions

    datetime*      string      The searchable date and time of the assets, in UTC.
                                It is formatted according to [RFC 3339, section 5.6](https://tools.ietf.org/html/
                                rfc3339#section-5.6).
                                `null` is allowed, but requires `start_datetime` and `end_datetime` from common
                                metadata to be set.

    start_datetime  string      The searchable date and time of the assets, in UTC.
                                It is formatted according to [RFC 3339, section 5.6](https://tools.ietf.org/html/
                                rfc3339#section-5.6).
                                `null` is allowed, but requires `start_datetime` and `end_datetime` from common
                                metadata to be set.

    end_datetime    string      The searchable date and time of the assets, in UTC.
                                It is formatted according to [RFC 3339, section 5.6](https://tools.ietf.org/html/
                                rfc3339#section-5.6).
                                `null` is allowed, but requires `start_datetime` and `end_datetime` from common
                                metadata to be set.

    PublicationDate string
    EvictionDate    string
    Checksum {
      checksum object containing the checksum value for the product file

      Algorithm      string      the hash function used for the calculation of the checksum value
      Value          string      the checksum value for the product file
      ChecksumDate   string      PATTERN:YYYY-MM-DDThh:mm:ss.sssZ
                                the date and time the checksum was calculated
    }
    ContentDate {
      object containing the start and end dates of the product

      Start string  PATTERN:YYYY-MM-DDThh:mm:ss.sssZ
                                the start date and time of the product

      End   string  PATTERN:YYYY-MM-DDThh:mm:ss.sssZ
                                the end date and time of the product
    }
    Name          string
    ContentType    string
    ProductionType string
    Online         boolean
    ContentLength  integer
    AdditionalAttributes {
      additional attributes/values segmented by value type

      StringAttributes {
        attributes with values of type string
      }
      IntegerAttributes {
        attributes with values of type integer

```



```

    }
    DoubleAttributes {
      attributes with values of type number
    }
    DateTimeOffsetAttributes {
      attributes with date+time values of type string
    }
    BooleanAttributes {
      attributes with values of type boolean
    }
  }
}
assets* {
}
}]
}

```

**STATUS CODE - default:** An error occurred.

**RESPONSE MODEL - application/json**

```

{
  Information about the exception: an error code plus an optional description.
  code*      string
  description string
}

```

---