Usage

- Deprecated

This site explains the different usage scenarios. The API basically can be accessed through two methods: Either from the user interface of an OpenAtlas application or, if the settings will allow it, from another application.

Please refer to the SwaggerHub documentation: https://app.swaggerhub.com/apis-docs/ctot-nondef/OpenAtlas/0.1

Paths

Limited parameters available:

- Request one entity: /api/0.1/entity/105
- Download a JSON of one entity: /api/0.1/entity/download/105
- Request list of entities of one node: /api/0.1/node_entities/26294
- Request list of entities of one node and subnodes: /api/0.1/node_entities_all/26294
- Request list of subunits of one place: /api/0.1/subunit/50505
- Request list of all subunits of one place: /api/0.1/subunit_hierarchy/50505
- Request content of the OA instance: /api/0.1/content (parameter: *lang*)
- URL for displaying images: /api/display/4215
    - The image will only be displayed if:
        1. A user is logged in
        2. API is set on public
        3. IP is on the whitelist

All Parameters available:

- Request by code: /api/0.1/code/actor
- Request by class: /api/0.1/class/E18
- Request by latest: /api/0.1/latest/16 (int = number of entries)

- Request one or more entities: /api/0.1/query?entities=115&entities=116
- Request one or more classes: /api/0.1/query?classes=E18&classes=E31
- Request one or more codes: /api/0.1/query?items=actor&items=source
- Combine requests: /api/0.1/query?items=actor&items=source&classes=E18&classes=E31&entities=115&entities=116

Parameter

The API paths also take in parameters.

path\parameter	sort	column	limit	filter	first	last	show	count	download	lang
entity							x		x	
entity/download							x			
node_entities								x	x	
node_entities_all								x	x	
subunit								x	x	
subunit_hierarchy								x	x	
content									x	x
display										

code	x	x	x	x	x	x	x	x	x	
class	x	x	x	x	x	x	x	x	x	
latest							x	x	x	
query	x	x	x	x	x	x	x	x	x	

Sort

```
<'asc', 'desc'>
```

The *sort* parameter controls the order of the results. These can be either ascending (asc) or descending (desc). To use this feature add the parameter:

```
?sort=<'asc','desc'>
```

Validation

If multiple *sort* parameter are used, the first valid sort input will be used.
It does not matter if the words are uppercase or lowercase (i.e. DeSc or aSC), but the query only takes **asc** or **desc** as valid input. If no valid input is provided, the result is orders ASC.

Column

```
<'id', 'class_code', 'name', 'description', 'created', 'modified', 'system_type', 'begin_from', 'b
egin_to', 'end_from', 'end_to'>
```

The *column* parameter declares which columns in the table are sorted with the *sort* parameter.

```
?column=<'id', 'class_code', 'name', 'description', 'created', 'modified', 'system_type', 'begin_f
rom', 'begin_to', 'end_from', 'end_to'>
```

Validation

If multiple *column* parameter are used, a list is created, by the order in which the parameters are given (i.e. ?column=name&column=description&column=id will order by name, description and id).
It does not matter if the words are uppercase or lowercase (i.e. Name, ID, DeScrIPtioN or Class_Code). If no valid input is provided, the results are ordered by name.

Limit

```
<number>
```

The *limit* parameter declares how many results will returned.

```
?limit=<number>
```

Validation

If multiple *limit* parameter are used, the first valid *limit* input will be used. Limit only take positive numbers.

Filtering

```
<=, !=, <, <=, >, >=, LIKE, IN, AND, OR, AND NOT, OR NOT>
```

The *filter* parameter is used to specify which entries should return.

```
?filter=<XXX>
```

Validation

Filter takes a lot of different parameters in various forms. *Filter* values need to be separated by |.

1. First value has to be a concatenation operator: 'and': 'AND', 'or': 'OR', 'onot': 'OR NOT', 'anot': 'AND NOT'.
2. Second value has to be the effected column: 'id', 'class_code', 'name', 'description', 'created', 'modified', 'system_type', 'begin_from', 'begin_to', 'end_from', 'end_to'.
3. Third value has to be a logical operator: 'eq': '=', 'ne': '!=', 'lt': '<', 'le': '<=', 'gt': '>', 'ge': '>=', 'like': 'LIKE',
4. Fourth value has to be the search term. If the 'in' operator is selected, then the search term has to be in brackets and separated by commas.

Please note, that the filter values will translate directly in SQL. For example:

```
?filter=and|name|like|Ach&filter=or|id|gt|5432
```

will result in

```
AND e.name LIKE %%Ach%% OR e.id > 5432
```

```
?filter=or|id|gt|150&filter=anot|id|ne|200
?filter=and|name|like|Ach
```

Pagination

```
first=<id> OR last=<id>
```

The *first* parameter takes ids and will show every entity after and including the named id.
The *last* parameter takes ids and will show every entity after the named id.

```
?first=<id>
```

```
?last=<id>
```

Validation

First and *last* take only one id, which is has to be a **number**. The table will be sorted AND filtered before the pagination comes in place.

```
?last=220
```

```
?first=219
```

Show/Hide Types

```
<'when', 'types', 'relations', 'names', 'links', 'geometry', 'depictions', 'none'>
```

The *show* parameter will take in the key values of a json. If no value is given, every key will be filled. If a value is given, it only will show the types which are committed. If the parameter contains *none*, no additional keys/values will be shown

```
?show=<'when', 'types', 'relations', 'names', 'links', 'geometry', 'depictions', 'none'>
```

Validation

For each value a new parameter has to be set. The value will be matched against a list of keywords, so wrong input will be ignored.

```
?show=when
?show=types
?show=types&show=when
?show=none
```

Count

```
<>
```

Returns a json with a number of the total count of the included entities.

```
?count
```

Validation

Only *count* will trigger the function. Count can have any numbers assigned to it, which makes no difference.

```
?count
```

Download

```
<>
```

Will trigger the download of the result of the request path.

```
?download
```

Validation

Only *download* will trigger the function. Download can have anything assigned to it, but this will be discarded.

```
?download
```

Lang (Language)

```
<'en', 'de'>
```

Only works with the */api/0.1/content* path. Select the language, which content will be displayed.

```
?lang
```

Validation

Default value is None, which means the default language of the OpenAtlas instance is taken.

```
?lang
?lang=en
?lang=DE
```