

API Whitepaper

- [API Authentication](#)
- [API Error Handling](#)
- [CORS](#)
- [Discussions](#)
- [JSON Templates](#)
- [Usage](#)
 - [Deprecated](#)

Issue: [#1050](#)

To make it easier for other application to use data from OpenAtlas directly we plan to implement an API (application programming interface).

There are already some use cases:

- THANADOS OpenAtlas presentation software, developed by Stefan Eichert
- ARCHE (for long time archiving), developed by ACDH-CH
- Pelagios

Purpose

Main purpose is the automatic data exchange between systems via URLs.

- All API request will include "api" after the domain URL e.g. **demo.openatlas.eu/api/0.1/1234** for one entity or **demo.openatlas.eu/api/0.1/actors** for a list of entities
- There will be multiple formats available (e.g. RDFS, JSON, XML)
- First step is to get basic information of an entity via including the entity id in the URL
- Next step will be to get (most) associated information e.g. information about super and sub entities, related entities like actors, events, ...
- Finally it should be possible to get all the information needed for e.g. OpenAtlas frontend presentation software, either especially developed or with a generic parameterized URL solution

Since the API should be very stable (additions are ok but no interface changes for possible other systems already using it) we will take some time to plan it in detail.

Resources

Great Resources on API development:

- [O'Reilly:RESTful Web APIs](#) - deposited with Alex
- [OpenAPI](#) - a machine-readable API documentation format/standard, with a large ecosystem of tools built around it
- [JSON:API](#) - a conceptual framework for API development and documentation
- [RF2616](#) - the HTTP specification, quite technical/theoretical, but good to be familiar with at least
- [FastAPI](#) - an interesting project to create a fast API
- [LinkedTraces](#) - Maybe for actors'?
- [JSON-LD-Playground](#) useful tool
- <http://www.geojsonlint.com/> -> Test if valid GeoJSON
- <https://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api>

Different Schema

Persons

- <https://json-ld.org/contexts/person.jsonld>

Ideas

Meeting Questions:

- The Linked Places field names (<https://github.com/LinkedPasts/linked-places#names-required>) requires at least one citation of the toponym name. Is it really required for our purpose? We can't provide a Gazetteer link or citation so it is not reasonable.
- Depictions Licenses, how to get the cc: namespace? We will make something like, if the string contains "cc" followed by "by" then add cc: in front of the name.

Todo:

Development:

- Research for automatic annotation for error handling and parameter
- Maybe re-implement everything with Flask-RESTful-API <https://flask-restful.readthedocs.io/en/latest/index.html>
 - Or search another Framework. We also need a Framework for rdfs