

### Discussions

- Output format: json as default, rdfls, ttl
  - strongly recommend using HTTP headers for content negotiation --> [RFC2616](#)

### Done

~~\* Do we allow the API to be accessed anonymously only after the data is already open or do we want a protected service. If yes, how can we secure it?~~

~~\* **there are multiple levels of access restriction, some on protocol level (such as CORS), some on content level (such as AUTH requirements)**~~

~~\* generally it should be considered if the API should provide CRUD functionality or be read only~~

~~\* What are the use cases? What queries do we want?~~

~~\* Password or token based authentication?~~

~~\* What shape type should we give an entity without real geometry? I would give it "Point" just out of simplicity~~

~~\* Is it really necessary for each project that we will give valid geojson? e.g. actor has geometry. Should we include more json schemas?~~

#### To discuss:

- How do we add CIDOC CRM related information and will this information be optional?
- Time format and especially for dates BC in GeoJSON T <https://github.com/kgeographer/geojson-t> and <https://www.loc.gov/standards/datetime/edtf.html>
- check names with Rainer Simon
- Should we add class to the json output? (e.g. E40, E18)

#### Discussion for 27.11

- For places we use the type "FeatureCollection", how should we name the other entities?
- Should we also use "feature": [] for other entities also? The thing is, that every information is in feature, but an actor doesn't have a "feature". Or do we say, it is ok "feature":[] includes any entity
- Should we differentiate between E18 and E53? Alex and Bernhard would say no
- Should we use comment at "when": {"timespans":[]}? Is it technically no problem and an additional information some could need
- We have to think about, how long we want to maintain old API versions and how we implement a version system
- What should happen, if a Type/Node is requested from the API? Should we show the parent and all subtypes? Or only show parent?
- What should happen with empty tags?
  - e.g. an entity doesn't have dates, is the "when":{} tag empty or not present?
  - e.g. an entity has no types... etc
- Should @context reference to the api Uri? The linked places documentation states: "In JSON-LD, labels for object elements are aliases for terms formally defined in linked ontologies. For Linked Places, those mappings are listed in this context document. (<http://linkedplaces.org/assets/linkedplaces-context.jsonld>)"

#### Discussion outcome:

- URL schema: /api/v2/entity/1234 or /api/v2/entity/1234.json for specific format
- Linked Places Format (LPF) provides a quite solid format and will be used
- The database structure is ideal for creating unique IDs. Rainer and Christoph approved of the doability, that every entity will be available with the URL shema /api/v1/1234
- The CIDOC shema could be included with the tag "relations", but further research is required.
  - Some ideas: [CIDOC JSON-LD Serialization](#)—Serialization attempt of a picture
- Content: What to show in output—linked places, linked traces, cidoc mapping included, various options

#### 21.02.

- every entry has to be type: "Feature"
- geometry Point, Polygon etc has to be case sensitiv e.g. Point not point
- geometry has to be in the json!!
- validate through <http://www.geojsonlint.com/> but linked places are our preferences
- Everything is a FeatureCollection !!