Common WP3 POI API and Format

Please feel free to comment!

REST API - General matters

- Overall two types of resources
  - POIs and their data components
  - Query result sets
- TODO: agree on singular or plural nouns - /api/poi vs /api/pois (preferred)
- TODO: handle data components as related resources?

REST Resources

- two types of resources: POI entities and Query results

POI Resource

```
{
        "fw_core": uri / object,
        "fw_media": uri / object,
        [...]
}
```

Query Result Resource

```
{
        "pois": {
                "30ddf703-59f5-4448-8918-0f625a7e1122": {
                        "fw_core": uri / object,
                        "fw_media": uri / object
                },
                "4df3efde-8d5c-42bd-95ae-7b0991899968": {
                        [...]
                },
                [...]
        },

        "query": {
        }
}
```

Accessing POI data directly

GET /api/pois/{{poi-uuid}}

- accessing a single POI by uuid
- examples:
  - /api/pois/30ddf703-59f5-4448-8918-0f625a7e1122
- TODO: which components to return by default? expanded or collapsed?
- TODO: discuss error handling
  - invalid uuid format

    - unknown uuid
    - known uuid, but no stored components at all
- TODO: specify return structure

GET /api/pois/{{poi-uuid}}/{{component-name}}

GET /api/pois/{{poi-uuid}}/components/{{component-name}}

- accessing a single expanded component of a POI
- examples:
    - /api/pois/30ddf703-59f5-4448-8918-0f625a7e1122/fw_media
- TODO: use components resource keyword?
- TODO: discuss error handling
    - invalid uuid format (see above)
    - unknown uuid (see above)
    - known uuid, but request component name is unknown
    - known uuid, known component name, but no data available for this component
- TODO: specify return structure

Querying POI data (filter by UUID)

GET /api/pois?ids={{csv(poi-uuid)}}

GET /api/queries?ids={{csv(poi-uuid)}}

GET /api/queries/results?ids={{csv(poi-uuid)}}

- retrieve POIs with regard to their uuids
- comma separated list (csv) of uuids
- list with one uuid only correspond to the direct access of a POI
- examples:
    - /api/queries/results/?ids=30ddf703-59f5-4448-8918-0f625a7e1122,5af4f923-9a5a-4d6e-af9a-3a53492228df
- TODO: discuss, whether to use “pois” or “queries” or “queries/results” (preferred)
- TODO: discuss error handling
    - invalid csv list format
    - others (see /api/pois/{{poi-uuid}}
- TODO: specify return structure

Querying POI data (filter by spatial data)

GET /api/queries/results?spatial={{spatial-operation}}

- unification of spatial query API
- merge the different spatial queries using a single url parameter
- keyword for the actual spatial operation (bbox, radial, nearest, polygon)
- parameters for spatial operation encoded in brackets, e.g. <params> or (params)
- examples:
    - /api/queries/?spatial=bbox<south,west,north,east>
      /api/queries/?spatial=bbox(south,west,north,east)
    - /api/queries/?spatial=radial<lat,lon,radius>
    - /api/queries/?spatial=nearest<lat,lon[,radius]>
    - /api/queries/?spatial=poly<lat,lon,lat,lon,lat,lon[,lat,lon,...]> (optional)
- TODO: discuss proposal and declare required and optional parameters
- TODO: specify return structure

GET /api/queries/?spatial=bbox&bbox=lat,lon,lat,lon

- bounding box
- *south,west,north,east*
- bbox=<lower left and upper right>
- TODO: double check with practice of established frameworks (e.g. OpenLayers)

GET /api/queries/?spatial=polygon&polygon= (optional)

- ODS supports polygons
- polygon=<list of lat/lon pairs>
- TODO: double check whether lat/lon or lon/lat is desired

GET /api/queries/?spatial=radial&lon=&lat=&radius=

- lon=<WGS84 longitude>
- lat=<WGS84 latitude>
- radius=<meters>

GET /api/queries/?spatial=nearest&lon=&lat=&radius=

- nearest - sorting by distance (simply special version of radial search)
- lon=<WGS84 longitude>
- lat=<WGS84 latitude>
- radius=<meters>

Querying POI data (filter by property values)

- generic and flexible approach to query for POI data with arbitrary values
- independent of actual POI structure and invariant to custom data components
- single-use queries with property paths encoded in request url
- query template and instancing mechanism for frequently executed queries

GET /api/queries/results?{{property-path}}={{value}}

- {{property-path}} is dot (".") separated property names starting with data component name
- {{value}} is any string value that is dynamically converted (e.g. to match an integer)
- examples:
    - /api/queries/results?fw_core.name=Hotel%20Paris
    - /api/queries/results?fw_marker.alvar_5x5.code=82
    - /api/queries/results?fic2_blebeacon.uuid=d58af054-a4bd-4105-a188-b21e34569677
    - /api/queries/results?fw_core.name=matches([hH]otel)
- TODO: specify special functions, i.e.
    - not(value) / !value
    - in(value1,value2,value3,value4)
    - contains(value) / matches(regex)
    - range(min, max)

POST /api/queries

- create a new query template based on jsonschema and return the uuid
- post body contains the jsonschema
- result set of this query will contain only POIs that validate against the schema
- POI webservice might
    - transform the jsonschema into a respective db query

    - do some kind of caching and return uuid of existing query if applicable

GET /api/queries/{{query-uuid}}/results

- instantiate the query and performs the actual request
- returns the result set, which contains only POIs that validate against the jsonschema of this query
- TODO: decide whether or not we desire similar special functions as for single-use queries?
- TODO: discuss lifetime of result set to enable pagination

GET /api/queries/{{query-uuid}}/results?{{param-name}}={{value}}

- query templates support parameter binding
- query's jsonschema might contain placeholders (e.g. {{param-name}}), which will be replaced during instantiation of the query with {{value}}
- thus, we can create complex parameterized queries
- approach enables offline query optimization

Customize retrieval of POI data

- union set of get_components and expand_components is actually delivered as response data per POI
- applicable to single POIs as well as query result sets

/api/pois/{{poi-uuid}}?get_components={{csv(component-name)}}

- retrieved POI data contains listed components as URIs (e.g. in the form of /api/pois/{{poi-uuid}}/components/{{component-name}})
- TODO: what about unavailable components
    - property present with null
    - property omitted
- examples:
    - /api/pois/{{poi-uuid}}?get_components=fw_core,fw_media,fw_xml

/api/pois/{{poi-uuid}}?expand_components={{csv(component-name)}}

- retrieved POI data contains listed components expanded with actual data
- TODO: what about unavailable components
    - property present with null (see above)
    - property omitted (see above)
- examples:
    - /api/pois/{{poi-uuid}}?expand_components=fw_core,fw_media

/api/poi/{{poi-uuid}}?language={{csv(iso-lang-code)}}

- reduces properties with multilingual content to specified languages
- either handled by Accept header or through this url parameter
- TODO: decide priority of header and url parameter
- examples:
    - /api/poi/{{poi-uuid}}?languages=de,en,fr

Service (meta) information

GET /api/meta/category

- retrieve (available) categories
- The response could be something similar like

```
"category": {
     "description": "List of categories the POI is connected to.",
     "type": "array",
     "items": {

          "name": {
          "description": "specifies the name of the category",
          "type": "string"
          },
          "description": {
          "description": "describes the category",
          "type": "string"
          },
     "level": {
          "description": "describes the level of the category",
          "type": "integer"
          },

     },

     "required": [
           "name",
     ],
     "additionalProperties": false
}
```

common parameters

- **limit /** max_results (default = 31)
- offset - for optional paging
- category
- title
- fulltextsearch
- has_component=<cs list of components>
  - e.g. fw_xml3d, fw_marker
  - same names as data components
- apikey

Open issues

- order of results The current data structure (hashmap) does not allow any ordering, but an easy access to an POI. Proposal:
  ```
  [
          { “id”: “30ddf703-59f5-4448-8918-0f625a7e1122”,
           “fw_core”: uri / object,
           “fw_media”: uri / object
          }, ...
  ]
  ```
  - relevance (reviews, matching conditions, …)
- how to retrieve supported categories and query appropriately
- how to retrieve supported components ?

Components

The following components are supported:

- fw_core
- fw_media
- fw_xml3d
- fw_relationships
- fw_marker
- fw_time
- fw_contact
- fic2_fusion_tracking
- fic2_dynamic_distance
- ...

fw_core

```
{
      "title": "Core information",
      "description": "For spatial search and finding that interesting one",
      "type": "object",
      "properties": {
        "category": {
          "title": "Category",
          "description": "A descriptive tag for narrowing the search: cafe, museum, 
etc.",
          "type": "string"
        },
        "location": {
          "title": "Location",
          "description": "Location of the POI",
          "$ref": "#/definitions/location"
        },
        "geometry": {
          "title": "Geometrical form of the POI",
          "description": "Format: Open Geospatial Consortium's 'Well-known text' 
ISO/IEC 13249-3:2011",
          "type": "string"
        },
        "short_name": {
          "title": "Short name",
          "description": "Short name (max. 31 chars) to be shown on the map or in a 
narrow list",
```

```
          "$ref": "#/definitions/intl_string_31"
        },
        "name": {
          "title": "Name",
          "description": "Descriptive name",
          "$ref": "#/definitions/intl_string"
        },
        "label": {
          "title": "Label",
          "description": "More info to complement the name, if enough space",
          "$ref": "#/definitions/intl_string_127"
        },
        "description": {
          "title": "Description",
          "description": "Text to facilitate decision to be interested or not",
          "$ref": "#/definitions/intl_string"
        },
        "thumbnail": {
          "title": "Thumbnail",
          "description": "Link to a small picture to be shown on a list, scene or
map. Preferably max. 256x256 pixels, e.g. 120x160.",
          "type": "string",
          "format": "uri"
        },
        "url": {
          "title": "Web address",
          "description": "URL to get more info, preferably official website of the
POI",
          "$ref": "#/definitions/intl_uri"
        },
        "source": {
          "title": "Source of information",
          "$ref": "#/definitions/source"
        },
        "last_update": {
          "title": "Last update",
          "description": "DO NOT EDIT! Information to identify the version of the
data component.",
          "$ref": "#/definitions/update_stamp"
        }
      },
      "required": [
        "category",
        "location"
      ],
      "additionalProperties": false
    }
```

fw_contact

```
{
      "properties": {
        "visit": {
          "description": "Visiting address good for a taxi driver or Google Maps",
          "type": "string"
        },
        "postal": {
          "description": "Postal address. One string per line",
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "mailto": {
          "description": "Email address",
          "type": "string"
        },
        "phone": {
          "description": "Phone number",
          "type": "string"
        },
        "source": {
          "title": "Source of information",
          "$ref": "#/definitions/source"
        },
        "last_update": {
          "$ref": "#/definitions/update_stamp"
        }
      },
      "additionalProperties": false
    }
```

fw_media

```
{
      "description": "Media items related to this POI item",
      "type": "object",
      "properties": {
        "entities": {
          "description": "",
          "type": "array",
          "items": {
            "description": "",
            "type": "object",
            "properties": {
              "type": {
                "description": "what kind of media item this is",
                "enum": [
                  "folder",
                  "photo",
                  "video",
                  "audio"
                ]
              },
              "short_label": {
                "description": "To be shown along the item",
                "$ref": "#/definitions/intl_string_31"
              },
              "caption": {
                "description": "To be shown along the item",
                "$ref": "#/definitions/intl_string_127"
              },
              "description": {
                "description": "More info about the item",
                "$ref": "#/definitions/intl_string"
              },
              "thumbnail": {
                "description": "A small picture to be shown in the list",
                "type": "string",
                "format": "uri"
              },
              "url": {
                "description": "URL of the actual media item",
                "type": "string",
                "format": "uri"
              },
              "copyright": {
                "description": "Copyright clause and/or attribution of the item",
                "type": "string"
              }
            },
            "required": [
              "type",
              "url"
            ],
            "additionalProperties": false
          }
        },
        "last_update": {
          "title": "Last update",
          "description": "DO NOT EDIT! Information to identify the version of the
data component.",
          "$ref": "#/definitions/update_stamp"
        }
      },
      "additionalProperties": false
    }
```

fw_xml3d

```
{
      "description": "3D description",
      "type": "object",
      "properties": {
        "model_id": {
          "description": "ID for XML3D engine",
          "type": "string"
        },
        "model": {
          "description": "Model for XML3D engine",
          "type": "string"
        },
        "source": {
          "title": "Source of information",
          "$ref": "#/definitions/source"
        },
        "last_update": {
          "title": "Last update",
          "description": "DO NOT EDIT! Information to identify the version of the
data component.",
          "$ref": "#/definitions/update_stamp"
        }
      },
      "additionalProperties": false
    }
```

fw_time

```
{
      "title": "fw_time",
      "description": "Temporal availability of the place or the associated
service",
      "type": "object",
      "properties": {
        "type": {
          "title": "",
          "description": "Open - available thru open time, show_times - available
at beginnings of shows",
          "enum": [
            "open",
            "show_times"
          ]
        },
        "time_zone": {
          "title": "Time zone, - under development -",
          "description": "TBD. Local time is assumed. Standardized notation
including daylight savings time reference is needed."
        },
        "schedule": {
          "title": "Schedule",
          "description": "Definition of times of availability",
          "$ref": "#/definitions/schedule"
        },
        "source": {
          "title": "Source of information",
          "$ref": "#/definitions/source"
        },
        "last_update": {
          "title": "Last update",
          "description": "DO NOT EDIT! Information to identify the version of the
data component.",
          "$ref": "#/definitions/update_stamp"
        }
      },
      "required": [
        "type",
        "schedule"
      ],
      "additionalProperties": false
    }
```

fw_relationships

```
{
      "description": "List of relationships the POI is connected to.",
      "type": "array",
      "items": {
        "description": "specifies ONE to MANY relation between POIs or other
entities",
        "type": "object",
        "properties": {
          "subject": {
            "description": "UUID of the ONE in the relation",
            "type": "string"
          },
          "predicate": {
            "description": "type of the relation",
            "type": "object",
            "additionalProperties": {
              "description": "defines the type of the relation within ontology
defined by the key",
              "type": "string"
            }
          },
          "objects": {
            "type": "array",
            "items": {
              "description": "UUIDs of the MANY in the relation",
              "type": "string"
            }
          },
          "last_update": {
            "$ref": "#/definitions/update_stamp"
          }
        },
        "additionalProperties": false
      }
    }
```

fw_marker

```
{
      "title": "fw_marker",
      "description": "Marker choices for different tracking techniques",
      "type": "object",
      "properties": {
        "alvar_3x3": {
          "title": "Alvar 3x3",
          "description": "3x3 marker used in Alvar (VTT Oulu) marker tracking
system is used as an example here.",
          "type": "object",
          "properties": {
            "code": {
              "description": "code embedded to marker as defined by Alvar",
              "type": "integer"
            },
            "image_ref": {
              "description": "image of the marker e.g. for printing",
              "type": "string",
              "format": "uri"
            }
          },
          "additionalProperties": false
        },
        "alvar_5x5": {
          "title": "Alvar 5x5",
          "description": "5x5 marker used in Alvar (VTT Oulu) marker tracking
system is used as an example here.",
          "type": "object",
          "properties": {
            "code": {
              "description": "code embedded to marker as defined by Alvar",
              "type": "integer"
            },
            "image_ref": {
              "description": "image of the marker e.g. for printing",
              "type": "string",
              "format": "uri"
            }
          },
          "additionalProperties": false
        },
        "image": {
          "title": "Image",
          "description": "Image-only marker for generic use",
          "type": "object",
          "properties": {
            "image_ref": {
              "description": "image of the marker e.g. for printing",
              "type": "string",
              "format": "uri"
            }
          },
          "additionalProperties": false
        },
        "source": {
          "title": "Source of information",
          "$ref": "#/definitions/source"
        },
        "last_update": {
          "title": "Last update",
          "description": "DO NOT EDIT! Information to identify the version of the
data component.",
          "$ref": "#/definitions/update_stamp"
        }
      },
      "additionalProperties": false
    }
  }
```

fic2_dynamic_relative_location_whatever (tbd)

- distance to own position (in meters)
- bearing - direction (in degrees)