Sociocortex Tutorial

Daniel Elsner – daniel.elsner@tum.de

Table of Contents

A hands-on tutorial for the sociocortex (https://sociocortex.com) REST API.

▶ Run in Postman

1. Introduction

1.1. Motivation

What is sociocortex and explanation how the REST API is used to communicate with the platform in client applications.

1.2. Concepts

- Architecture (Layers)
- Class Hierarchy (Entities, Workspaces, Tasks, etc.)
- Basic types in sociocortex (Numbers, String, Enum, Date, etc.)
- Usage (server.sociocortex (CM?), REST API, sc-angular, modeler, visualizer) (?)

2. Configuration

- Links to Bitbucket/Github repo
- Offline version: Sociocortex setup
- Online version: Authentication
  - POSTMAN test-setup (create?!)

3. Getting started with the REST API

3.1. Introduction to demo scenario

The demo scenario that will serve as a real world example to demonstrate the usage of the *sociocortex REST API* is a simple publication management system.

Before we can start with the actual scenario, we check that the authentication is working for your credentials and shortly outline the differences of jwt and basic auth.

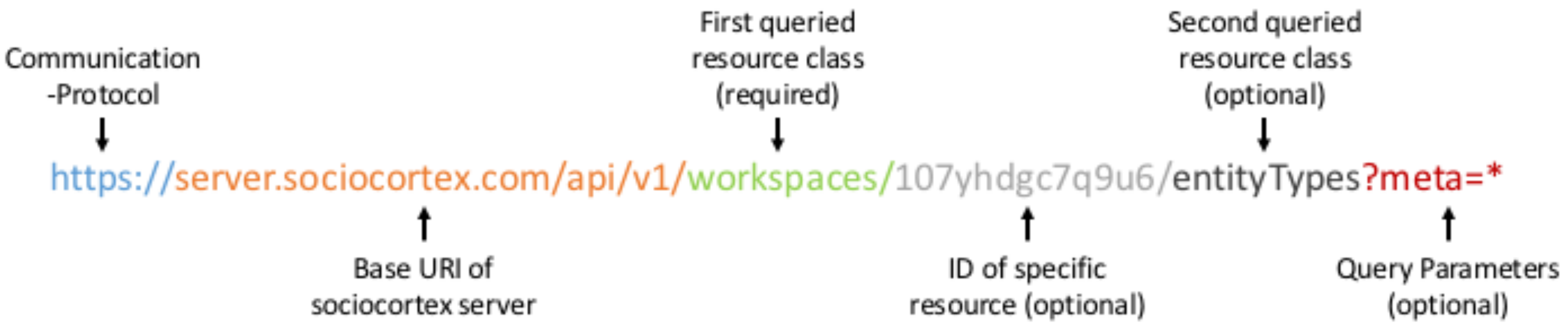
Once the authentication works, we create a workspace which we then use to store the required entity types in. We identify the required entity types as *publication* and *conference*. The latter we use to add entities, i.e. publications or conferences, to the workspace with respective entity types.

Furthermore, tasks and their definitions are introduced and combined with the attribute definitions of entity types.

While going through the demo scenario all HTTP verbs will be taken into account in order to make you feel comfortable with all kinds of actions necessary to interact with the *sociocortex REST API*.

3.1.1. Structure of resource URIs

Example URI for accessing all resources of type entityType within a workspace with a specific ID:



Communication -Protocol
First queried resource class (required)
Second queried resource class (optional)
https://server.sociocortex.com/api/v1/workspaces/107yhdgc7q9u6/entityTypes?meta=*
Base URI of sociocortex server
ID of specific resource (optional)
Query Parameters (optional)

Additionally, one can provide query parameters to further refine for instance the requested meta attributes. More on query parameters can be found in Query Parameters.

3.2. Authentication

In order to get started with *sociocortex*, we first need to create an account to be able to create an own workspace as well as other entities.

3.2.1. Registering a new user (POST)

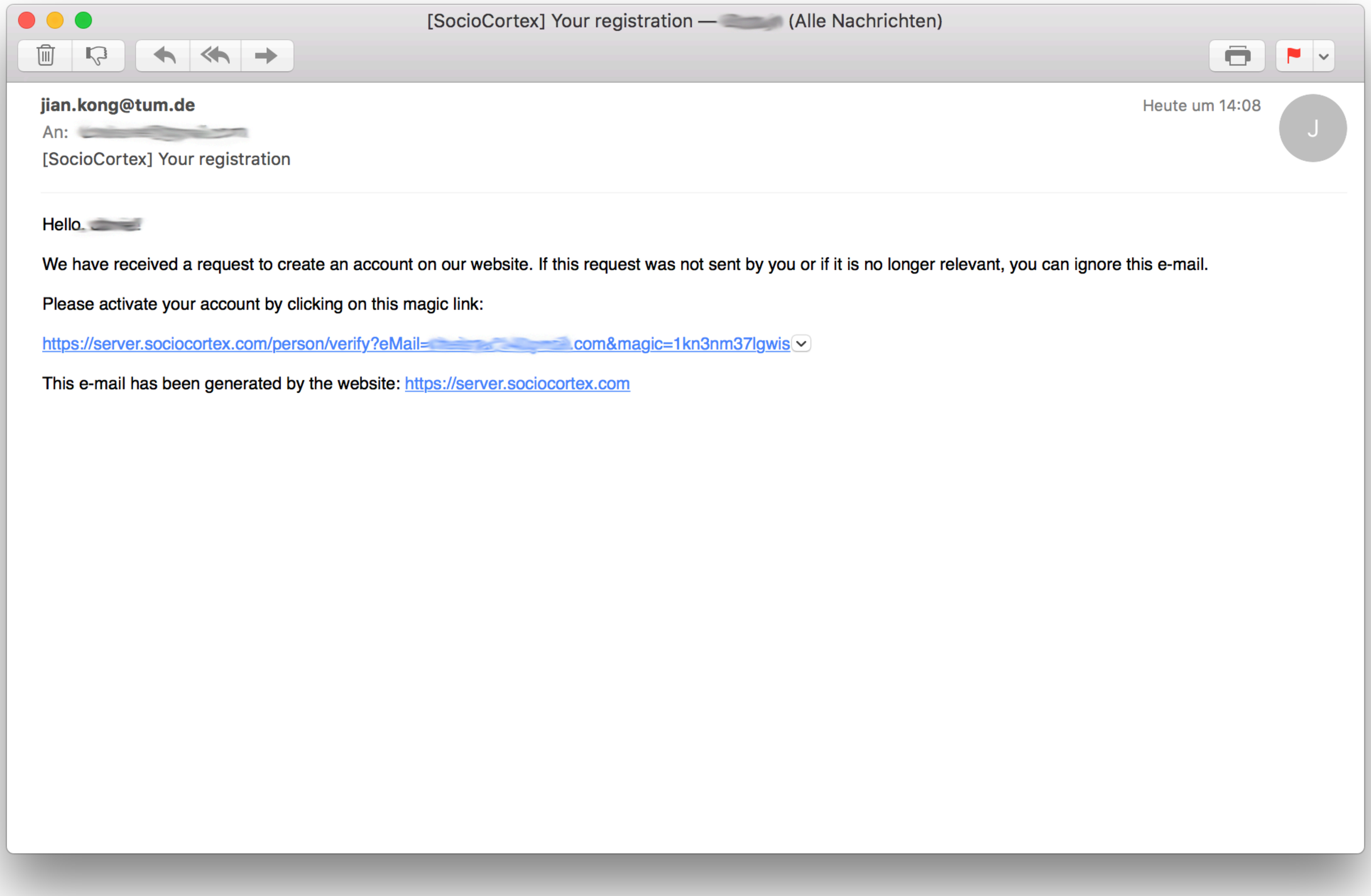
Request:

```bash
POST https://server.sociocortex.com/api/v1/users
BODY
{
    "name":"$NAME",
    "email":"$EMAIL@ADDRESS.com",
    "password":"$PASSWORD"
}
```

Response:

```json
{
  "isVerified": false,
  "locale": "de-DE",
  "mayCreateWorkspace": false,
  "mayCreateGroup": false,
  "id": "$ID",
  "versions": [],
  "email": "$EMAIL@ADDRESS.com",
  "name": "$NAME",
  "href": "https://server.sociocortex.com/api/v1/users/$ID",
  "mayEdit": true,
  "isAdministrator": false,
  "mayCreateUser": false,
  "groups": []
}
```

After having successfully send this information to the server, we check the respective provided email account for an email containing an activation link for *sociocortex*.



[SocioCortex] Your registration — [redacted] (Alle Nachrichten)

jian.kong@tum.de

An: [redacted]

[SocioCortex] Your registration

Heute um 14:08

Hello [redacted]

We have received a request to create an account on our website. If this request was not sent by you or if it is no longer relevant, you can ignore this e-mail.

Please activate your account by clicking on this magic link:

https://server.sociocortex.com/person/verify?eMail=[redacted].com&magic=1kn3nm37lgwis

This e-mail has been generated by the website: https://server.sociocortex.com

Once we click the link in the email, the account is activated and we are ready to go.

3.2.2. Log in to *sociocortex*

There are two ways to log in to *sociocortex*, either by HTTP Basic authentication or by using JSON Web Tokens (JWT, https://jwt.io/). Throughout this tutorial we will be using the predefined HTTP requests from the Postman-Collection. Therefore, we encourage you to simply replace the placeholder "$TOKEN" in all the Postman requests with your own JWT (see JWT).

Basic Auth

Basic Auth requires to send a respective HTTP Authorization Header in every request to the server. This Authorization Header contains a username and password combined with a single colon and base64-encoded with a "Basic" prefix, denoting the authorization method. Thus, it does not require any cookies, session identifiers or tokens provided and signed by the server.

An example request to get a users details could be the following:

Request:

```BASH
GET https://server.sociocortex.com/api/v1/users/me
Authorization: Basic $BASE64_ENCODED_CREDENTIALS
```

JWT

A JWT can be received by providing the credentials to the server in the payload of a HTTP POST request. The token has an expiry date in its payload as well as user information.

Request:

```BASH
POST https://server.sociocortex.com/api/v1/jwt
BODY
{
    "username":"$EMAIL@ADDRESS.com",
    "password":"$PASSWORD"
}
```

Response:

```json
{
  "token": "$TOKEN",
  "payload": {
    "exp": "$EXPIRY_DATE",
    "sub": "$USER_ID",
    "name": "$NAME"
  }
}
```

After having received the token, the token is sent with every request on access-restricted resources in a respective HTTP Authorization Header with the prefix "Bearer".

An example request to get a users details could be the following:

Request:

```bash
GET https://server.sociocortex.com/api/v1/users/me
Authorization: Bearer $TOKEN
```

3.3. Creating a workspace

Before creating our first workspace we check for the currently available workspaces because who knows, maybe someone else did the job for us already.

We do this with a simple HTTP GET request on the workspaces resource. The server will respond with a list of workspaces stored.

3.3.1. Looking for existing workspaces (GET)

Request:

```bash
GET https://server.sociocortex.com/api/v1/workspaces
```

Response:

```JSON
[
  {
    "id": "944wcmiregxu",
    "name": "sc-visualizer",
    "href": "https://server.sociocortex.com/api/v1/workspaces/944wcmiregxu"
  },
  {
    "id": "107yhdgc7q9u6",
    "name": "Northwind",
    "href": "https://server.sociocortex.com/api/v1/workspaces/107yhdgc7q9u6"
  }
]
```

We can tell that no one has yet created a publication management workspace so we continue by creating one with a HTTP POST request.

3.3.2. Creating workspace *publication management* (POST)

Request:

```JSON
POST https://server.sociocortex.com/api/v1/workspaces
BODY
{
    "name": "Publication Management"
}
```

Response:

```json
{
  "id": "10sasph8dvxhg",
  "versions": ["..."],
  "rootEntity": {
    "id": "bqr7jmg37vqt",
    "name": "Publication Management - Home",
    "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt"
  },
  "mayCreateEntities": true,
  "isFavorite": false,
  "entityTree": {
    "id": "bqr7jmg37vqt",
    "name": "Publication Management - Home",
    "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt"
  },
  "name": "Publication Management",
  "permissions": {
    "contributors": [],
    "administrators": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "writers": [],
    "readers": []
  },
  "customFunctions": [],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
}
```

3.3.3. Looking at a single workspace (GET)

As we have seen in the response of the server on our POST request the workspace has been created with an id (in my case the id *10sasph8dvxhg*). We can easily use this id now to get this specific workspace. In the following requests replace the id *10sasph8dvxhg* with your own "$WORKSPACE_ID".

Request:

```bash
GET https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg
```

Response:

```json
{
  "id": "10sasph8dvxhg",
  "versions": [
    {
      "action": "Edited",
      "date": "2016-07-15T17:06:42.012Z",
      "type": "_edit",
      "user": {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    },
    {
      "action": "Added",
      "date": "2016-07-15T17:06:41.371Z",
      "type": "_new",
      "user": {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    }
  ],
  "rootEntity": {
    "id": "bqr7jmg37vqt",
    "name": "Publication Management - Home",
    "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt"
  },
  "mayCreateEntities": true,
  "isFavorite": false,
  "entityTree": {
    "id": "bqr7jmg37vqt",
    "name": "Publication Management - Home",
    "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt"
  },
  "name": "Publication Management",
  "permissions": {
    "contributors": [],
    "administrators": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "writers": [],
    "readers": []
  },
  "customFunctions": [],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
}
```

3.4. Creating entity types

Now that we have created the workspace where the publication management system will store its contents, we can continue by creating two entity types required to store *publications* prepared for and presented on multiple *conferences*.

3.4.1. Creating *publication* entity type (POST)

To create an entity type in a workspace we simply concat the URI of the workspace by */entityTypes* and use the HTTP POST method to create a new entity type *publication*.

Request:

```bash
POST https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg/entityTypes
BODY
{
    "name": "Publication"
}
```

Response:

```json
{
  "processes": [],
  "attributeDefinitions": [],
  "taskDefinitions": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "entitiesCount": 0,
  "structurednessMetric": 1,
  "id": "dt56spnul01p",
  "versions": [
    {
      "action": "Added",
      "date": "2016-07-15T17:26:37.202Z",
      "type": "_new",
      "user": {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    }
  ],
  "consistencyMetric": 1,
  "derivedAttributeDefinitions": [],
  "name": "Publication",
  "allowFreeAttributes": false,
  "associations": [],
  "incomingAssociations": [],
  "customFunctions": [],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
}
```

3.4.2. Creating *conference* entity type (POST)

Request:

```bash
POST https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg/entityTypes
BODY
{
    "name": "Conference"
}
```

Response:

```json
{
  "processes": [],
  "attributeDefinitions": [],
  "taskDefinitions": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "entitiesCount": 0,
  "structurednessMetric": 1,
  "id": "19c48jgd4b8m2",
  "versions": [
    {
      "action": "Added",
      "date": "2016-07-15T17:29:20.655Z",
      "type": "_new",
      "user": {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    }
  ],
  "consistencyMetric": 1,
  "derivedAttributeDefinitions": [],
  "name": "Conference",
  "allowFreeAttributes": false,
  "associations": [],
  "incomingAssociations": [],
  "customFunctions": [],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
}
```

Finally, we have our workspace and the two required entity types set up. In the next steps we refine the attributes of the entity types before we start creating entities in the step after.

3.5. Creating attribute definitions

Sociocortex offers the possibility to create custom attribute definitions for attributes of entity types. This is a common task and you can make use of the attribute types introduced in Concepts or define relations between entity types.

3.5.1. Creating text attributes (POST)

The simplest attribute defintion type is surely a text attribute. We create a text attribute *Abstract* which should store the abstract of a *publication*. Therefore we send a HTTP POST request on the attribute definitions of the respective entity type - in this case the *publication*.

Request:

```BASH
POST
https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/attributeDefinitions
BODY
{
    "name": "Abstract",
    "multiplicity": "maximalOne",
    "attributeType": "Text"
}
```

Response:

```JSON
{
  "id": "2gegqgdueyqd",
  "versions": [],
  "readOnly": false,
  "name": "Abstract",
  "multiplicity": "maximalOne",
  "attributeType": "Text",
  "entityType": {
    "id": "dt56spnul01p",
    "name": "Publication",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/2gegqgdueyqd"
}
```

More attribute definitions of the various types are created in the next sections. For the matter of simplicity, we leave out the responses for the following requests.

3.5.2. Creating numeric attributes (POST)

Every conference needs a year in which it is taking place. We use a numeric attribute respectively:

Request:

```bash
POST
https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2/attributeDefinitions
BODY
{
    "name": "Year",
    "multiplicity": "maximalOne",
    "attributeType": "Number"
}
```

3.5.3. Creating enumeration attributes (POST)

Every publication needs one of the statuses: Planned, Submitted, Accepted. We will create an attribite defintion of type enum to realize this.

Request:

```bash
POST
https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/attributeDefinitions
BODY
{
    "name": "Status",
    "multiplicity": "maximalOne",
    "attributeType": "Enumeration",
    "options": {
        "enumerationValues": [
          "Planned",
          "Submitted",
          "Accepted"
        ]
    }
}
```

Conferences also require an enum attribute - the rank of the conference:

Request:

```bash
POST
https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2/attributeDefinitions
BODY
{
    "name": "Rank",
    "multiplicity": "maximalOne",
    "attributeType": "Enumeration",
    "options": {
        "enumerationValues": [
          "A",
          "B",
          "C"
        ]
    }
}
```

3.5.4. Creating relation attributes (POST)

A simple example for a relation among two entity types is an *acceptedIn* relation for publications. Once they are accepted by a conference the conferences are stored in this attribute.

Request:

```bash
POST
https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/attributeDefinitions
BODY
{
    "name": "acceptedIn",
    "multiplicity": "any",
    "attributeType": "Link",
    "options": {
        "entityType": {
          "id": "19c48jgd4b8m2"
        },
        "resourceType": "entities"
    }
}
```

Another straightforward relation is the mapping of authors to publications:

Request:

BASH

```
POST
https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/attributeDefinitions
BODY
{
    "name": "authors",
    "multiplicity": "any",
    "attributeType": "Link",
    "options": {
        "resourceType": "users"
    }
}
```

3.6. Creating entities

To create entities of the two entity types send HTTP POST requests to the types' endpoints, defined by their respective id.

3.6.1. Creating a *conference* entity (POST)

Request:

BASH

```
POST https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2/entities
BODY
{
    "name": "International Conference on Software Engineering (ICSE)",
    "attributes":[
        {
            "name":"Rank",
            "values":["A"]
        },
        {
            "name":"Year",
            "values":[2016]
        }
    ]
}
```

Response:

JSON

```
{
  "files": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "children": [],
  "tasks": [],
  "entityType": {
    "id": "19c48jgd4b8m2",
    "name": "Conference",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
  },
  "incomingReferences": {},
  "creator": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "id": "n6eidlmr0by7",
  "versions": ["..."],
  "createdAt": "2016-07-24T11:12:00.000Z",
  "name": "International Conference on Software Engineering (ICSE)",
  "lastModifier": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "permissions": {
    "writers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "readers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ]
  },
  "attributes": [
    {
      "id": "pme5rnjsvx5z",
      "values": [
        "A"
      ],
      "name": "Rank",
      "href": "https://server.sociocortex.com/api/v1/attributes/pme5rnjsvx5z"
    },
    {
      "id": "voa7iqjet7pf",
      "values": [
        2016
      ],
      "name": "Year",
      "href": "https://server.sociocortex.com/api/v1/attributes/voa7iqjet7pf"
    }
  ],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entities/n6eidlmr0by7",
  "lastModifiedAt": "2016-07-24T11:12:00.000Z"
}
```

3.6.2. Creating a *publication* entity (POST)

Request:

```BASH
POST https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/entities
BODY
{
    "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in
Collaborative Information Systems",
    "attributes":[
        {
            "name":"acceptedIn",
            "values":[{"id":"n6eidlmr0by7"}]
        },
        {
            "name":"Abstract",
            "values": ["Today's enterprises have to align their information
systems continuously with their dynamic business and IT environment. ..."]
        },
        {
            "name":"Status",
            "values":["Accepted"]
        },
        {
            "name":"authors",
            "values":[{
                "id":"1hda5st2csfom"
            }]
        }
    ]
}
```

Response:

```JSON
{
  "files": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "children": [],
  "tasks": [],
  "entityType": {
    "id": "dt56spnul01p",
    "name": "Publication",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  "incomingReferences": {},
  "creator": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "id": "127trtufzdgcw",
  "versions": ["..."],
  "createdAt": "2016-07-24T11:27:26.000Z",
  "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in
Collaborative Information Systems",
  "lastModifier": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "permissions": {
    "writers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "readers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ]
  },
  "attributes": [
    {
      "id": "7mgjht1rkna8",
      "values": [
        {
          "id": "1hda5st2csfom",
          "name": "Thomas Reschenhofer",
          "href": "https://server.sociocortex.com/api/v1/users/1hda5st2csfom"
        }
      ],
      "name": "authors",
      "href": "https://server.sociocortex.com/api/v1/attributes/7mgjht1rkna8"
    },
    {
      "id": "iqqlpxfbnr8f",
      "values": [
        {
          "id": "n6eidlmr0by7",
          "name": "International Conference on Software Engineering (ICSE)",
          "href": "https://server.sociocortex.com/api/v1/entities/n6eidlmr0by7"
        }
      ],
      "name": "acceptedIn",
      "href": "https://server.sociocortex.com/api/v1/attributes/iqqlpxfbnr8f"
    },
    {
      "id": "rvzij9nn9of4",
      "values": [
        "Accepted"
      ],
      "name": "Status",
      "href": "https://server.sociocortex.com/api/v1/attributes/rvzij9nn9of4"
    },
    {
      "id": "ys69v6zjfz54",
      "values": [
        "Today's enterprises have to align their information systems continuously with their dynamic business and IT environment. ..."
      ],
      "name": "Abstract",
      "href": "https://server.sociocortex.com/api/v1/attributes/ys69v6zjfz54"
    }
  ],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entities/127trtufzdgcw",
  "lastModifiedAt": "2016-07-24T11:27:26.000Z"
}
```

3.7. Getting entity types

As we have created two entity types with respective instances, it might be interesting to query these types and their attribute definitions. The *sociocortex REST API* offers the possibility to further refine queries by query parameters, e.g. by specifying desired attributes or meta attributes. A more detailed description can be found in Query Parameters.

3.7.1. Looking at all created entity types (GET)

Request:

BASH

```
GET https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg/entityTypes
```

Response:

```json
[
  {
    "id": "dt56spnul01p",
    "name": "Publication",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  {
    "id": "19c48jgd4b8m2",
    "name": "Conference",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
  }
]
```

If more specific information about these entity types is required, we simply add the query parameter *meta* with value * :

Request:

```bash
GET https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg/entityTypes?
meta=*
```

Response:

```json
[
  {
    "processes": [],
    "attributeDefinitions": [
      {
        "id": "2gegqgdueyqd",
        "versions": [],
        "readOnly": false,
        "name": "Abstract",
        "multiplicity": "maximalOne",
        "attributeType": "Text",
        "entityType": {
          "id": "dt56spnul01p",
          "name": "Publication",
          "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
        },
        "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/2gegqgdueyqd"
      },
      {
        "id": "1knrlnoj7kjer",
        "versions": [],
        "readOnly": false,
        "name": "acceptedIn",
        "multiplicity": "any",
    "attributeType": "Link",
    "entityType": {
     "id": "dt56spnul01p",
     "name": "Publication",
     "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
    },
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer",
    "options": {
     "entityType": {
      "id": "19c48jgd4b8m2",
      "name": "Conference",
      "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
     },
     "resourceType": "entities"
    }
   },
   {
    "id": "18b1hu00gnxdi",
    "versions": [],
    "readOnly": false,
    "name": "authors",
    "multiplicity": "any",
    "attributeType": "Link",
    "entityType": {
     "id": "dt56spnul01p",
     "name": "Publication",
     "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
    },
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/18b1hu00gnxdi",
    "options": {
     "resourceType": "users"
    }
   },
   {
    "id": "mfpco2wx3zqp",
    "versions": [],
    "readOnly": false,
    "name": "Status",
    "multiplicity": "maximalOne",
    "attributeType": "Enumeration",
    "entityType": {
     "id": "dt56spnul01p",
     "name": "Publication",
     "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
    },
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/mfpco2wx3zqp",
    "options": {
     "enumerationValues": [
      "Planned",
      "Submitted",
            "Accepted"
          ]
        }
      }
    ],
    "taskDefinitions": [],
    "workspace": {
      "id": "10sasph8dvxhg",
      "name": "Publication Management",
      "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
    },
    "entitiesCount": 6,
    "structurednessMetric": 1,
    "id": "dt56spnul01p",
    "versions": ["..."],
    "consistencyMetric": 1,
    "derivedAttributeDefinitions": [],
    "name": "Publication",
    "allowFreeAttributes": false,
    "associations": [
      {
        "id": "1knrlnoj7kjer",
        "versions": [],
        "readOnly": false,
        "name": "acceptedIn",
        "multiplicity": "any",
        "attributeType": "Link",
        "entityType": {
          "id": "dt56spnul01p",
          "name": "Publication",
          "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
        },
        "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer",
        "options": {
          "entityType": {
            "id": "19c48jgd4b8m2",
            "name": "Conference",
            "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
          },
          "resourceType": "entities"
        }
      }
    ],
    "incomingAssociations": [],
    "customFunctions": [],
    "mayEdit": true,
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  {
    "processes": [],
    "attributeDefinitions": [
      {
        "id": "5ybnu95cmhfq",
        "versions": [],
        "readOnly": false,
        "name": "Rank",
        "multiplicity": "maximalOne",
        "attributeType": "Enumeration",
        "entityType": {
          "id": "19c48jgd4b8m2",
          "name": "Conference",
          "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
        },
        "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/5ybnu95cmhfq",
        "options": {
          "enumerationValues": [
            "A",
            "B",
            "C"
          ]
        }
      },
      {
        "id": "m4kghlbyf1e5",
        "versions": [],
        "readOnly": false,
        "name": "Year",
        "multiplicity": "maximalOne",
        "attributeType": "Number",
        "entityType": {
          "id": "19c48jgd4b8m2",
          "name": "Conference",
          "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
        },
        "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/m4kghlbyf1e5"
      }
    ],
    "taskDefinitions": [],
    "workspace": {
      "id": "10sasph8dvxhg",
      "name": "Publication Management",
      "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
    },
    "entitiesCount": 5,
    "structurednessMetric": 1,
    "id": "19c48jgd4b8m2",
    "versions": ["..."],
    "consistencyMetric": 1,
    "derivedAttributeDefinitions": [],
    "name": "Conference",
    "allowFreeAttributes": false,
    "associations": [],
    "incomingAssociations": [
      {
        "id": "1knrlnoj7kjer",
        "versions": [],
        "readOnly": false,
        "name": "acceptedIn",
        "multiplicity": "any",
        "attributeType": "Link",
        "entityType": {
          "id": "dt56spnul01p",
          "name": "Publication",
          "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
        },
        "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer",
        "options": {
          "entityType": {
            "id": "19c48jgd4b8m2",
            "name": "Conference",
            "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
          },
          "resourceType": "entities"
        }
      }
    ],
    "customFunctions": [],
    "mayEdit": true,
    "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
  }
]
```

3.7.2. Looking at the *publication* entity type (GET)

To retrieve details about a single entity we add the respective id to the URI.

Request:

BASH

```bash
GET https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p
```

Response:

JSON

```json
{
  "processes": [],
  "attributeDefinitions": [
    {
      "id": "2gegqgdueyqd",
      "versions": [],
      "readOnly": false,
      "name": "Abstract",
      "multiplicity": "maximalOne",
      "attributeType": "Text",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/2gegqgdueyqd"
    },
    {
      "id": "1knrlnoj7kjer",
      "versions": [],
      "readOnly": false,
      "name": "acceptedIn",
      "multiplicity": "any",
      "attributeType": "Link",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer",
      "options": {
        "entityType": {
          "id": "19c48jgd4b8m2",
          "name": "Conference",
          "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
        },
        "resourceType": "entities"
      }
    },
    {
      "id": "18b1hu00gnxdi",
      "versions": [],
      "readOnly": false,
      "name": "authors",
      "multiplicity": "any",
      "attributeType": "Link",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/18b1hu00gnxdi",
      "options": {
        "resourceType": "users"
      }
    },
    {
      "id": "mfpco2wx3zqp",
      "versions": [],
      "readOnly": false,
      "name": "Status",
      "multiplicity": "maximalOne",
      "attributeType": "Enumeration",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/mfpco2wx3zqp",
      "options": {
        "enumerationValues": [
          "Planned",
          "Submitted",
          "Accepted"
        ]
      }
    }
  ],
  "taskDefinitions": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "entitiesCount": 6,
  "structurednessMetric": 1,
  "id": "dt56spnul01p",
  "versions": ["..."],
  "consistencyMetric": 1,
  "derivedAttributeDefinitions": [],
  "name": "Publication",
  "allowFreeAttributes": false,
  "associations": [
    {
      "id": "1knrlnoj7kjer",
      "versions": [],
      "readOnly": false,
      "name": "acceptedIn",
      "multiplicity": "any",
      "attributeType": "Link",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer",
      "options": {
        "entityType": {
          "id": "19c48jgd4b8m2",
          "name": "Conference",
          "href":
"https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
        },
        "resourceType": "entities"
      }
    }
  ],
  "incomingAssociations": [],
  "customFunctions": [],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
}
```

3.8. Getting attribute definitions

3.8.1. Looking at attribute definitions of entity type (GET)

For our entity types we have created custom attribute definitions which might also be useful to have another look at. To query these we add the *attributeDefinitions* keyword:

Request:

BASH

```
GET
https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/attributeDefinition
s
```

Response:

```json
[
  {
    "id": "2gegqgdueyqd",
    "name": "Abstract",
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/2gegqgdueyqd"
  },
  {
    "id": "1knrlnoj7kjer",
    "name": "acceptedIn",
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/1knrlnoj7kjer"
  },
  {
    "id": "18b1hu00gnxdi",
    "name": "authors",
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/18b1hu00gnxdi"
  },
  {
    "id": "mfpco2wx3zqp",
    "name": "Status",
    "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/mfpco2wx3zqp"
  }
]
```

3.9. Getting entities

Obviously, one of the most common tasks is to retrieve all the entities in a workspace or of a certain type. This can either be done by specifically searching for entities through the search endpoint (see Searching) or by querying by entity type, workspace or id of a single entity.

3.9.1. Looking at all existing entities in workspace (GET)

Request:

```bash
GET https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg/entities
```

Response:

```JSON
[{
    "id": "bqr7jmg37vqt",
    "name": "Publication Management - Home",
    "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt"
  },
  {
    "id": "n6eidlmr0by7",
    "name": "International Conference on Software Engineering (ICSE)",
    "href": "https://server.sociocortex.com/api/v1/entities/n6eidlmr0by7"
  },
  {
    "id": "127trtufzdgcw",
    "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in
Collaborative Information Systems",
    "href": "https://server.sociocortex.com/api/v1/entities/127trtufzdgcw"
}]
```

3.9.2. Looking at all existing entities of entity type *publication* (GET)

Request:

```BASH
GET https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/entities
```

Response:

```JSON
[{
    "id": "127trtufzdgcw",
    "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in
Collaborative Information Systems",
    "href": "https://server.sociocortex.com/api/v1/entities/127trtufzdgcw"
}]
```

3.9.3. Looking at single entity (GET)

Request:

```BASH
GET https://server.sociocortex.com/api/v1/entities/127trtufzdgcw
```

Response:

```JSON
{
  "files": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "children": [],
  "tasks": [],
  "entityType": {
    "id": "dt56spnul01p",
    "name": "Publication",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  "incomingReferences": {},
  "creator": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "id": "127trtufzdgcw",
  "versions": ["..."],
  "createdAt": "2016-07-24T11:27:26.000Z",
  "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in
Collaborative Information Systems",
  "lastModifier": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "permissions": {
    "writers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "readers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ]
  },
  "attributes": [
    {
      "id": "7mgjht1rkna8",
      "values": [
        {
          "id": "1hda5st2csfom",
          "name": "Thomas Reschenhofer",
          "href": "https://server.sociocortex.com/api/v1/users/1hda5st2csfom"
        }
      ],
      "name": "authors",
      "href": "https://server.sociocortex.com/api/v1/attributes/7mgjht1rkna8"
    },
    {
      "id": "iqqlpxfbnr8f",
      "values": [
        {
          "id": "n6eidlmr0by7",
          "name": "International Conference on Software Engineering (ICSE)",
          "href": "https://server.sociocortex.com/api/v1/entities/n6eidlmr0by7"
        }
      ],
      "name": "acceptedIn",
      "href": "https://server.sociocortex.com/api/v1/attributes/iqqlpxfbnr8f"
    },
    {
      "id": "rvzij9nn9of4",
      "values": [
        "Accepted"
      ],
      "name": "Status",
      "href": "https://server.sociocortex.com/api/v1/attributes/rvzij9nn9of4"
    },
    {
      "id": "ys69v6zjfz54",
      "values": [
        "Today's enterprises have to align their information systems continuously with their dynamic business and IT environment. ..."
      ],
      "name": "Abstract",
      "href": "https://server.sociocortex.com/api/v1/attributes/ys69v6zjfz54"
    }
  ],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entities/127trtufzdgcw",
  "lastModifiedAt": "2016-07-24T11:27:26.000Z"
}
```

3.10. Tasks and task definitions

Attributes of entity types can be linked to tasks, which means that for instance a task is not yet completed until a certain attribute it is linked to is set. Task definitions create those links between attributes and tasks to be fulfilled. In the following, we will first create a task definition for the previously defined *status* attribute of the *publication* entity type. Then, we create a new *publication* entity whose status is not set to any value yet. Last, the status of new entity is updated and set to *accepted*. This will fulfill the respective task.

3.10.1. Create task definition (POST)

For creating the task definition, we require the id of the previously defined entity type *publication* as well as the id of the attribute definition of *Status* which it contains. You need to replace them with your own values in order to make the requests work.

Request:

```bash
POST https://server.sociocortex.com/api/v1/taskDefinitions
BODY
{
  "name": "Publication Status",
  "entityType": {
    "id":"dt56spnul01p"
  },
  "attributeDefinitions": [
    {
      "id":"mfpco2wx3zqp"
    }
  ]
}
```

Response:

```json
{
  "attributeDefinitions": [
    {
      "id": "mfpco2wx3zqp",
      "versions": [],
      "readOnly": false,
      "name": "Status",
      "multiplicity": "maximalOne",
      "attributeType": "Enumeration",
      "entityType": {
        "id": "dt56spnul01p",
        "name": "Publication",
        "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
      },
      "href":
"https://server.sociocortex.com/api/v1/attributeDefinitions/mfpco2wx3zqp",
      "options": {
        "enumerationValues": [
          "Planned",
          "Submitted",
          "Accepted"
        ]
      }
    }
  ],
  "id": "875jp376lljp",
  "satisfies": [],
  "name": "Publication Status",
  "tasks": [],
  "preconditions": [],
  "entityType": {
    "id": "dt56spnul01p",
    "name": "Publication",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p"
  },
  "href": "https://server.sociocortex.com/api/v1/taskDefinitions/875jp376lljp"
}
```

3.10.2. Create entity *publication* (POST)

Next, we create a new publication where we purposefully do not set all the defined attributes yet.

Request:

```bash
POST https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p/entities
BODY
{
    "name": "Ha15a - Empowering End-Users to Collaboratively Structure Processes
for Knowledge Work",
    "attributes":[
        {
            "name":"acceptedIn",
            "values":[]
        },
        {
            "name":"Abstract",
            "values": ["Knowledge work is becoming the predominant type of work in
many countries and is involved in the most important processes in organizations.
..."]
        },
        {
            "name":"Status",
            "values":[]
        },
        {
            "name":"authors",
            "values":[]
        }
    ]
}
```

Response:

```json
{
  "statusCode": 406,
  "message": "cannot make persistent because is not consistent: Ha15a - Empowering
End-Users to Collaboratively Structure Processes for Knowledge Work
[uid=page/1ig6fs784zcr0, state: transient] invalid features: \"parent\", value:
\"null\", error message: \"No space found.\", \"space\", value: \"null\", error
message: \"Cannot be null.\"",
  "cause": "IllegalStateException",
  "handler": "EntitiesByEntityTypeHandler"
}
```

3.10.3. Updating status of publication (PUT)

Task before update:

Request:

```bash
*Response:*
[source,json]
```

Updating status:

Request:

```bash
*Response:*
[source,json]
```

Task after update:

Request:

```bash
*Response:*
[source,json]
```

3.11. Updating entity types / entities

3.11.1. Updating name of entity (PUT)

The easiest way to demonstrate how entities are updated is by sending a HTTP PUT request on an existing *conference* which is supposed to change the name of the conference. Let's assume we just want to get rid of the abbreviation of the conference (ICSE).

Request:

```bash
{
    "name":"International Conference on Software Engineering"
}
```

Response:

```json
{
  "files": [],
  "workspace": {
    "id": "10sasph8dvxhg",
    "name": "Publication Management",
    "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg"
  },
  "children": [],
  "tasks": [],
  "entityType": {
    "id": "19c48jgd4b8m2",
    "name": "Conference",
    "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2"
  },
  "incomingReferences": {
    "acceptedIn": [
      {
        "id": "127trtufzdgcw",
        "name": "Re16a - Lessons Learned in Aligning Data and Model Evolution in Collaborative Information Systems",
        "href": "https://server.sociocortex.com/api/v1/entities/127trtufzdgcw"
      }
    ]
  },
  "creator": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "id": "n6eidlmr0by7",
  "versions": ["..."],
  "createdAt": "2016-07-24T11:12:00.000Z",
  "name": "International Conference on Software Engineering",
  "lastModifier": {
    "id": "$USER_ID",
    "name": "$NAME",
    "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
  },
  "permissions": {
    "writers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ],
    "readers": [
      {
        "id": "$USER_ID",
        "name": "$NAME",
        "href": "https://server.sociocortex.com/api/v1/users/$USER_ID"
      }
    ]
  },
  "attributes": [
    {
      "id": "pme5rnjsvx5z",
      "values": [
        "A"
      ],
      "name": "Rank",
      "href": "https://server.sociocortex.com/api/v1/attributes/pme5rnjsvx5z"
    },
    {
      "id": "voa7iqjet7pf",
      "values": [
        2016
      ],
      "name": "Year",
      "href": "https://server.sociocortex.com/api/v1/attributes/voa7iqjet7pf"
    }
  ],
  "mayEdit": true,
  "href": "https://server.sociocortex.com/api/v1/entities/n6eidlmr0by7",
  "lastModifiedAt": "2016-07-24T18:18:06.000Z"
}
```

3.12. Request/Response

TBD …

4. Permisssions

5. Query Parameters

- query parameters - list vs. single (Attribute alle dabei)
- attributes (leer lassen nicht erklären) (* oder , getrennt)
- meta
- content
- task

6. Searching

6.1. Searching for entities

6.1.1. Searching for entities in a workspace (GET)

Request:

```bash
GET https://server.sociocortex.com/api/v1/searchResults?workspace=10sasph8dvxhg
```

Response:

```json
{
  "results": [
    {
      "id": "dt56spnul01p",
      "name": "Publication",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p",
      "resourceType": "entityTypes"
    },
    {
      "id": "1bbxmsluu87f7",
      "name": "Text Page",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/1bbxmsluu87f7",
      "resourceType": "entityTypes"
    },
    {
      "id": "bqr7jmg37vqt",
      "name": "Publication Management - Home",
      "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt",
      "resourceType": "entities"
    },
    {
      "id": "10sasph8dvxhg",
      "name": "Publication Management",
      "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg",
      "resourceType": "workspaces"
    },
    {
      "id": "19c48jgd4b8m2",
      "name": "Conference",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2",
      "resourceType": "entityTypes"
    },
    {
      "id": "15qr31m7ewuje",
      "name": "A sample paper",
      "href": "https://server.sociocortex.com/api/v1/entities/15qr31m7ewuje",
      "resourceType": "entities"
    },
    {
      "id": "24ixbhft7i4b",
      "name": "A sample conference",
      "href": "https://server.sociocortex.com/api/v1/entities/24ixbhft7i4b",
      "resourceType": "entities"
    }
  ],
  "totalCount": 7
}
```

6.1.2. Searching for entities with filter parameters (GET)

Request:

BASH

```bash
GET https://server.sociocortex.com/api/v1/searchResults?text=publication&workspace=10sasph8dvxhg&orderBy=alphabetically
```

Response:

JSON

```json
{
  "results": [
    {
      "highlightedDisambiguation": "/<span class=\"sc-search-highlight\">Publication</span> Management/_",
      "id": "24ixbhft7i4b",
      "name": "A sample conference",
      "href": "https://server.sociocortex.com/api/v1/entities/24ixbhft7i4b",
      "resourceType": "entities"
    },
    {
      "highlightedDisambiguation": "/<span class=\"sc-search-highlight\">Publication</span> Management/_",
      "id": "15qr31m7ewuje",
      "name": "A sample paper",
      "href": "https://server.sociocortex.com/api/v1/entities/15qr31m7ewuje",
      "resourceType": "entities"
    },
    {
      "highlightedDisambiguation": "<span class=\"sc-search-highlight\">Publication</span> Management",
      "id": "19c48jgd4b8m2",
      "name": "Conference",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/19c48jgd4b8m2",
      "resourceType": "entityTypes"
    },
    {
      "highlightedDisambiguation": "<span class=\"sc-search-highlight\">Publication</span> Management",
      "id": "dt56spnul01p",
      "highlightedName": "<span class=\"sc-search-highlight\">Publication</span>",
      "name": "Publication",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/dt56spnul01p",
      "resourceType": "entityTypes"
    },
    {
      "id": "10sasph8dvxhg",
      "highlightedName": "<span class=\"sc-search-highlight\">Publication</span> Management",
      "name": "Publication Management",
      "href": "https://server.sociocortex.com/api/v1/workspaces/10sasph8dvxhg",
      "resourceType": "workspaces"
    },
    {
      "highlightedDisambiguation": "<span class=\"sc-search-highlight\">Publication</span> Management",
      "id": "bqr7jmg37vqt",
      "highlightedName": "<span class=\"sc-search-highlight\">Publication</span> Management - Home",
      "name": "Publication Management - Home",
      "href": "https://server.sociocortex.com/api/v1/entities/bqr7jmg37vqt",
      "resourceType": "entities"
    },
    {
      "highlightedDisambiguation": "<span class=\"sc-search-highlight\">Publication</span> Management",
      "id": "1bbxmsluu87f7",
      "name": "Text Page",
      "href": "https://server.sociocortex.com/api/v1/entityTypes/1bbxmsluu87f7",
      "resourceType": "entityTypes"
    }
  ],
  "totalCount": 7
}
```

Last updated 2016-07-24 20:22:13 CEST