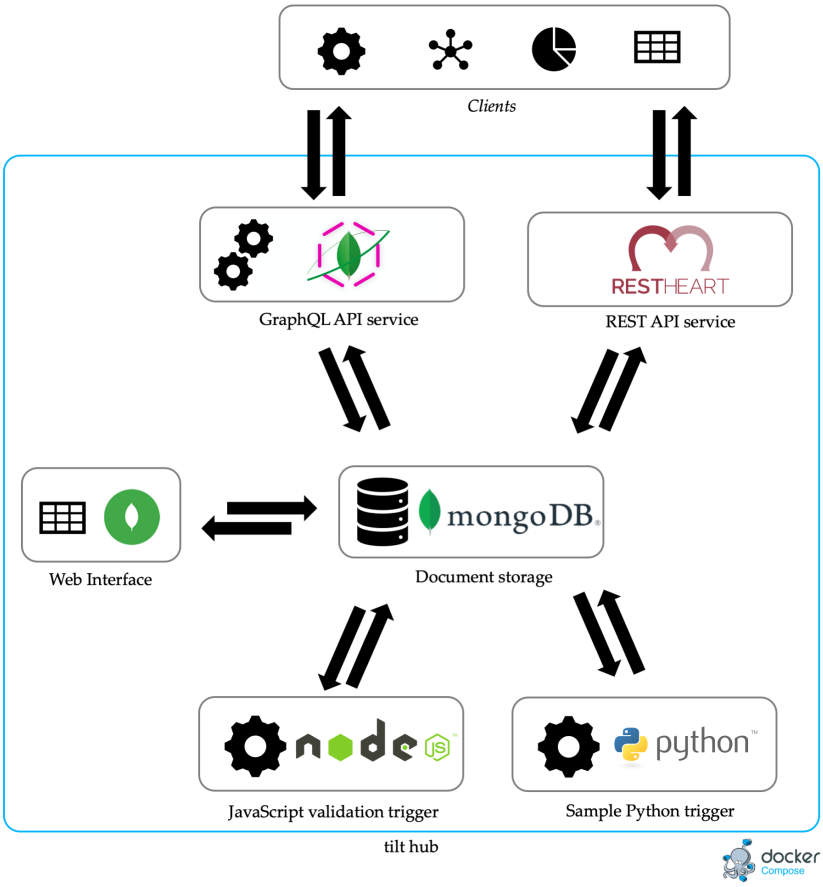
tilt-hub: Transparency Information Language Document Hub

tilt-hub is a document storage for transparency information specified in the *Transparency Information Language* and is part of the *Transparency Information Language and Toolkit*.

This document storage comprises five services in order to perform CRUD operations on transparency information documents and serves for clients, that either only want plain information about GDPR compliance of a data controller or carry out arbitrary complex research/statistics/analysis tasks.



Usage

Simply run

```
docker-compose up [-d]
```

After some startup time, there will be the following five containers:

```
docker-compose ps
    Name                   Command               State
    ---
mongo           docker-entrypoint.sh --con ...   Up      0.0.0.0:27017->27017/tcp, 0.0.0.0:2
mongo-express   tini -- /docker-entrypoint ...   Up      0.0.0.0:8081->8081/tcp
mongoke         dumb-init -- /entrypoint.s ...   Up      0.0.0.0:8082->80/tcp
restheart       java -Dfile.encoding=UTF-8 ...   Up      4443/tcp, 8009/tcp, 0.0.0.0:8080->8
triggers        docker-entrypoint.sh node  ...   Up
```

Mongo DB

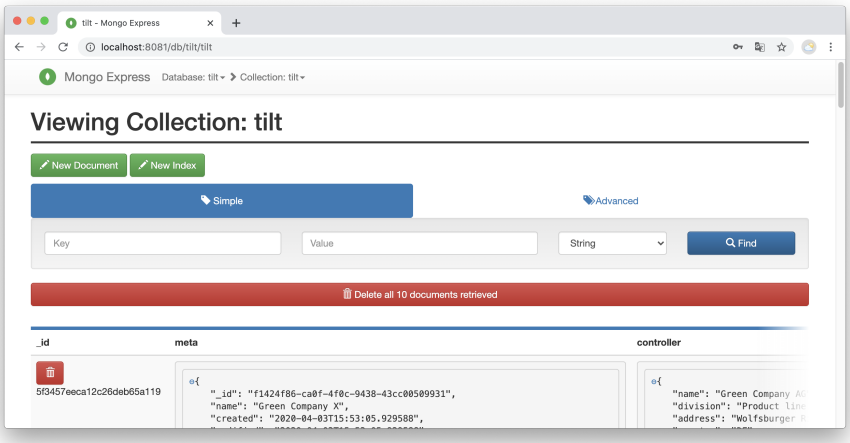
The mongo database is available via mongodb://root:SuperSecret@mongo. A client application written in Python could look like:

```
from pymongo import MongoClient
from pprint import pprint

client = MongoClient('mongodb://root:SuperSecret@mongo:27017/?authSource=admin&readPreferenc


filter={
    'controller.name': 'Green Company'
}

result = client['tilt']['tilt'].find(
  filter=filter
)

for document in result:
    pprint(document)
```

Mongo Express

A graphical interface to the database can be reached under http://localhost:8081/.

```
Username: admin
Password: SuperSecret
```



REST API

In order to retrieve all documents from the database `tilt` and collection `tilt`.

```
http GET "localhost:8080/tilt/tilt" -a admin:secret
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Sun, 16 Aug 2020 21:36:05 GMT
...

[
    {
        "_etag": {
            "$oid": "5f39a2cb5937e6719cbb3bed"
        },
        "_id": {
            "$oid": "5f39a2cb8f9a233727fb7fa2"
        },
        "accessAndDataPortability": {
            "administrativeFee": {
                "amount": 0,
                "currency": "EUR"
            },
...
```

GraphQL API

Use the *GraphiQL* interface running at http://localhost:8082 to perform GraphQL operations. To return the `meta.name` field of all documents run:

```
{
  TiltNodes {
    edges {
      node {
        meta {
          name
        }
      }
    }
  }
}
```

Triggers

In this basic scenario there will be a JavaScript trigger microservice which validates all documents on every change that occurs using the tilt-schema.

The output of the validation can be obtained directly from the logs:

```
docker-compose logs -f triggers
```

```
triggers         | Downloaded schema from https://raw.githubusercontent.com/Transparency-Inf
triggers         | Trigger on tilt collection initialized successfully!
triggers         | Document changes detected!
triggers         | tilt schema validation successful!
```

Similarly, there could by a Python based trigger which is located here. In order to use it, configure the docker-compose.yml accordingly.

Author

Elias Grünewald

License

GNU General Public License, Version 3