OBA: An Ontology-Based Framework for Creating REST APIs for Knowledge Graphs

Abstract of your accepted paper:

In recent years, Semantic Web technologies have been increasingly adopted by researchers, industry and public institutions to describe and link data on the Web, create web annotations and consume large knowledge graphs like Wikidata and DBpedia. However, there is still a knowledge gap between ontology engineers, who design, populate and create knowledge graphs; and web developers, who need to understand, access and query these knowledge graphs but are not familiar with ontologies, RDF or SPARQL. In this paper we describe the Ontology-Based APIs framework (OBA), our approach to automatically create REST APIs from ontologies while following RESTful API best practices. Given an ontology (or ontology network) OBA uses standard technologies familiar to web developers (OpenAPI Specification, JSON) and combines them with W3C standards (OWL, JSON-LD frames and SPARQL) to create maintainable APIs with documentation, unit tests, automated validation of resources and clients (in Python, Javascript, etc.) for non-Semantic Web experts to access the contents of a target knowledge graph. We showcase OBA with three examples that illustrate the capabilities of the framework for different ontologies.

Link to a public github/bitbucket/gitlab repository:

https://github.com/KnowledgeCaptureAndDiscovery/OBA

A copy of the README file from your repository, containing information on the platform required to run your code.

Readme file can be downloaded at https://github.com/KnowledgeCaptureAndDiscovery/OBA/blob/master/README.md (or see additional materials). Note that additional documentation may be found at: https://oba.readthedocs.io/en/latest/

Information on how to obtain the data needed to evaluate your submission. The data must be made openly available:

OBA is a tool for generating APIs from ontologies. Therefore, to test it out you only need **an ontology** in OWL and a configuration file. You may choose some of the sample configurations, which either use local or ontologies on their respective URLs. For instance, in https://github.com/KnowledgeCaptureAndDiscovery/OBA/tree/master/examples/modelcatalog_reduced you will find a configuration file with a local ontology. You can try your own ontology as well.

Once the OpenAPI specification has been created, you may test it online with an editor such as https://editor.swagger.io/

To create your own server with OBA, **you need a target knowledge graph** to test the API. We recommend using an existing one such as DBpedia. https://github.com/KnowledgeCaptureAndDiscovery/OBA/tree/master/examples/dbpedia has some examples that will help set it up. The documentation provides details on how to set up and generate an OBA API server from an OpenAPI specification: https://oba.readthedocs.io/en/latest/server/

Environment

We have tested OBA in Windows and Unix systems. The only requirement for running the tool is Java 8. For generating the server, a Docker installation is needed (for Windows, users will need Windows10 Pro). For the Docker installation of the server, we recommend at least 4 GB of RAM. The openapi server images needs 328 MB. The first time the server executes will take some time (depending on the complexity of your API).

System

Most of these questions are already answered in the documentation of the tool and the readme file. Here we will add only the pointers to such documentation.

How to obtain the system? Follow the instructions listed on the readme file. You can download the binary from the latest release page; or build the project yourself.

Note that for running the generated API as a server, Docker is needed.

How to configure the environment if need be (e.g., environment variables, paths)?N/A

How to compile the system? (existing compilation options should be mentioned and why they are needed)

The project uses Java 8 (or higher) and Maven.

How to use the system? (What are the configuration options and parameters to the system and where to set them?)

The tool has a single command-line option:

```
java -jar oba-*-jar-with-dependencies.jar -c config.yaml
```

How to make sure that the system is installed correctly?

Type: `java -jar oba-*-jar-with-dependencies.jar`

You should see the following message:

```
usage: utiConfiguration filelity-name

 -c,--config <arg>   configuration file path
```

For the server generation, additional instructions may be found at: https://oba.readthedocs.io/en/latest/server/

Tests

Tests are not needed for making OBA run. However, we have unit tests at:

src/test/java/edu/isi/oba/

These are Java unit tests, so by opening the project in an IDE (Eclipse/NetBeans/ IntelliJ IDEA) anyone should be able to re-run the tests.

The API has also integration tests. These are run through python in the server container. More information can be found at:

https://oba.readthedocs.io/en/latest/test/

The open license under which you plan to provide the source code and data.

Apache-2.0 License