

GraphQL Servers generation from R2RML mappings with morph-GraphQL (DEMO)

1st Ahmad Alobaid
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
aalobaid@fi.upm.es

2nd Freddy Priyatna
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
fpriyatna@fi.upm.es

3rd David Chaves-Fraga
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
dchaves@fi.upm.es

4th Oscar Corcho
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
ocorcho@fi.upm.es

Abstract—The adoption of GraphQL is on the rise, many companies and institutes are adopting it due to its ease of use, ease of maintenance, and hide the complexity from the user. Such advantages come from using a unified global schema and mapping it to the underlying data sources. The semantic web community has already adopted a similar way to map different data resources (e.g., R2RML). We present a novel way of generating GraphQL server from R2RML mappings.

Index Terms—GraphQL, R2RML, Wrapper, Adapter

I. INTRODUCTION

Facebook developed GraphQL¹ as an alternative to REST and made it open source to be used by the public in 2015. A GraphQL server consists of multiple components: schema, resolvers, and data sources. The GraphQL schema is the exposed view - an interface that the user (the person who writes queries) can use to access the underlying data sources. GraphQL resolvers are written codes (in a programming language) to link fields in the data sources to the exposed schema. The data sources are the where the data are stored and can be retrieved from such as a Relation Database (e.g., MySQL).

R2RML [1], published in 2012 by the RDB2RDF W3C Working Group, is a W3C recommendations for transforming the content of relational databases into RDF datasets. It allows the users to specify rules of how this transformation being done, such as how the URIs be generated, or which columns to be used in the transformation rules.

II. MORPH-GRAPHQL

In [2] we introduce morph-GraphQL², that takes as its input R2RML mappings and generates the corresponding GraphQL server. GraphQL engine interprets queries written in GraphQL and use the corresponding resolvers to fetch the data from the data sources. This workflow is shown in Fig. 1. R2RML

will be the input to morph-GraphQL and it will output the corresponding GraphQL resolvers and schema. The schema, resolvers and the data source are the input to the GraphQL engine.

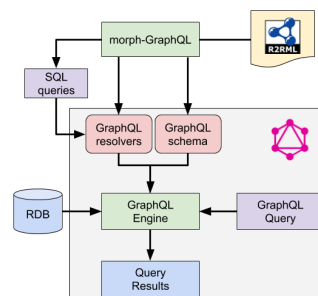


Fig. 1. Workflow of morph-GraphQL

In this demo we will show how we create R2RML mappings for the CSV files³ containing Star Wars data that is used as the example in the reference implementation⁴. Then we will use morph-GraphQL to generate GraphQL schema and resolvers from created mappings. Finally, we will evaluate some queries corresponding to the Star Wars example.

Acknowledgement: This work is supported by the Spanish Ministerio de Economía, Industria y Competitividad and EU FEDER funds under the DATOS 4.0: RETOS Y SOLUCIONES - UPM Spanish national project (TIN2016-78011-C4-4-R)

REFERENCES

- [1] S. Das, S. Sundara, and R. Cyganiak, “R2RML: RDB to RDF Mapping Language,” <https://www.w3.org/TR/r2rml/>, accessed: 2018-12-07.
- [2] F. Priyatna, D. Chaves-Fraga, A. Alobaid, and O. Corcho, “morph-GraphQL: GraphQL resolvers generation from R2RML mappings.” in *SEKE*, 2019.

¹<https://graphql.org/>

²<https://github.com/oeg-upm/morph-graphql>, deployed at <http://graphql.morph.oeg-upm.net>

³<https://github.com/oeg-upm/morph-graphql/tree/master/examples/starwars>

⁴<https://github.com/graphql/graphql-js>