### Enabling GraphQL Adoption

2<sup>nd</sup> Vienna Software Seminar

Erik Wittern - IBM Research @ErikWittern witternj@us.ibm.com

### What is GraphQL?

- A query language for networked APIs...
- ...and a **runtime** for servers to fulfill queries
- Specification + reference implementations
- Clients send type-checked queries, servers respond with requested data:



## Why GraphQL?

- Serve diverse & changing client requirements with one API *providers*
- API evolution *providers / consumers* 
  - Schema extensions
  - Deprecation messages built-in
- Resource composition (APIs, DBs, Cloud functions...) providers / consumers
- Operations and performance *providers / consumers* 
  - Fewer roundtrips
  - No over-fetching
- Great developer experience *consumers* 
  - Type-checked API (auto-completion, validation)
  - Tooling like Graph*i*QL
  - Documentation built-in



# Challenge: How can we help REST APIs providers in adopting GraphQL?





### An example:





### **Dissemination: OpenAPI-to-GraphQL**

Generating GraphQL-Wrappers for REST(-like) APIs

Erik Wittern<sup>1</sup>, Alan Cha<sup>2\*</sup>, and Jim A. Laredo<sup>1</sup>

<sup>1</sup> IBM T.J Watson Research Center, Yorktown Heights, NY, 10598 USA witternj@us.ibm.com, laredoj@us.ibm.com <sup>2</sup> Columbia University, New York, NY, 10025 USA ac3805@columbia.edu

Abstract. GraphQL is a query language and thereupon-based paradigm for implementing web Application Programming Interfaces (APIs) for client-server interactions. Using GraphQL, clients define precise, nested data-requirements in typed queries, which are resolved by servers against (possibly multiple) backend systems, like databases, object storages, or other APIs. Clients receive only the data they care about, in a single request. However, providers of existing REST(-like) APIs need to implement additional GraphQL interfaces to enable these advantages. We here assess the feasibility of automatically generating GraphQL wrappers for existing REST(-like) APIs. A wrapper, upon receiving GraphQL queries, translates them to requests against the target API. We discuss the challenges for creating such wrappers, including dealing with data sanitation, authentication, or handling nested queries. We furthermore present a prototypical implementation of OASGraph. OASGraph takes as input an OpenAPI Specification (OAS) describing an existing REST(like) web API and generates a GraphQL wrapper for it. We evaluate OASGraph by running it, as well as an existing open source alternative, against 959 publicly available OAS. This experiment shows that OASGraph outperforms the existing alternative and is able to create a GraphQL wrapper for 89.5% of the APIs - however, with limitations in many cases. A subsequent analysis of errors and warnings produced by OASGraph shows that missing or ambiguous information in the assessed OAS hinders creating complete wrappers. Finally, we present a use case of the IBM Watson Language Translator API that shows that small changes to an OAS allow OASGraph to generate more idiomatic and more expressive GraphQL wrappers

### 1 Introduction

Page 6

Created by Facebook in 2012 and released as open source in 2015, GraphQLis a query language and thereupon-based paradigm for building web Application Programming Interfaces (APIs) for client-server interactions. According to the lore<sup>3</sup>, Facebook created GraphQL in response to observed issues with more

**2018 ICWE paper** 

https://arxiv.org/abs/1809.08319

\* Work performed during Summer Internship at IBM Research in 2017 <sup>3</sup> https://code.facebook.com/posts/1691455094417024/ graphql-a-data-query-language/

| IBM / openapi-to-graphqi     Code     Issues 34 1 Pul                             | l requests 5 🛛 🗐 Projec                             | ts 0 ■ Wiki 『           | 3 O Unwate<br>Security               | nsights ⊅    | Tunstar Settings        | 420 ¥ Fo       | rk 3    |
|---|---|-------------------------|--------------------------------------|--------------|-------------------------|----------------|---------|
| Translate APIs described by Open<br>graphql rest-api translation<br>Manage topics | nAPI Specifications (OA<br>wrapper openapi-specific | S) into GraphQL ht      | tps://developer<br>vagger-to-graphql | .ibm.com/op  | pen/projec<br>hql opena | pi-to-graphql  | Edi     |
| ⑦ <b>450</b> commits  | ₽ 2 branches  | es 🛇 53 releases 🎎 9 c  |                                      |              | ontributors বু          |                |         |
| Branch: master - New pull request   |   | (                       | Create new file                      | Upload files | Find File               | Clone or dov   | wnload  |
| In Alan-Cha Publish   |   |                         |                                      |              | Latest comm             | it 78feac0 5 c | lays ag |
| .github/ISSUE_TEMPLATE  | Update issue  | e templates             |                                      |              |                         | 10 mon         | ths ag  |
| docs  | Changed vic   | leo ID.                 |                                      |              |                         | las            | t mont  |
| packages  | Publish   |                         |                                      |              |                         | 5 d            | ays ag  |
| juitignore  | add tests for                                       | r PR#97                 |                                      |              |                         | 6 mon          | ths ag  |
| :travis.yml   | Clean up Tra  | avis config             |                                      |              |                         | las            | t mont  |
| CONTRIBUTING.md   | Rename rep  | ository                 |                                      |              |                         | 2 mon          | ths ag  |
| LICENSE.md  | Updated lice  | ense to confirm with M  | IT template                          |              |                         | 2 mon          | ths ag  |
| README.md   | Remove sub  | Operations and fix link |                                      |              |                         | 27 d           | ays ag  |
| 🖹 lerna.json  | Prettier JSO  | N files                 |                                      |              |                         | 2 mon          | ths ag  |
| package.json  | Tests in Type                                       | escript                 |                                      |              |                         | 20 d           | ays ag  |
| I README.md   |   |                         |                                      |              |                         |                | 6       |
| Isst commit last thursday build   | passing chat on gitter                              |                         |                                      |              |                         |                | đ       |

### https://github.com/ibm/openapi-to-graphql



### **Challenge: How to manage GraphQL APIs?**

- **Goal**: provide management capabilities for GraphQL (think threat prevention, rate limiting, access control etc.)
- **Challenge**: highly diverse, client-defined queries

```
query fetchAllTheData {
    users (limit: 1000) {
        orders (first: 1000) {
            paymentDetails { status }
        }
    }
}
Equivalent to ~1000s
of REST requests!
```

• **Solution approach**: static query analysis – i.e., try to understand what a query intends to do prior to execution



### **Static analysis of a query**

```
query {
   users (first: 5) {
      name
      pastEmployers (limit: 2) {
      name
      }
   }
}
```





### **GraphQL Management: Architecture**





### "Enabling GraphQL Adoption" by...

- Helping REST API providers quickly experiment with GraphQL
  - OpenAPI-to-GraphQL @ <u>https://github.com/ibm/openapi-to-graphql</u>
- Offering GraphQL API management
  - Prototypical implementation, evaluation, and ongoing productization with API Connect & DataPower Gateway
- Addressing further challenges in the future:
  - Composing...
    - ...diverse backend resources
    - ...GraphQL schemas (schema stitching, federation)
  - Transactions
  - (Distributed) schema development and maintenance
  - Etc.



Thank you!

