The benefits of GraphQL for more efficient APIs: A better version of REST

Mathias Wiesbauer, *George Mason University, June 2019*

Contents

1 **Introduction** 2

2 **The problem with REST** 2
  2.1 Too much data 2
  2.2 Too many requests 3

3 **Benefits of GraphQL** 3

4 **Conclusion** 3

List of Figures

1 API design for client server communication (Stubailo, 2017) 3

With the continued growth of internet usage application developers had to support communication with mobile devices with varying screen sizes, desktop computers and a variety of server appplications. Each of those interfaces had different requirements and the developers had to maintain an API for each of the interfaces. This led to a lot of overhead and cost in developing and maintaining the APIs with the most popular one being REST.
With the introduction of GraphQL many of these issues can now be solved. We will show how GraphQL can make your organizations API development more efficient.

1 Introduction

The continued growth of the internet with the introduction of smart devices such as phones, and internet of things (IoT) devices has increased the need for data exchange with service providers. Thermostats have to send their current temperature values to a server and the owner can then fetch the current temperature on demand via his mobile phone. Social media has grown exponentially and hundreds of millions of people share information via online services.

A lot of this communication happenes via REST (REpresentational State Transfer) APIs. REST was introduced by Fielding (2000) in his dissertation where he describes REST as a set of principles through which resources can be accessed. REST provides an abstraction layer in client server communication enabling the client to exchange data with the server without communicating with the database (Helgason, 2017, p.5). REST has contributed largely to the widespread use of web-services by becoming the defacto standard (Harrison, 2018, p.50).

With this increase in popularity certain shortcomings became apparent and Facebook started developing a better version of REST called GrapQL addressing some of the shortcomings.

GraphQL has been developed to be a better version of REST for a more modern appaorch to API design. It has been developed and used at Facebook before being released as open source software in 2015 (Harrison, 2018).

2 The problem with REST

2.1 Too much data

In REST each resource has a Uniform Resource Identifier (URI), the resources are then being accessed via regular HTTP requests. This limits the available operations to only those that were enabled and with each transaction the entire resource will be transmitted. For instance if a person resource is requested then all the information stored about the person will be sent. (Wittern et al., 2018, p.66).

A REST query has the following format

```
GET http://127.0.0.1/api/person
```

The response from the server in JSON (Java Script Object Notation) format can be seen below. If we are only interested in the name of the person then the address and all other fields still have to be transmitted with REST.

```
{
  "id": 88,
  "name": "Mathias Wiesbauer",
  "age": 14,
  "address": {
      "street": "200 Honor Lane",
      "city": "Herndon",
      "zip": 20170
  }
}
```

Lets now compare the same query with GraphQL, we can specify the fields we would like to be returned in the request body.

In the example below we request only the 'name' field.

```
--REQUEST--
POST http://127.0.0.1/graphql

--BODY--
query {person {name}}
```

The following response in JSON format is being returned by GraphQL.

```
{
  "data": {
    "person": [
      {
        "name": "Mena Meseha",
      },
    ]
  }
}
```

We can see that the number of fields in the GraphQL response contains only the requested field 'name' and no other data. This demonstrates the overhead in the amount of data we incur with REST.

2.2 Too many requests

Another issue with REST is that each resource we need to access on the server requires a separate network request. Each request adds additional overhead and increases the response time.

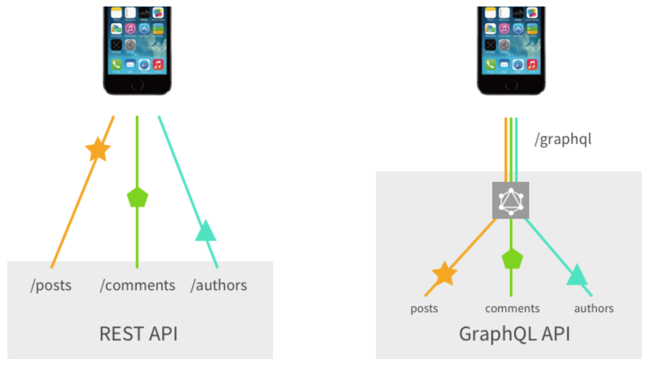

Figure 1: *API design for client server communication (Stubailo, 2017)*

GraphQL uses a schema that allows all data to be transmitted in a single request (Yegulalp, 2018). In the figure below we that three resources require three requests with REST and only one with GraphQL.

3 Benefits of GraphQL

Harrison (2018) lists the follwong benefits of GraphQL:

- A friendlier syntax
- Allows the splitting of monolithic applications into microservices
- More efficient
- Reduction of network overhead

Brito et al. (2019) provide a more detailed analysis of the performance benefits of using GraphQL. They demonstrate the performance increase in the following areas:

- API call reduction by 17% after migrating from REST to GraphQL
- Number of fields transmitted reduced by 94% after migrating from REST to GraphQL
- Data transmitted reduced by more than 98% from REST to GraphQL

4 Conclusion

We have clearly demonstrated the increase in efficiency and the potential to save money by reducing resource utilization and data transfer volumes, while increasing the responsiveness of web-services to improve user satisfaction. Many companies have already started the adoption to GraphQL "...among them GitHub, Yelp, and the New York Times" (Wittern et al., 2018).

Although GraphQL implementations are not yet available for all platforms and libraries (Stubailo, 2017) with the continued growth of internet services and the need for responsiveness and efficiency GraphQL provides solutions to many of the technical issues that makes it the better version of REST, and is expected to play an even more important role in the future.

References

Brito, G., Mombach, T., & Valente, M. (2019). Migrating to GraphQL: A Practical Assessment.

Fielding, R. T. (2000). *Architectural styles and the design of network -based software architectures*. Ph.D., University of California, Irvine, United States – California.

Harrison, G. (2018). Web Services Move Forward With GraphQL. *Database Trends and Applications; Chatham*, *32*(6), 50.

Helgason, A. F. (2017). *Performance analysis of Web Services: Comparison between RESTful & GraphQL web services*. PhD thesis.

Stubailo, S. (2017). GraphQL vs. REST. https://blog.apollographql.com/graphql-vs-rest-5d425123e34b.

Wittern, E., Cha, A., & Laredo, J. A. (2018). Generating GraphQL-Wrappers for REST(-like) APIs. In Mikkonen, T., Klamma, R., & Hernndez, J. (Eds.), *Web Engineering*, Lecture Notes in Computer Science, (pp. 65–83). Springer International Publishing.

Yegulalp, S. (2018). What is GraphQL? Better APIs by design. https://www.infoworld.com/article/3269074/what-is-graphql-better-apis-by-design.html.