REST vs GraphQL: A Controlled Experiment

Gleison Brito, Marco Tulio Valente
ASERG Group, Department of Computer Science (DCC),
Federal University of Minas Gerais, Brazil

Motivation

- Empirical study to understand real gains achieved by graphQL
  - Effort required to implement queries in this language (quantitative)
  - Developer perception (qualitative)

Research Questions

- RQ1: how much time do developers spend when implementing queries in REST vs GraphQL?
  - RQ1.1: How does this time vary between the types of queries
  - RQ1.2: How does this time vary among undergraduate and graduate students?
  - RQ1.3: How does this time vary depending on experience with REST vs GraphQL?
- RQ2: What are the participants' perceptions about REST vs graphQL?

Methodology

- Researchers identified 22 subjects - 10 undergrad and 12 graduate level computer science students
- The subjects were asked to complete 8 tasks
  - interacting with GitHub's Repositories and Users APIs
  - each task completed once in REST and again in graphQL
  - time to completion is observed
- Subjects were given questionnaires to pre-screen for pre-requisite programming skills and to counterbalance them into two groups
- The lead author gave the subjects a talk on the basics of REST and GraphQL. This material along with all the documentation was available to the subjects during the experiment

Results

- RQ1: how much time do developers spend when implementing queries in REST vs GraphQL?
  - RQ1.1: How does this time vary between the types of queries
    - GraphQL outperforms REST mainly in queries that require several parameters. In such queries, auto complete-as provided by GraphQL's IDEs is a powerful feature to help developers. For example, a novice developer spent 63% of this time in REST and 37% in GraphQL
  - RQ1.2: How does this time vary among undergraduate and graduate students?
    - Both undergraduate and graduate stu- dents have taken benefit of GraphQL and implemented the tasks in less time.
  - RQ1.3: How does this time vary depending on experience with REST vs GraphQL?
    - GraphQL outperforms REST even among participants with previous experience in REST and no previous contact with GraphQL.
- RQ2: What are the participants' perceptions about REST vs graphQL?
  - According to the subjects, the main benefit of GraphQL are the tool support provided by GraphiQL, e.g., auto complete feature. Another mentioned benefit is a better syntax to read the code and less effort to specify parameters. By contrast, two participants commented about the poor quality of GitHub's GraphQL API documentation.

Positives

- Paper is written in plain language
- Takes a pragmatic approach to comparing REST vs graphQL
- Extends on previous work and leaves ideas on future work

Negatives

- Study needs to be done out in the industry with professionals
- GraphQL IDE contributed to the results through autocomplete. It is questionable if this advantage will remain if such a developer tool was developed for REST