# Article information:

REST vs GraphQL: A Controlled Experiment | IEEE Conference Publication | IEEE Xplore
<https://ieeexplore.ieee.org/abstract/document/9101226>

# Article summary:

1. A controlled experiment was conducted with 22 students to compare the effort required to implement queries in REST and GraphQL for accessing a web service.

2. The results showed that GraphQL requires less effort to implement remote service queries when compared to REST, especially for more complex endpoints with several parameters.

3. Participants also mentioned two key benefits of using GraphQL: tool support for building and testing queries and a syntax and semantics similar to standard programming languages.

# Article rating:

Appears moderately imbalanced: The article provides some useful information, but is missing several important points or pieces of evidence that would be required to present the discussed topics in a balanced and reliable way. You are encouraged to seek a more balanced perspective on the presented issues by exploring the provided research topics and looking at different information sources.

# Article analysis:

The article "REST vs GraphQL: A Controlled Experiment" presents a study comparing the implementation of queries using REST and GraphQL. The study involved 22 students, both undergraduate and graduate, with varying levels of experience in REST and GraphQL. The authors aimed to answer two research questions: how much time do developers spend implementing queries in REST and GraphQL, and what are the participants' perceptions about REST and GraphQL?

The article provides a clear overview of REST and GraphQL, highlighting their differences in terms of architectural models, endpoints, and query languages. The authors also explain the motivation behind their study, which is to provide empirical evidence on the real gains achieved by GraphQL compared to REST.

The study found that GraphQL requires less effort to implement remote service queries when compared to REST. However, the gains were mostly restricted to REST queries that have several parameters. Interestingly, even experienced participants (graduate students) and those with previous experience in REST but no previous experience in GraphQL found that GraphQL outperformed REST.

While the study provides valuable insights into the implementation of queries using REST and GraphQL, there are some potential biases that need to be considered. Firstly, the sample size is relatively small (22 students), which may limit the generalizability of the findings. Additionally, all participants had at least one year of programming experience, which may not reflect the wider population of developers.

Furthermore, while the authors acknowledge previous studies comparing REST and GraphQL under a quantitative perspective, they do not explore these studies in detail or consider any potential limitations or biases within them. This could be seen as a one-sided reporting approach.

Overall, while the article presents interesting findings on the implementation of queries using REST and GraphQL, it is important to consider potential biases and limitations within the study design and acknowledge other relevant research in this area.

# Topics for further research:

* Quantitative studies comparing REST and GraphQL
* Limitations of REST and GraphQL in different scenarios
* Best practices for implementing REST and GraphQL
* Comparison of REST and GraphQL in terms of performance and scalability
* Real-world use cases of REST and GraphQL
* Adoption rates of REST and GraphQL in industry

# Report location:

<https://www.fullpicture.app/item/f7eaf681f5b466a26299425c6c5c188a>