

# Architecture Selection for API: Rest vs GraphQL

**Team members:** Arsalan Imran (002344722), Guan Wang (002341590), Dan Luo (002352806)

## Introduction:

The choice of API architecture can significantly impact the performance, scalability, and overall user experience of an application. The decision between Rest and GraphQL often poses a significant challenge while REST offers a well-established approach based on resource-oriented principles, GraphQL introduces a more dynamic and efficient method for fetching data. This project aims to provide empirical insights into the performance of REST and GraphQL APIs by evaluating key metrics such as error rates, throughput, response times, and load. We offer developers a comprehensive understanding of how each architecture performs in real-world scenarios.

## Objective

- **Identify Use Cases:** Identify specific use cases where REST or GraphQL architecture excels or faces challenges, providing developers with practical guidance on architectural selection. Evaluate the performance of REST and GraphQL APIs under different scenarios.
- **Metric Analysis:** analyze key performance metrics such as error rates, throughput, response times, and system scalability for both REST and GraphQL architectures. The analysis of REST and GraphQL APIs will be done by its ability to handle increasing numbers of concurrent users and data requests.

## Methodology:

- **Dataset Creation:** We compile a dataset with comprehensive attribute including Api type, category, labels, num of users, average response time, category, errors, median, min response time, max response time, 20th percentile, 90%, 95%, 99% Line (Percentiles), Throughput, KB/sec (Kilobytes per second).
- **Analysis:** We will analyze the performance of Apis using the dataset and statistical analysis, including:
  - **Multivariate Regression with User Count:** Understand how user count affects API performance (e.g., error rate, average response time, throughput).
  - **Comparative EDA Analysis of Two APIs' Categories:** Understand the distribution and relative importance of categories for each API in the dataset.
  - **Significance Testing for Error Rate** (e.g., t-tests, Wilcoxon signed-rank tests): Identify if one API has significantly lower error rates.
  - **Significance Testing for Throughput** (e.g., t-tests, Wilcoxon signed-rank tests): Ascertain if there are significant differences in throughput data between the two APIs
  - **Significance Testing for Average Response Time** (e.g., t-tests, Wilcoxon signed-rank tests): Assess if one API consistently responds faster.
- **Validation:** To validate our findings, we test publicly available REST and GraphQL APIs, corroborating dataset-derived insights with real-world observations.

## 6. Resources:

IDE: Jupyter Notebook, Language: Python, Libraries: Numpy, Pandas, PyTorch, transformers, datasets, matplotlib etc., Tools: JMeter and Chrome's developer tool

## 7. Learning Experience

Understand Rest and GraphQL architecture, learn Api's testing, factors influencing Api performance and dataset creation. Process data and perform analysis to find patterns. learn tools i.e. JMeter and chrome's developer tool.

## 8. Deliverables

Api performance dataset, Analysis Report, Implementation after analysis

## 9. References

<https://stfalconcom.medium.com/understanding-api-performance-and-enhancing-its-efficiency-374c1a5ee865>