THE DIFFERENCES BETWEEN GRAPHQL AND REST APIS: UNDERSTANDING THEIR DEVELOPMENT LIFECYCLES

Tkachov V.M., Afanasieva A.M.
Kharkiv national university of radioelectronics, Kharkiv, Ukraine

Most people think that GraphQL is a database technology, but this is not accurate. GraphQL is a query language API, and most applications today have their data hosted on a remote server in a database.

The API only has to provide an interface for the location information that meets the application's requirements.

GraphQL is an alternative option for building APIs in REST. As a query language, GraphQL defines the specifications for how a client application can request the necessary data from a remote server. In REST, one API endpoint is called to request the data needed by the client application, and the response is returned by the server based on the requested query. When someone uses the REST API, they would first have an endpoint, this endpoint would get the information. You may notice two problems:

1. Multiple round trips with REST
2. Overseeking and underseeking issues with REST

Frequently with REST, one will end up with unnecessary information at a specific stage. Due to outsourcing, you can see that the call may not be adequate to retrieve what the client application was looking for.

During the consumption of the REST APIs, the front-end development groups have to wait a certain amount of time for the back-end group to finish writing these APIs for the client application to get and post the data. The GraphQL lifecycle provides more efficient approach where frontend and backend developers can work in parallel without obstructing the overall development process.

The REST API and GraphQL API development lifecycle approaches are useful on an as-needed basis. GraphQL is gaining popularity exponentially, mainly due to its "no more, no less" ability. It provides more efficient collaboration mechanisms. In short, GraphQL is a tool to achieve specific query-oriented goals; however, it is not a solution to all API-related challenges and certainly not a replacement for REST.

References

1. M. Hunko. Application Architecture For Obtaining Data From Scientometric Databases / M. Hunko, V. Tkachov, O. Liashenko, J. Rabčan // 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek). – Kharkiv, 2022.

2. Tkachov V. Method to Determine Fault-Tolerant Performance Probability of High Survivability Computer Network based on Mobile Platform / Vitalii Tkachov, Mykhailo Hunko, Olga Morozova, Artem Tetskyi, Andrii Nicheporuk // IEEE International ScientificPractical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 05-07 oct. 2021, Kharkiv