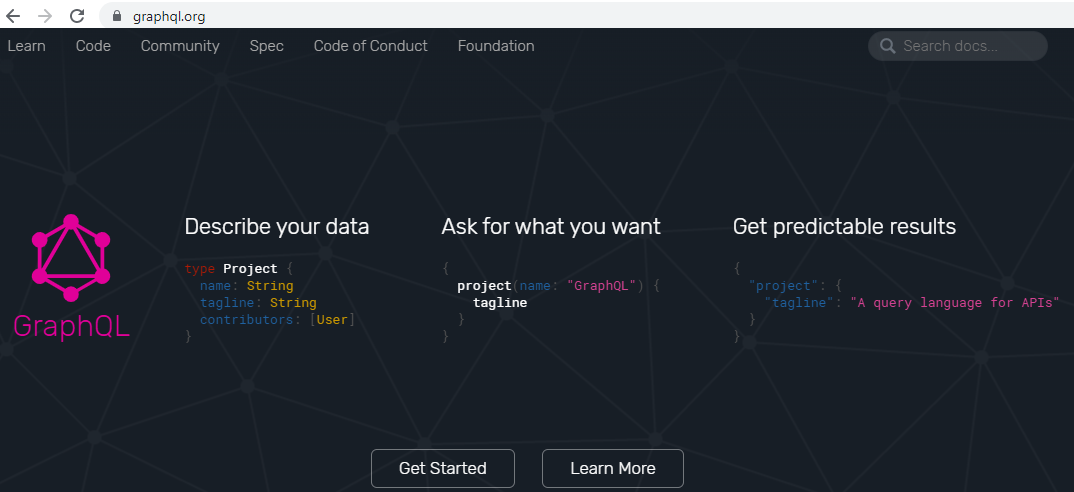
Лекция 14

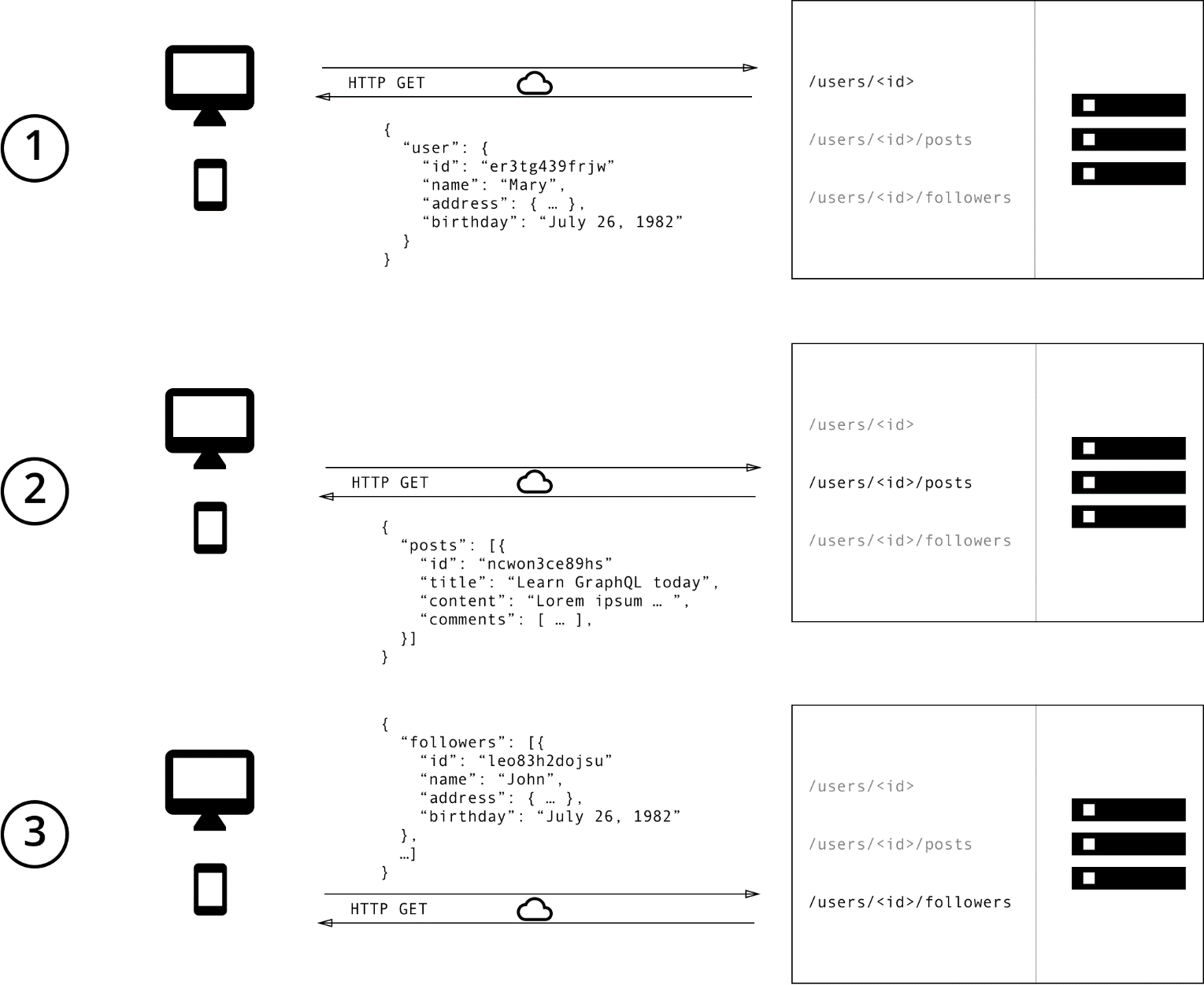
ПСКП, ПОИТ-3

**Node.js: GraphQL**

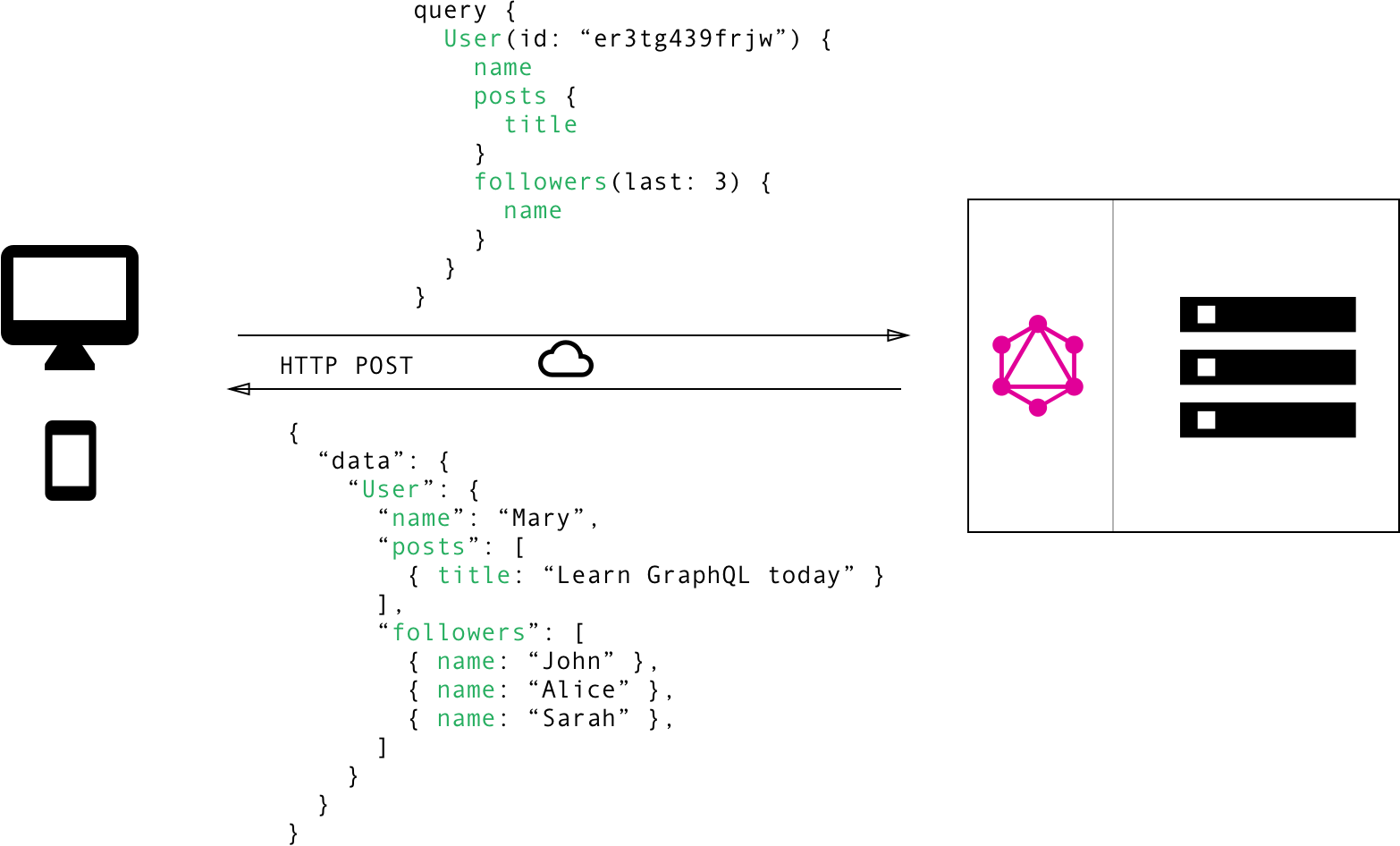
1. **GraphQL:** формальныйязык запросов;особенности: позволяет делать агрегированные запросы (можно в одном запросе запросить все необходимые данные), использует систему типов.

****

1. **REST vs GraphQL: REST**



1. **REST vs GraphQL: GraphQL**

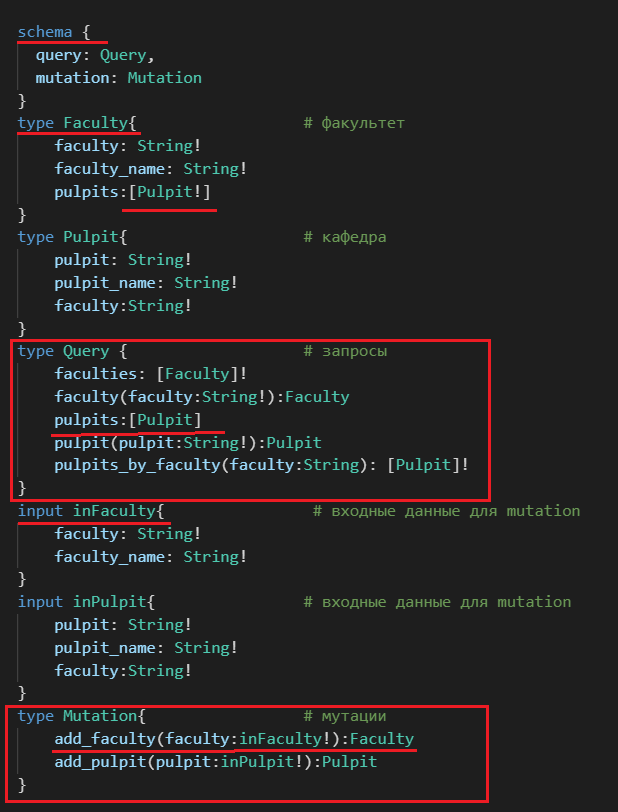


1. **REST vs GraphQL (сходства):** единый интерфейс, ответ в формате JSON, разделение запросов на чтение и на запись.
2. **GraphQL (преимущества перед REST):** клиентоориентированность (ожидаемый результат, только нужные данные, меньший объем ответа, быстрее), один endpoint, сильная типизация, комбинация запросов, большая независимость фронта и бэка.

1. **GraphQL:** diskstation, /Для\_студентов\_ФИТ\_БГТУ/ЛИТЕРАТУРА/GRAPHQL
2. **GraphQL:** несколько реализаций: **Node.js**, Java, .NET C#, Python,…
3. **GraphQL в Node.js**: graphql, apollo-sever, express-graphql, graphql-yoga.
4. **GraphQL:** npm install **graphql**, <https://graphql.org/graphql-js/>
5. **GraphQL:** основные компоненты: **schema**, **query/mutation/subscription**, **resolvers**.
6. **GraphQL:** архитектура



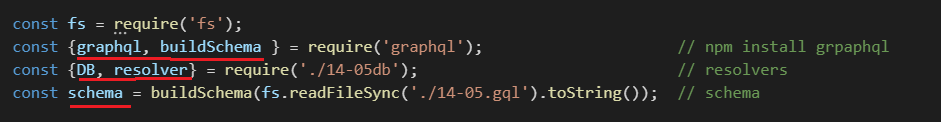
1. **GraphQL:** схема, SDL (Schema Definition Language), .gql/.graphql

****

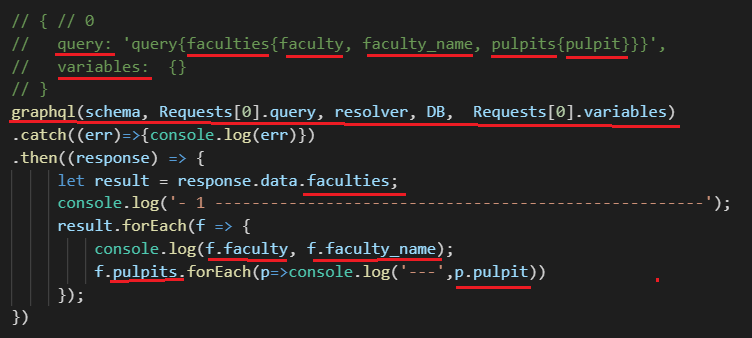
1. **GraphQL: *типы данных*** (объектные и скалярные (Int, Float, Booleam, String, ID)) ***и модификаторы типов*** (! (non-null), [] (list))
2. **GraphQL:** query, execute

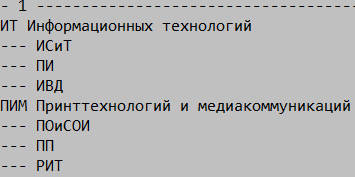
|  |  |
| --- | --- |
| query {      faculties {          faculty,          faculty\_name,          pulpits {              pulpit          }      }  } |  |

1. **GraphQL:** schema, resolvers, context

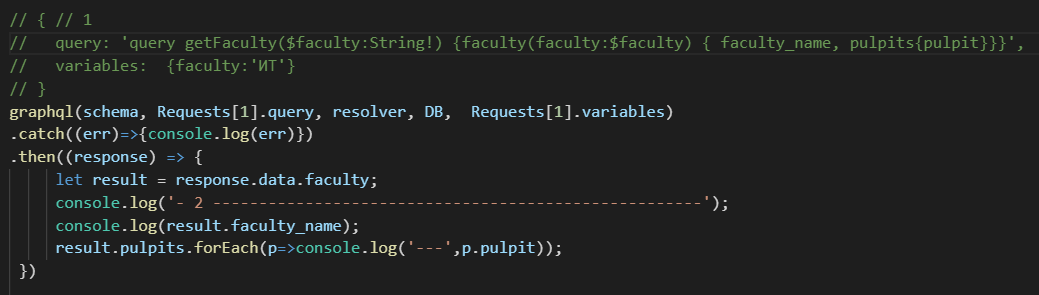
****

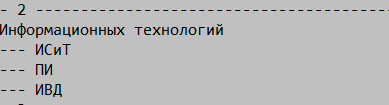
1. **GraphQL:** graphql, schema, resolvers, context(DB)

****

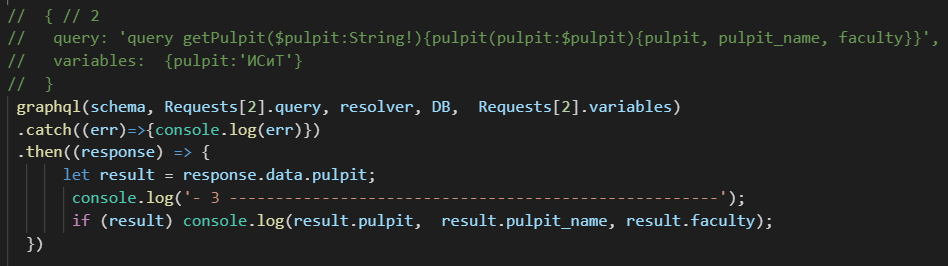
****

1. **GraphQL:** graphql, schema, resolvers, context(DB)

****

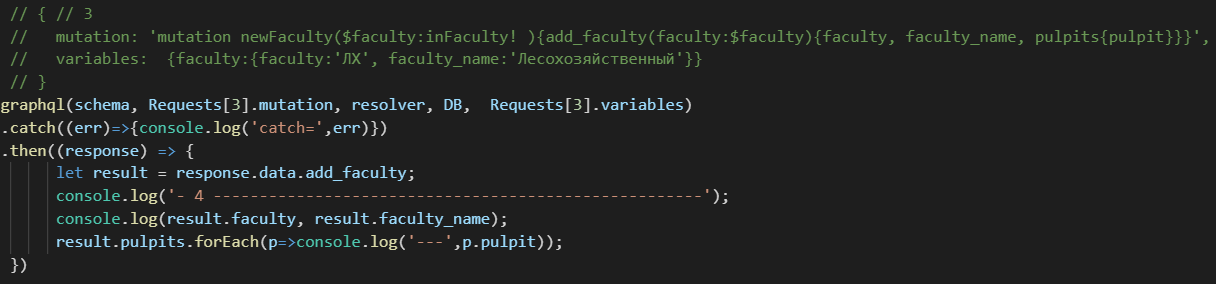
****

1. **GraphQL:** graphql, schema, resolvers, context(DB)

****

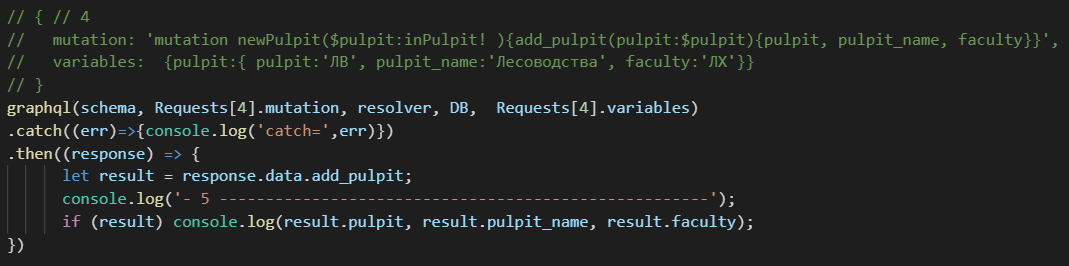
****

1. **GraphQL:** graphql, schema, resolvers, context(DB)

****

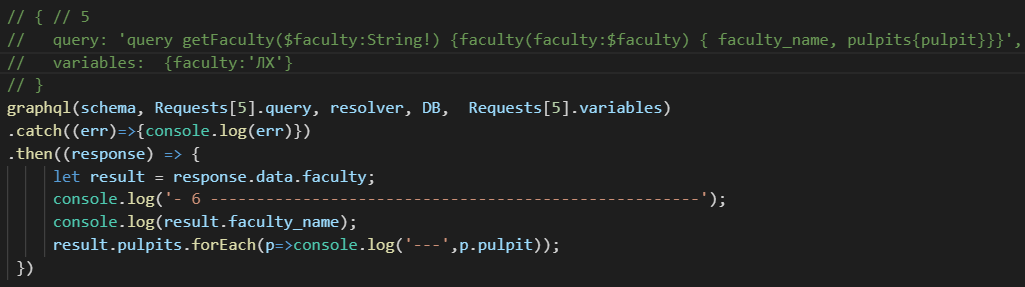
****

1. **GraphQL:** graphql, schema, resolvers, context(DB)

****

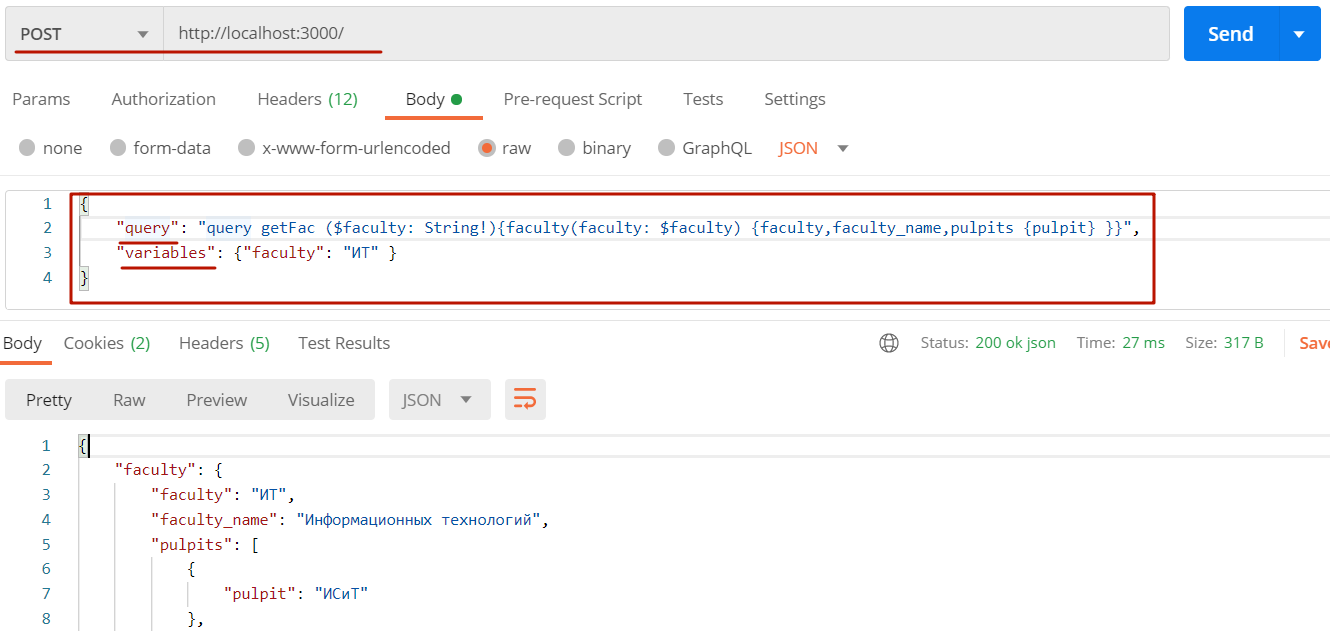
****

1. **GraphQL:** graphql, schema, resolvers, context(DB)

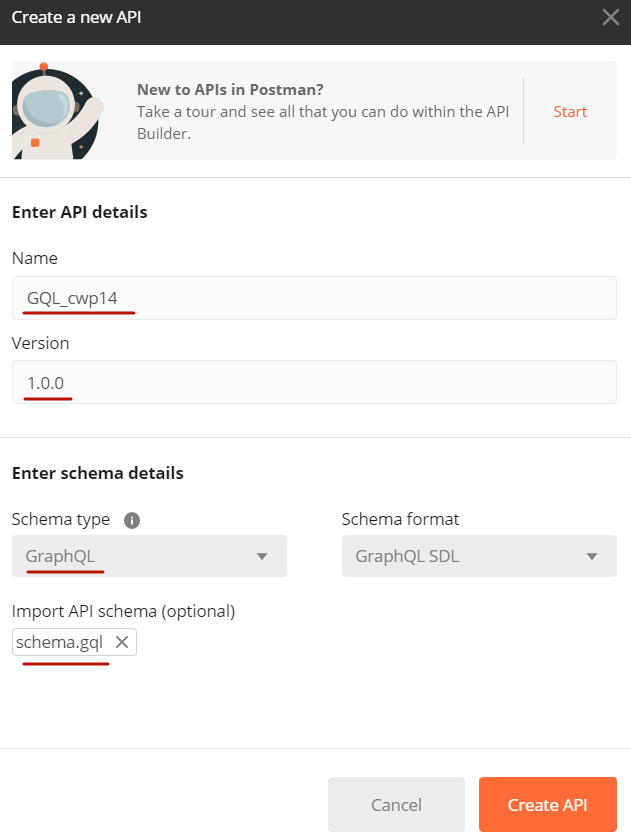
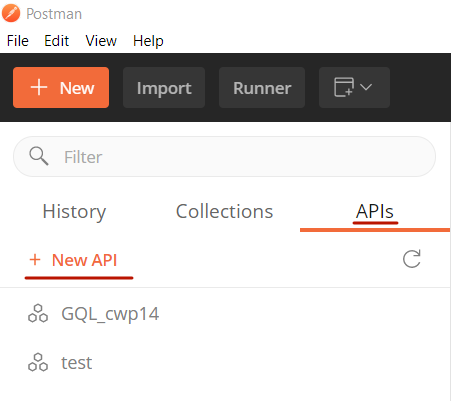
****

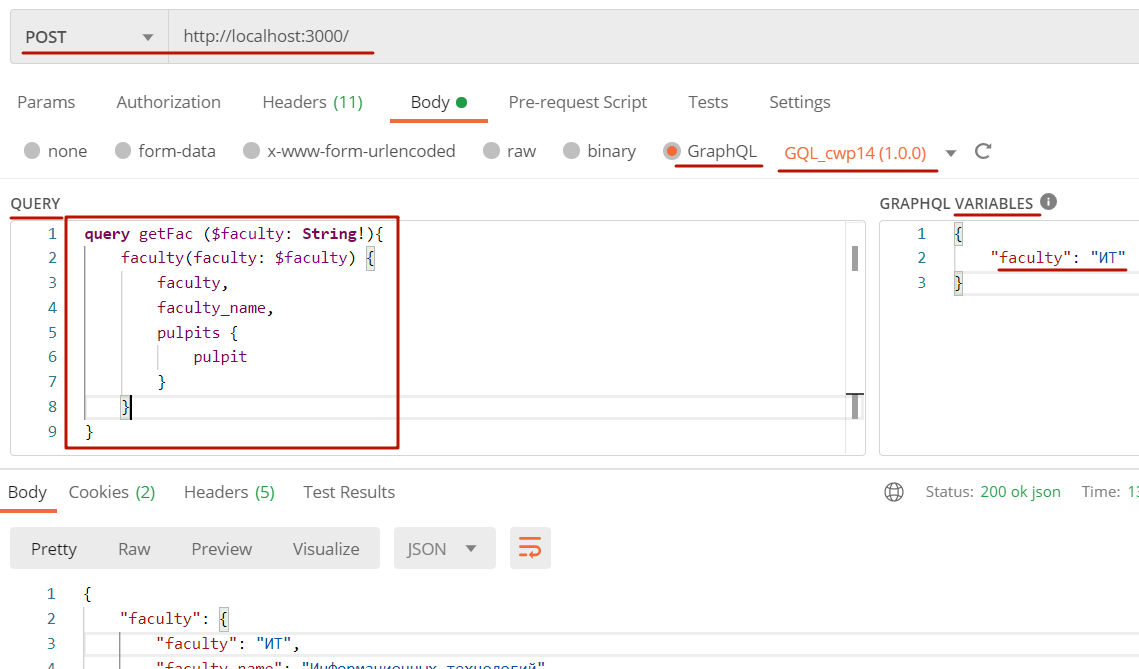
****

1. **GraphQL:** Postman

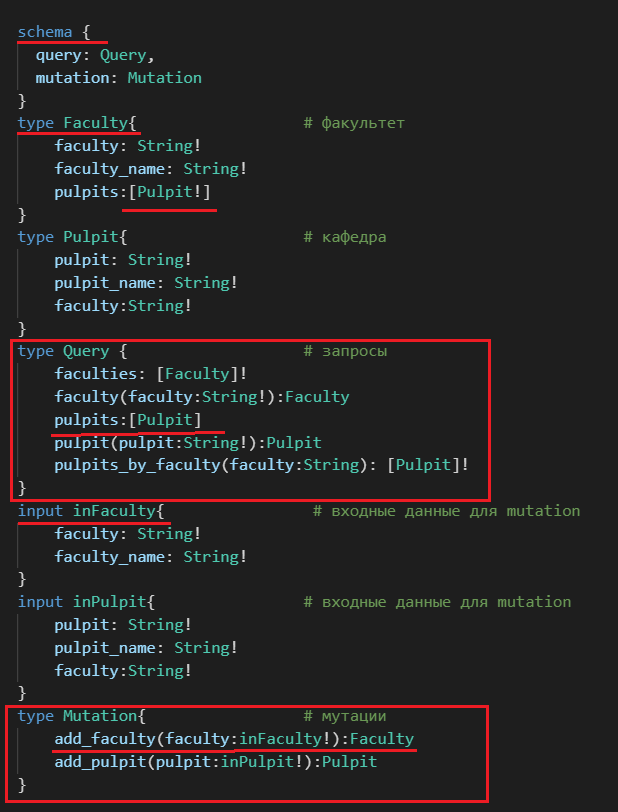


1. **GraphQL:** Postman (API => +New API)

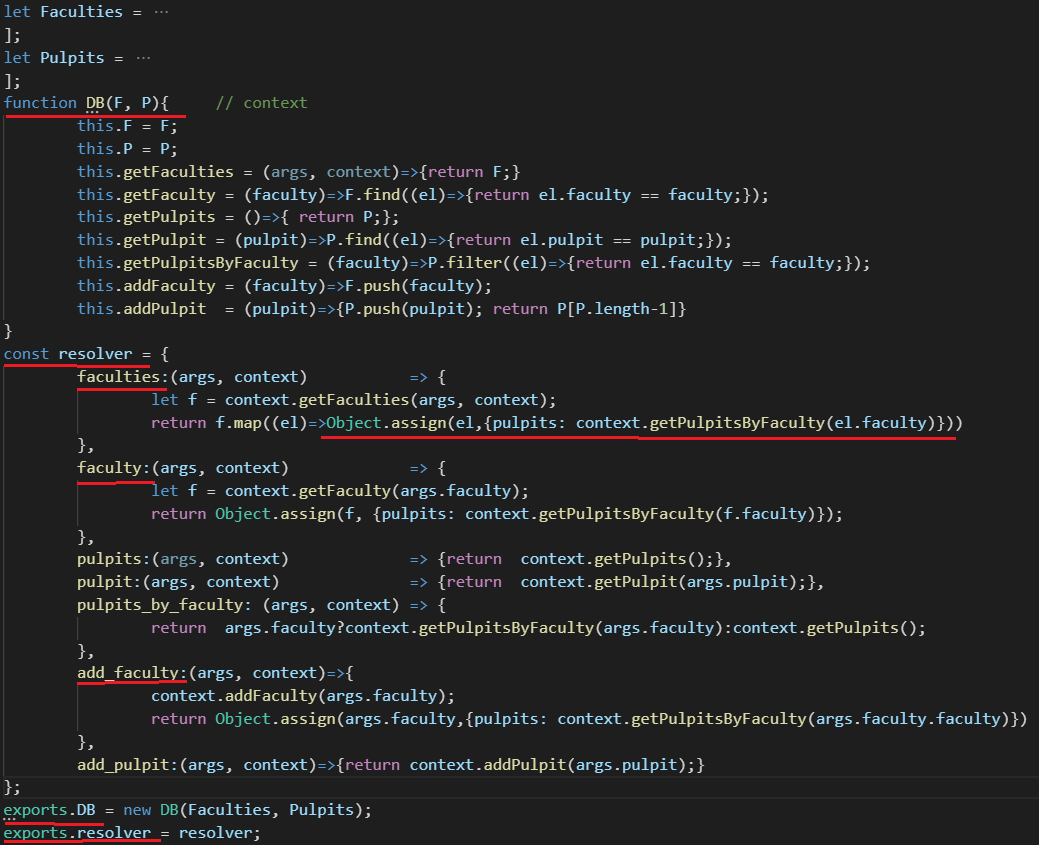




1. **GraphQL:** schema, query, mutation

****

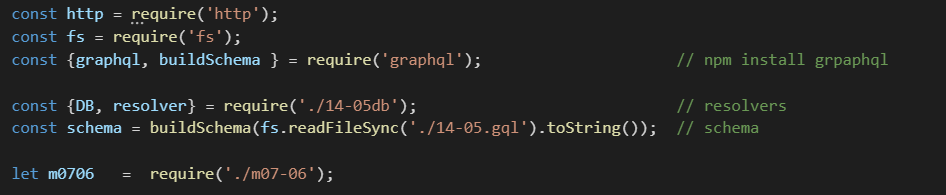
1. **GraphQL:** query, mutation, resolvers

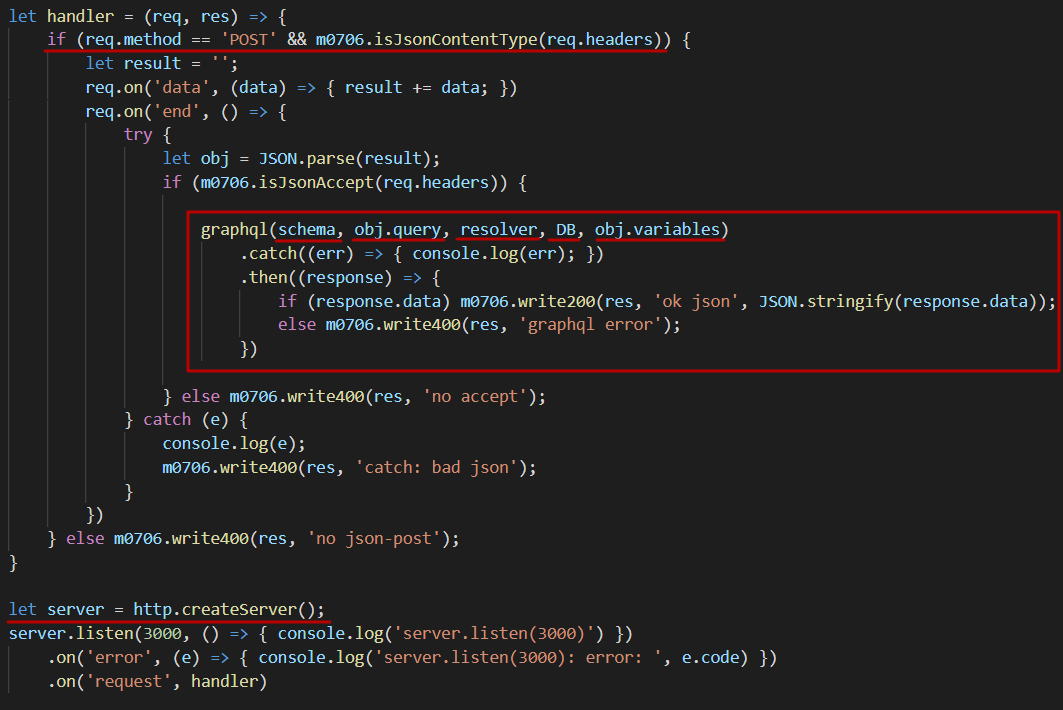


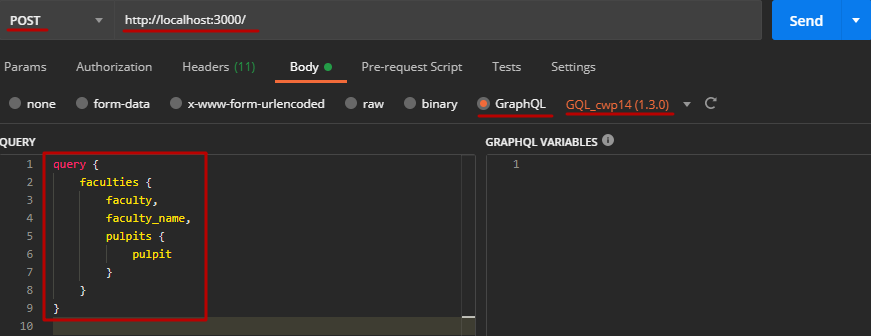
**m07-06.js**

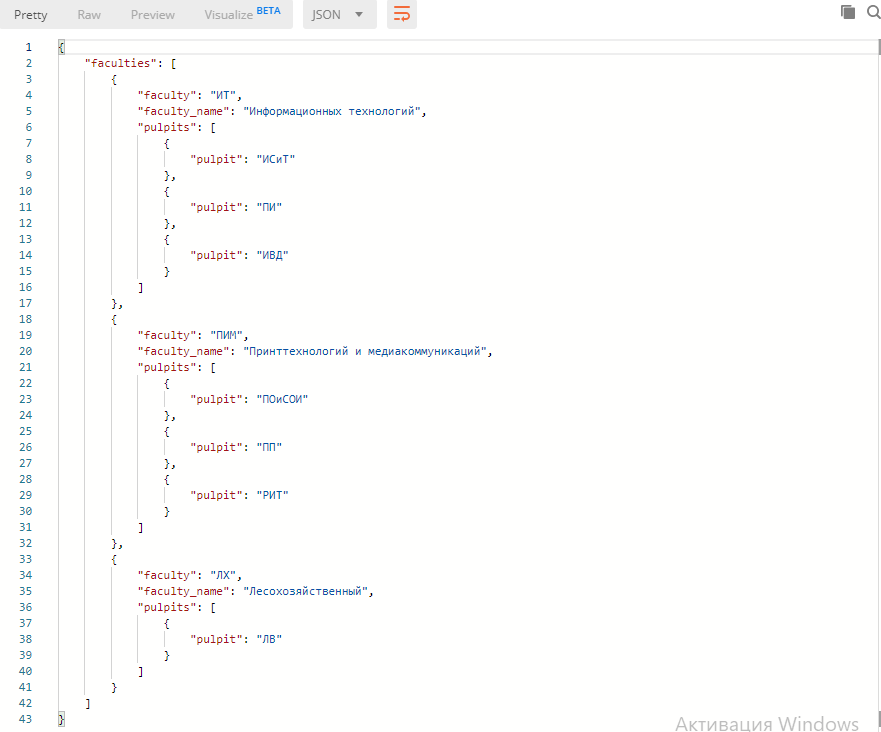


1. **GraphQL:** http, graphql, postman

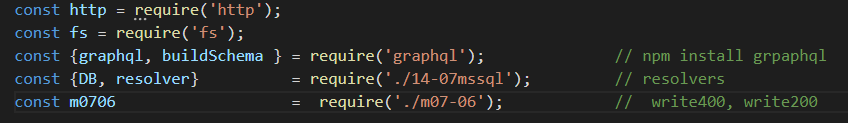
****

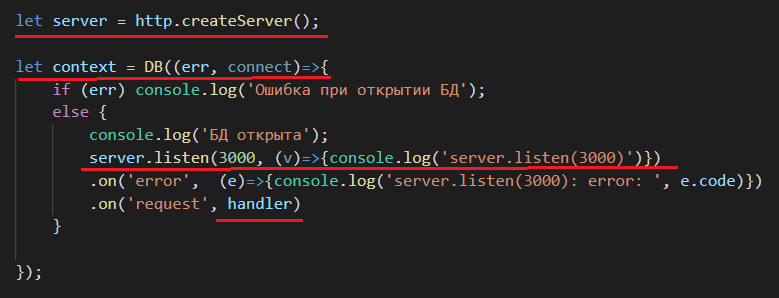


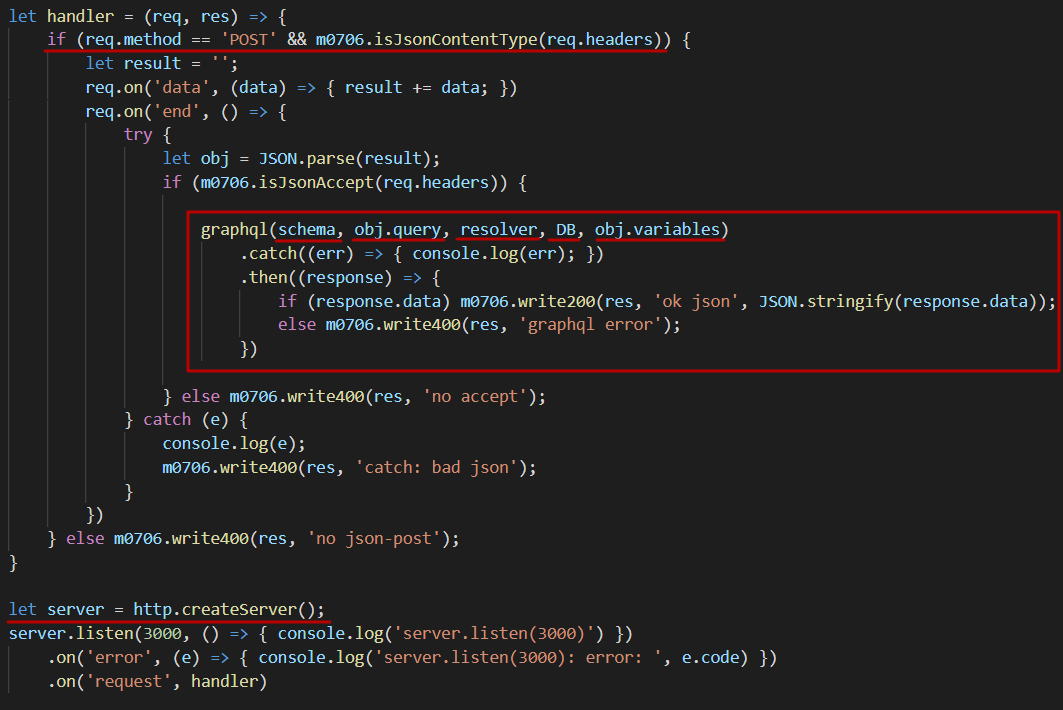


****

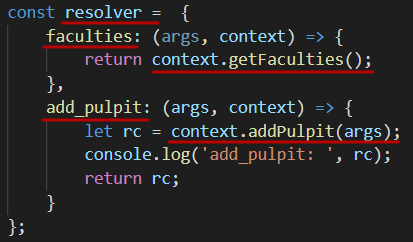
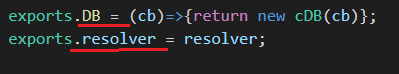
1. **GraphQL:** http, graphql, mssql

****

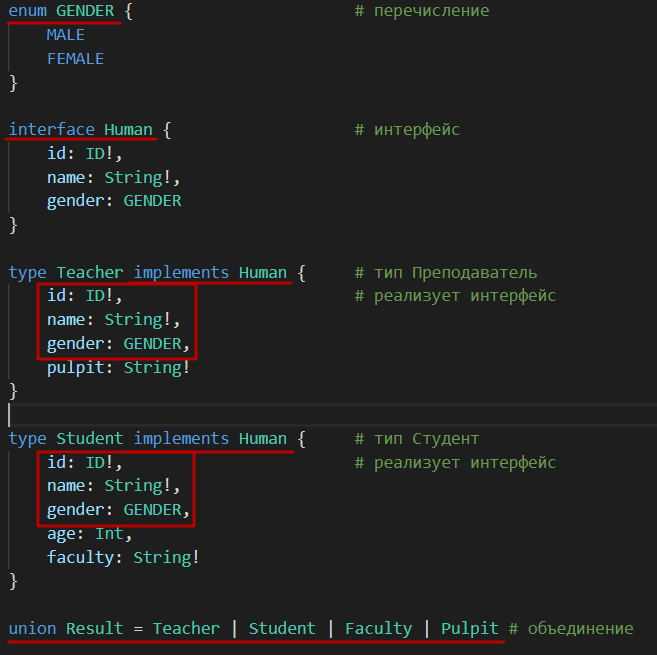
****

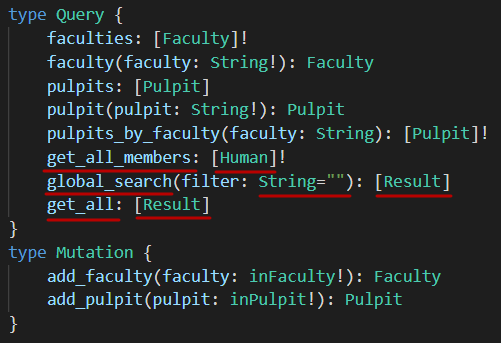


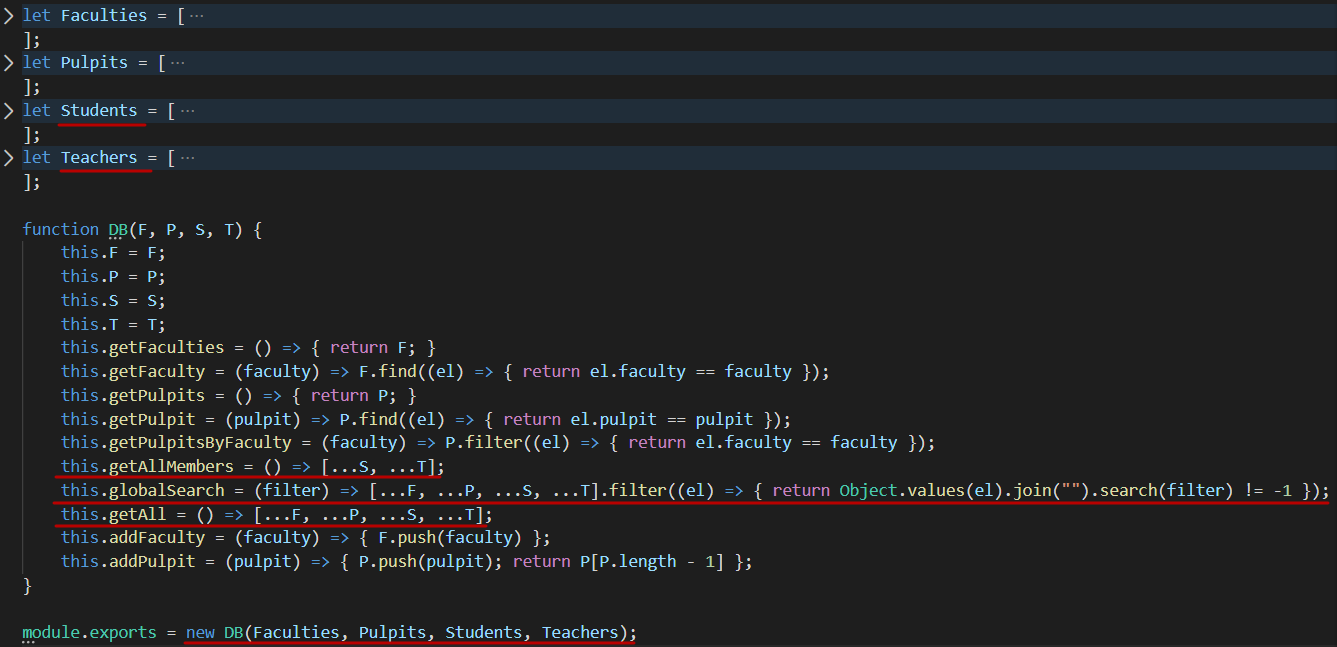


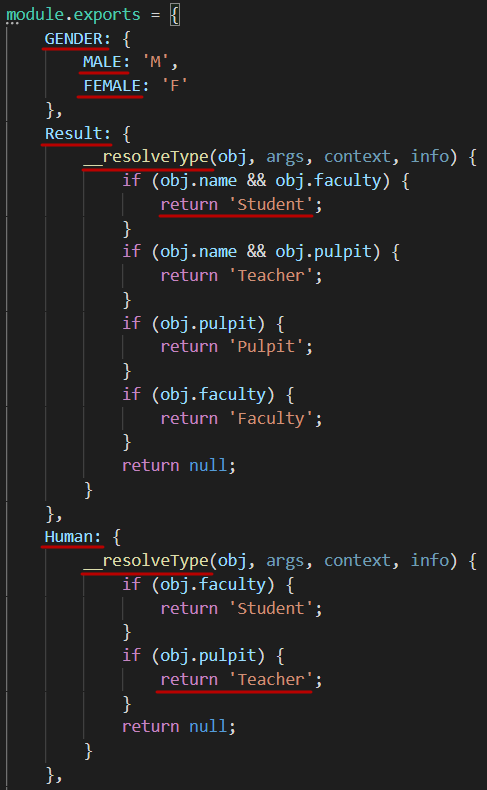
****

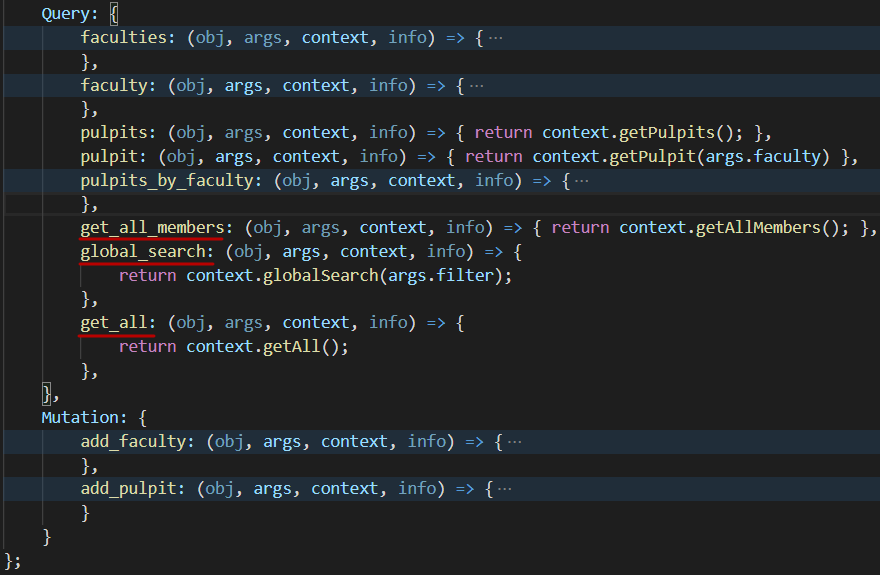
1. **GraphQL:** npm install **graphql-tools**
2. **GraphQL:** enum, interface, union, inline fragment, fragment

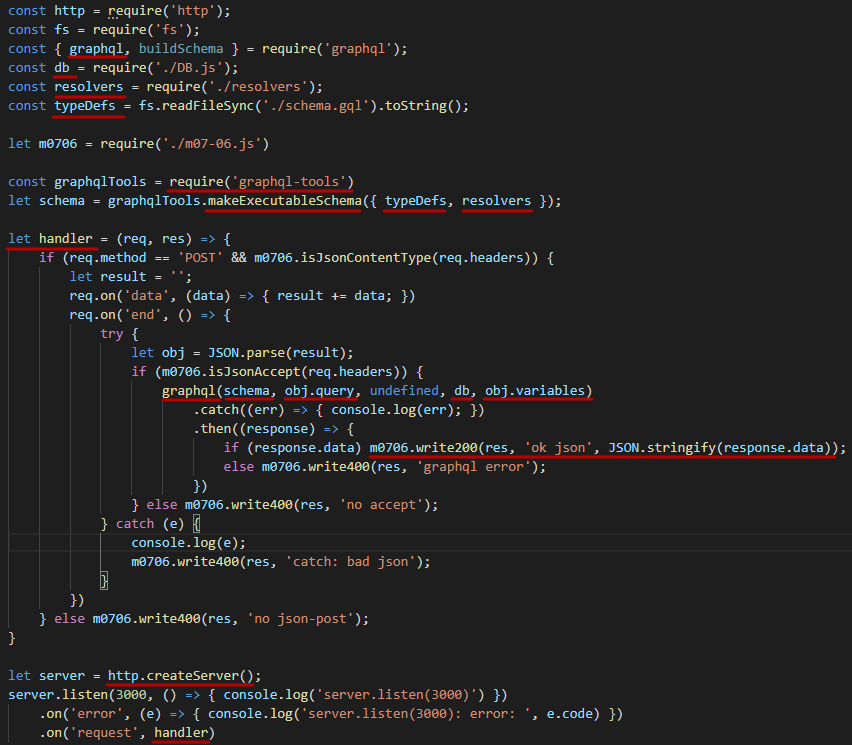


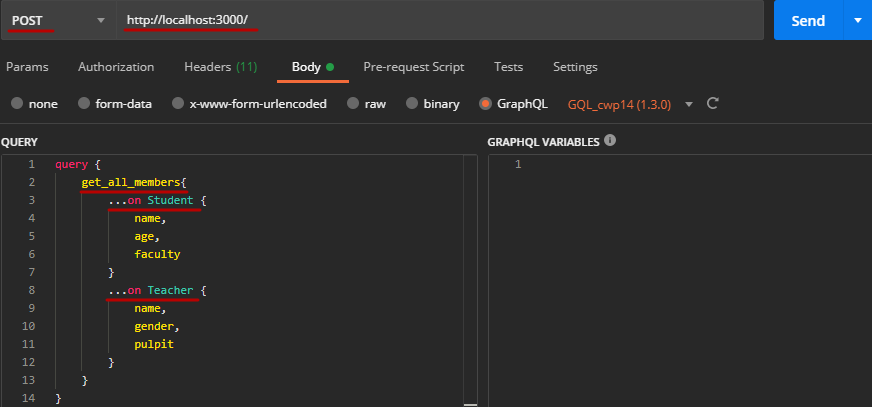




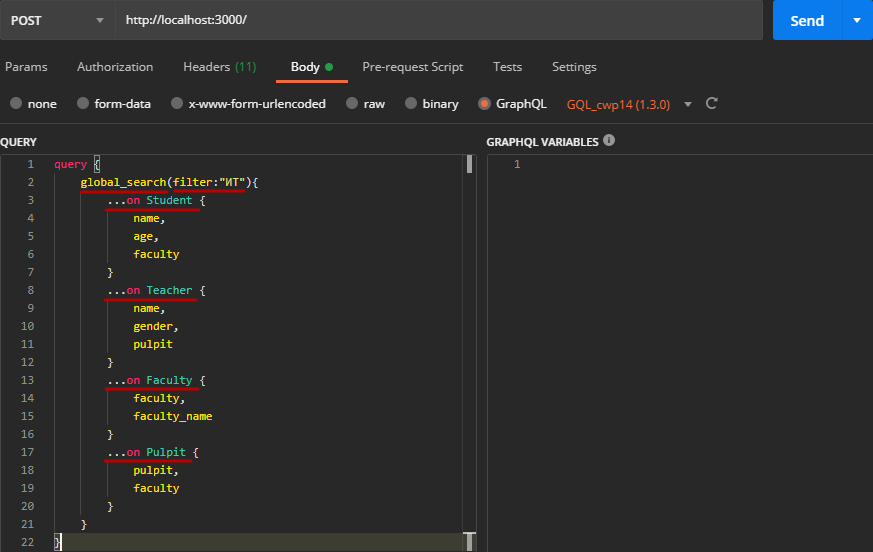


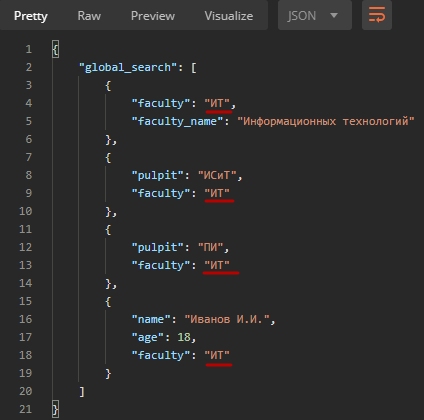


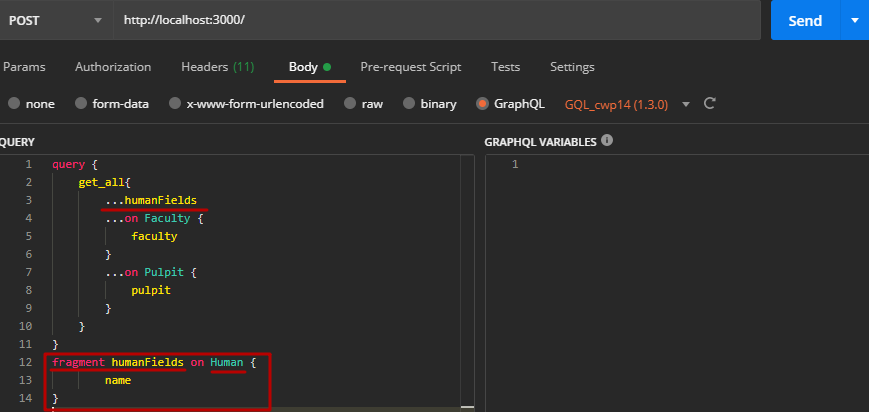


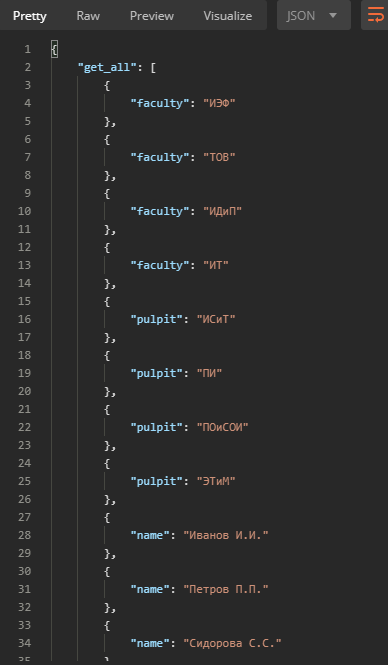


|  |  |
| --- | --- |
|  |  |
|  |

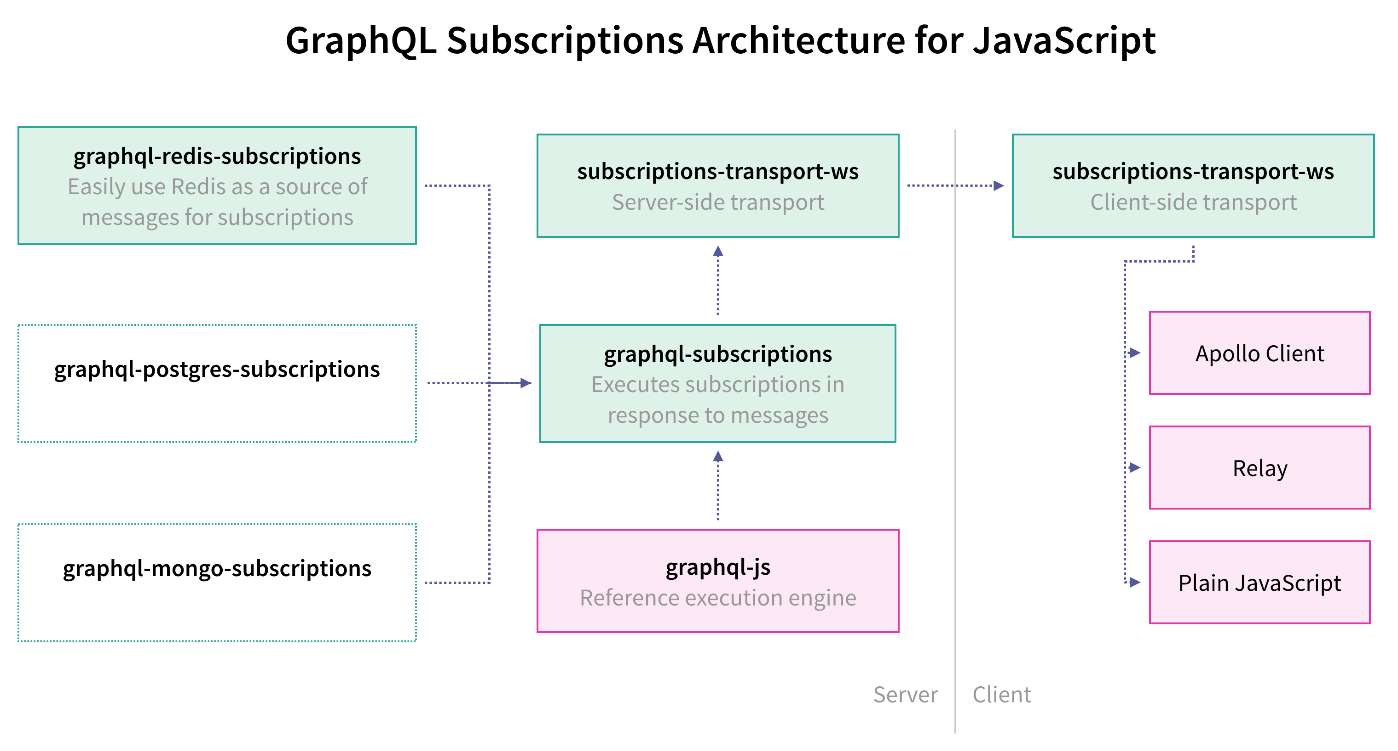




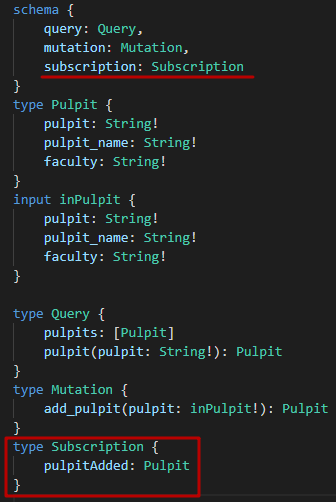


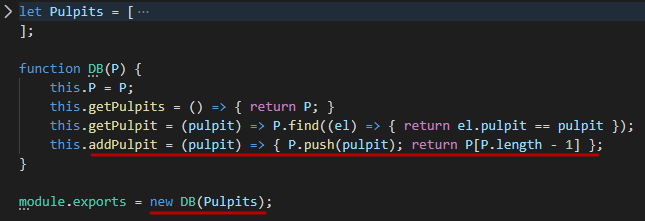


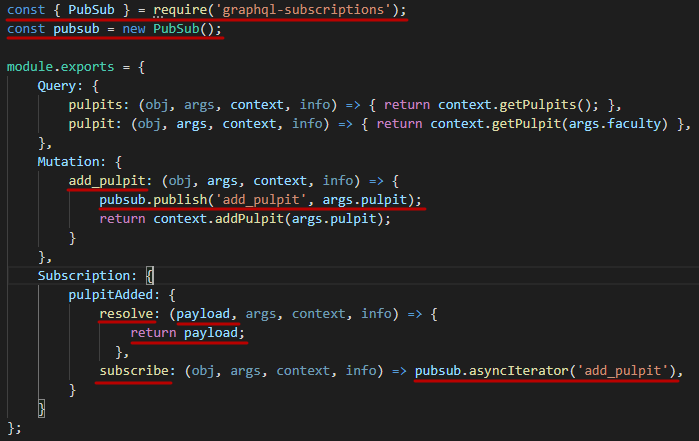
1. **GraphQL:** npm install **graphql-subscriptions**
2. **GraphQL:** npm install **subscriptions-transport-ws**

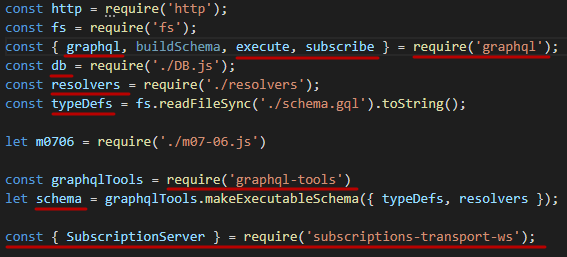
****

1. **GraphQL:** subscription



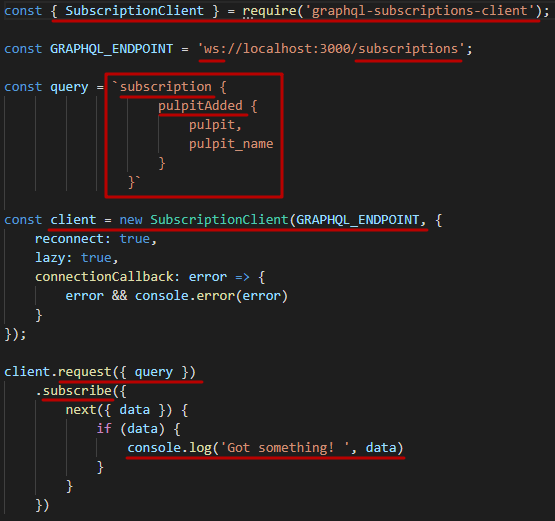


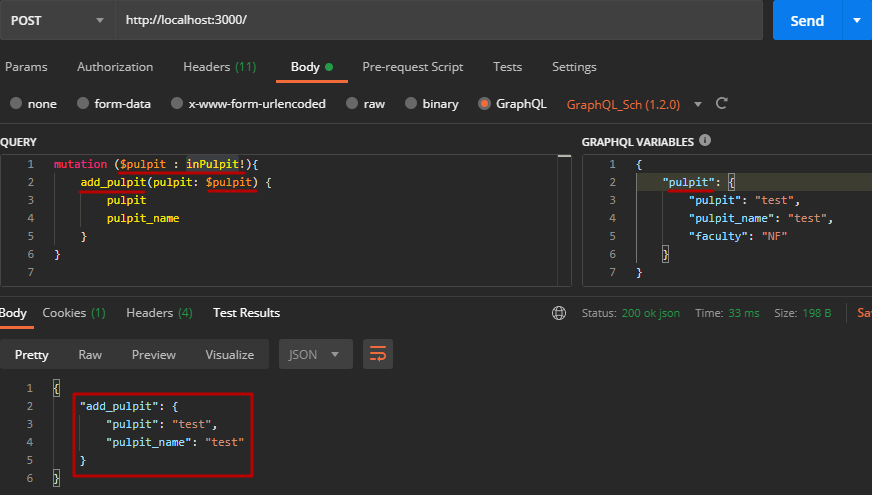






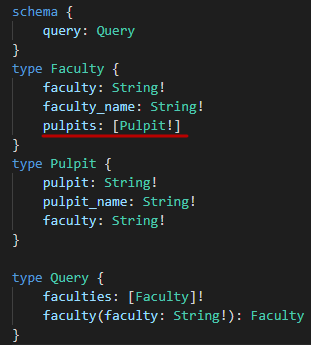
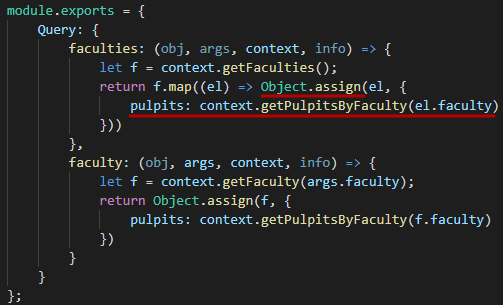
1. **GraphQL:** client**,** npm install **graphql-subscriptions-client**

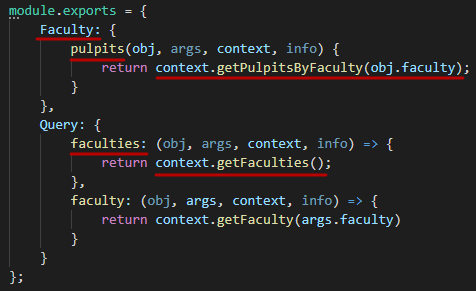
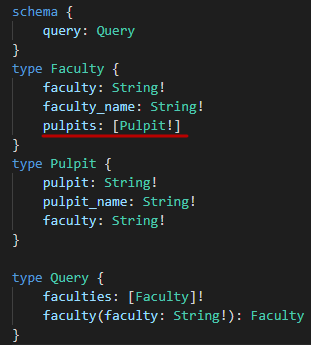




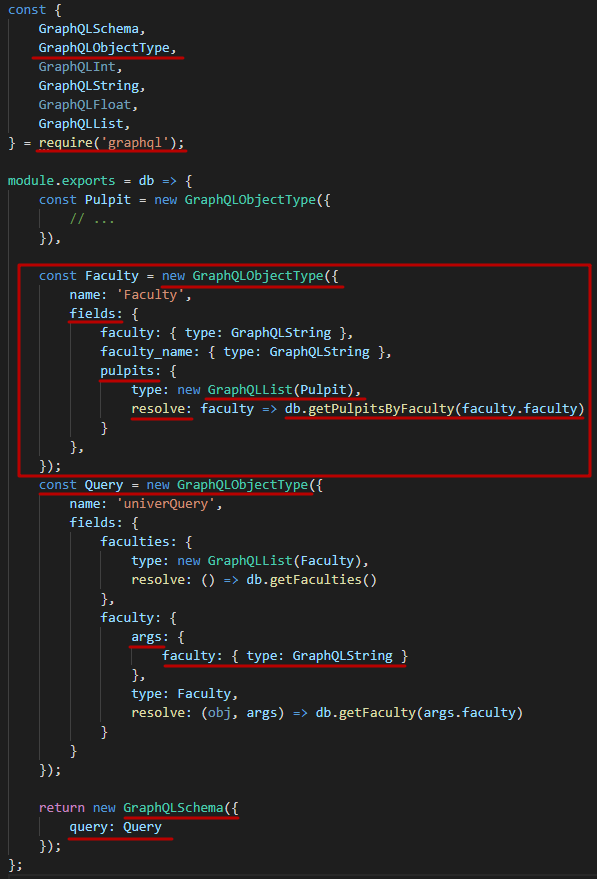


1. **GraphQL:** рекомендуется использовать какую-либо реализацию PubSubEngine из пакета graphql-subscriptions (например, graphql-google-pubsub, graphql-redis-subscriptions, graphql-postgres-subscriptions, …)
2. **GraphQL:** field resolvers, 1 способ



1. **GraphQL:** field resolvers, 2 способ



1. **GraphQL:** GraphQL IDE**: Postman, GraphiQL, GraphQL Playground, …**
2. **GraphQL:** GraphQL client**:** запросы с помощью **fetch и др., graphql-request, Relay, Apollo Client, …**