Sebastian Bentley (cph-sb287)

Period 3 GraphQL

GraphQL

Note: This description is too big for a single exam-question. It will be divided up into separate questions for the exam

Explain shortly about GraphQL, its purpose and some of its use cases
GraphQL is a query language for our API. It uses a type system to define our data. It allows us to only have ONE endpoint, and fetch only the data we need. This means that we wont over -and underfetch data.
Example: fullstack-startcode/src/graphql

Explain some of the Server Architectures that can be implemented with a GraphQL backend
Example: https://www.howtographql.com/basics/3-big-picture/

What is meant by the terms over- and under-fetching in GraphQL, compared to REST
We only fetch the data we need, with a single fetch. Instead of fetching too much, or too little from a REST endpoint.

Explain shortly about GraphQL's type system and some of the benefits we get from this
Once the schema is defined, frontend and backend teams can work independently from one another

Explain shortly about GraphQL Schema Definition Language, and provide examples of schemas you have defined.
A schema has queries and mutations for CRUD operations.
Example: fullstack-startcode/src/graphql/schema.ts

Provide examples demonstrating data fetching with GraphQL. You should provide examples both running in a Sandbox/playground and examples executed in an Apollo Client
Example Graphiql: *npm run dev* fullstack-startcode/ localhost/5555/graphql
Example Apollo Client: *npm start* startcode-frontend

Provide a number of examples demonstrating; creating, updating and deleting with Mutations. You should provide examples both running in a Sandbox/playground and examples executed in an Apollo Client.
Example Mutation: startcode-frontend/src/components/AddFriend.tsx ***l. 7***

Example Graphiql: *npm run dev* fullstack-startcode/ localhost/5555/graphql
Example Apollo Client: *npm start* startcode-frontend

Explain the Concept of a Resolver function, and provide a number of simple examples of resolvers you have implemented in a GraphQL Server.
A resolver function has direct access to our database, and returns/updates the data we need.
Like a link between graphql and and our database.

Example: fullstack-startcode/src/graphql/resolvers.ts

Sebastian Bentley (cph-sb287)

Explain the benefits we get from using a library like Apollo-client, compared to using the plain fetch-API

"Apollo Client is a comprehensive state management library for JavaScript that enables you to manage both local and remote data with GraphQL. Use it to fetch, cache, and modify application data, all while automatically updating your UI."[1]

In an Apollo-based React Component, demonstrate how to perform GraphQL Queries, including:

- **Explain the purpose of ApolloClient and the ApolloProvider component**
  - ApolloProvider puts ApolloClient on a global scope, and removes requirements for lifting state up.
  - **Example:** startcode-frontend/src/components/App.tsx

- **Explain the purpose of the gql-function (imported from graphql-tag)**
  - the gql-function allows us to write and run graphql quries and mutations, in JavaScript.
  - **Example:** startcode-frontend/src/components/AllFriends.tsx
- **Explain Custom Hooks used by your Client Code**
  - We can create our hooks form Apollo-client, to get our data in a hook, and determine how we get it (from network or cache etc.)
  - **Example:** startcode-frontend/src/components/AllFriends.tsx ***l. 23***
- **Explain and demonstrate the caching features built into Apollo Client**
  - on our hooks, we can set a cache policy, to determine how our data should be cached
  - **Example:** startcode-frontend/src/components/AddFriend.tsx ***l. 37***

In an Apollo-based React Component, demonstrate how to perform GraphQL Mutations?
Example: startcode-frontend/src/components/AddFriend.

Demonstrate and highlight important parts of a "complete" GraphQL-app using Express and MongoDB on the server-side, and Apollo-Client on the client.

Backend:
Example (resolvers): fullstack-startcode/src/graphql/resolvers.ts

Example (schema): fullstack-startcode/src/graphql/schema.ts

Example (using middelware): fullstack-startcode/src/app.ts

Example (Routes): fullstack-startcode/src/routes/friendRoutesAuth.ts

Frontend:
Example (Apollo-client): startcode-frontend/src/components/App.tsx

Example (gql and use of it): startcode-frontend/src/components/AllFriends.tsx

[1] https://www.apollographql.com/docs/react/