

ONE PAGE SUMMARY

IT & Software > Backend Development > GraphQL

Introduction to GraphQL

Unlocking Data Flexibility with GraphQL

DEFINITION

GraphQL is a query language for APIs that allows clients to request specific data. It enables more efficient data retrieval compared to traditional REST APIs.

KEY CONCEPTS

Queries

Queries are requests for specific data from the server, allowing clients to specify exactly what they need.

Schemas

Schemas define the structure of the data, including types and relationships, serving as a contract between client and server.

Subscriptions

Subscriptions allow clients to receive real-time updates from the server when data changes.

Mutations

Mutations are used to modify server-side data, such as creating, updating, or deleting records.

Resolvers

Resolvers are functions that handle fetching the data for a specific query or mutation.

EXAMPLES

→ Query for user data: `{ user(id: 1) { name, email } }`

→ Mutation to update user: `mutation { updateUser(id: 1, name: 'John') }`

→ Subscription for new messages: `subscription { messageAdded { content } }`

MEMORY TIPS

- ★ Think of GraphQL as a restaurant menu where you order exactly what you want.
- ★ Remember 'CRUD' for Mutations: Create, Read, Update, Delete.
- ★ Visualize Schemas as blueprints that outline how data is structured.

COMMON MISTAKES

- ✗ Over-fetching data by not specifying fields in queries.
- ✗ Neglecting to implement error handling in resolvers.
- ✗ Confusing queries with mutations in usage.

QUICK RECAP

GraphQL is a powerful API query language that allows precise data requests. Understanding queries, mutations, and schemas is essential for effective use. Remember to avoid common pitfalls like over-fetching and ensure proper error handling.