GraphQL Practice. Apollo

by Roman Savchenko

Agenda

1. Quick theory overview
2. Apollo SDK
3. Github GraphQL API
4. Live coding

Rest API vs GraphQL

1. Single endpoint
2. GraphQL exposes single endpoint and responds to queries
3. Increased mobile usage creates need for efficient data loading
4. No more Overfetching and Underfetching

GraphQL Schema

- defines capabilities of the API by specifying how a client can fetch and update data
- represents CONTRACT between client and server
- collection of GraphQL types with special root types

GraphQL Operations

1. Queries
   a. Fetch Data
2. Mutations
   a. Create Data
   b. Update Data
   c. Delete Data

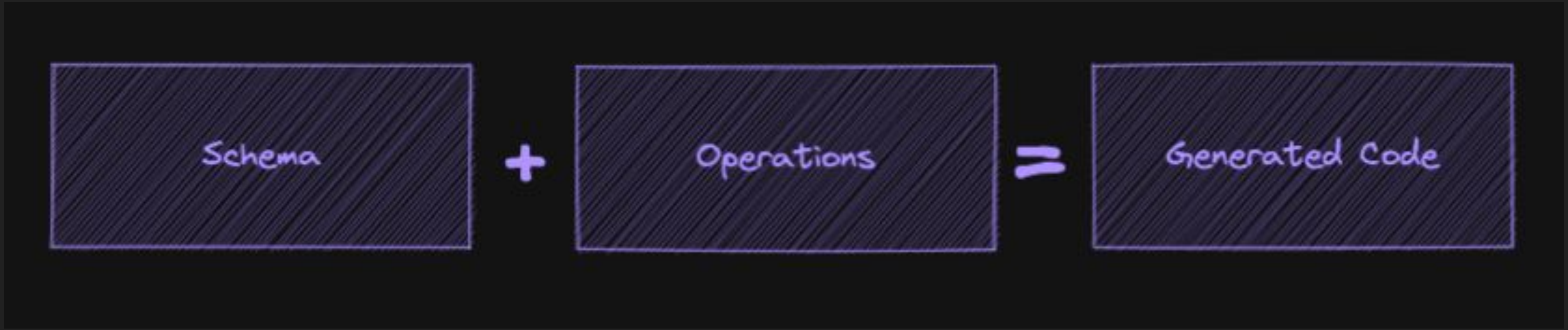
GraphQL Fragments

Reusable components for using in multiple queries and mutations

Apollo SDK

1. Open-source GraphQL client
2. Written in Swift
3. Executes queries and mutations against a GraphQL server and returns results as pre-generated, operation-specific Swift types
4. You don't have to deal with forming spec-compliant GraphQL requests, parsing JSON responses, or manually validating and type-casting response data
5. Includes caching mechanisms that are designed specifically for GraphQL data, enabling you to execute your GraphQL queries against locally cached data directly

Apollo Code Generation



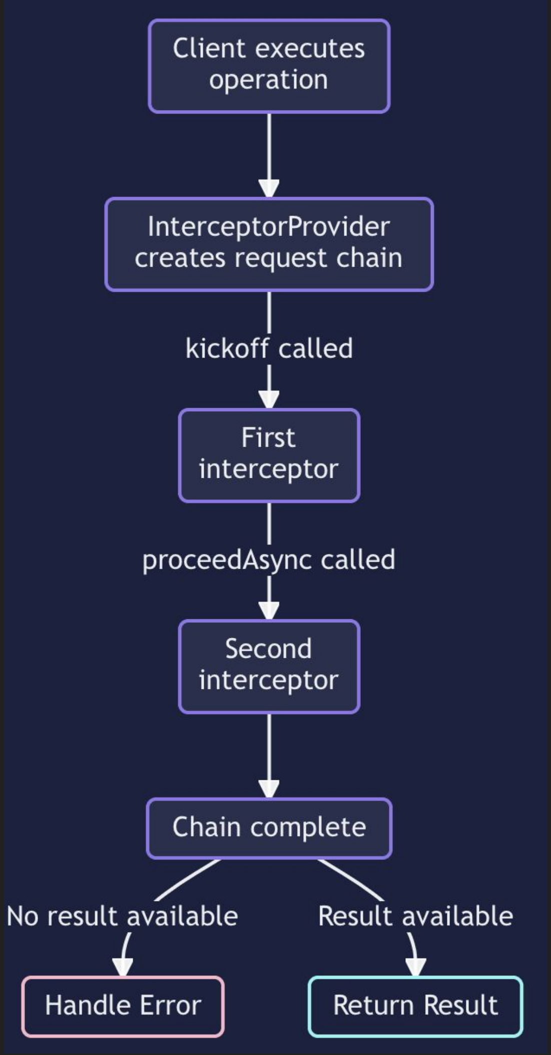
Caching

1. Normalized cache
2. Shorter loading times
3. Reduction of server load and cost
4. Less data usage for users of your application

Interceptors

A request chain defines a sequence of interceptors that handle the lifecycle of a particular GraphQL operation's execution. One interceptor might add custom HTTP headers to a request, while the next might be responsible for actually *sending* the request to a GraphQL server over HTTP. A third interceptor might then write the operation's result to the Apollo iOS cache.

When an operation is executed, an object called an InterceptorProvider generates a RequestChain for the operation. Then, kickoff is called on the request chain, which runs the first interceptor in the chain:

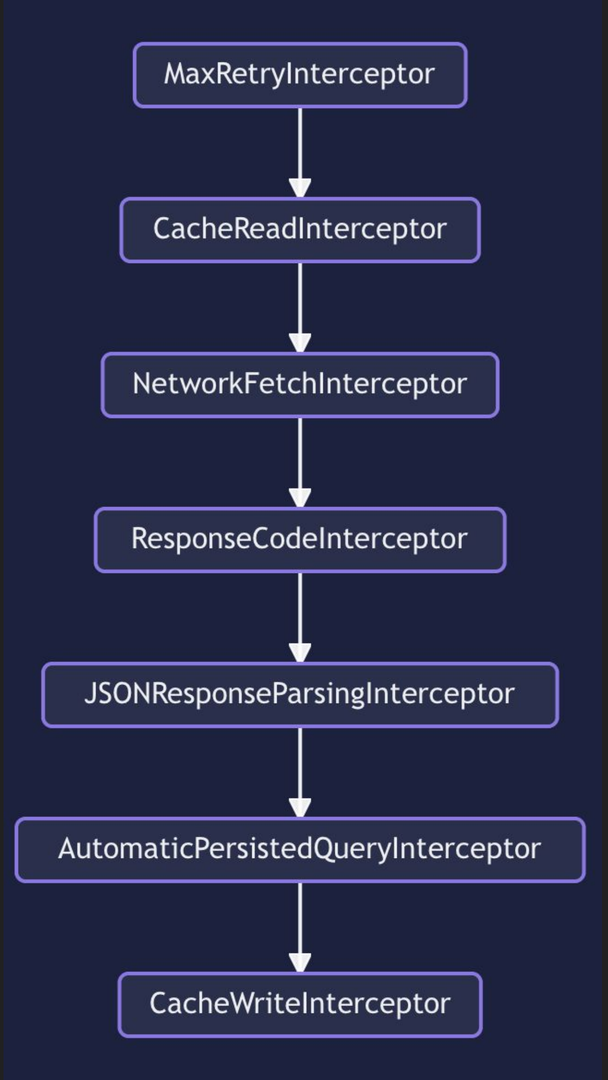


Interceptor provider

To generate a request chain for each GraphQL operation, Apollo iOS passes operations to an object called an interceptor provider. This object conforms to the InterceptorProvider protocol.

Default Provider:

- Reading/writing response data to the normalized cache.
- Sending network requests using URLSession.
- Parsing GraphQL response data in JSON format
- Automatic Persisted Queries.



GitHub GraphQL API

1. Latest repositories (Query)
2. Top rated repositories (Query)
3. Create repository (Mutation)

Live coding

Useful links

1. https://www.apollographql.com/docs/ios/
2. https://github.com/apollographql/apollo-ios
3. https://docs.github.com/en/graphql/overview/explorer

Thank You