Introduction to GraphQL and Relay

Presenter: Eric W. Greene

Welcome to the Webinar!

- Welcome to the webinar on GraphQL and Relay!!!
- We will review a few slides, then experiment with GraphQL and review GraphQL code, followed by a few more slides and a Relay demonstration and code review, then spend the last 10 mins answering any questions (we can go longer if needed for questions)
- The session is being recorded for downloading afterwards

What is GraphQL?

- GraphQL is a replacement interface for traditional REST services
- When developers think of REST services, typically they envision the HTTP verbs and such, but also think of the implementation of business logic, data access, etc…
- A better way to think of REST services is to strictly think of only the HTTP interface, the URIs and Verbs used to access resources without regard to the business logic or database access

What is GraphQL?

- When thinking of the interface only, GraphQL and REST are two ways of accomplishing the same goal
- REST services use a simple combination of HTTP Verb, URI (collection or element), and optional request body content to perform any number of data operations from the perspective of the client
- Compared to technologies such as SOAP or binary data connections, REST services make accessing data via HTTP very easy, and therefore it has become very popular

What is GraphQL?

- While easy and popular, REST service have some problems
- First, there is no standard, each organization can implement their REST services however they want. There are conventions and best practices, but even those can vary from organization to organization
- Second, REST services deliver static data structures unless the developer intentionally codes them to be customized

What is GraphQL?

- When REST services were first implemented most developers looked at their data as a large table (two-dimensional)
- From that table of data, developers retrieve, insert, update and delete data
- A lot has changed in the last 10-20 years

What is GraphQL?

- Data structures today are more complex, they are better thought of as graphs (multi-dimensional) instead of tables (two-dimensional)
- Querying those graphs requires a standard query language, and dynamic result structures which reflect what was actually queried for

What is GraphQL?

- GraphQL, created by Facebook, provides a more flexible, and standardized interface for querying these data graphs, and returning back result sets with a customizable structure
- GraphQL does not care about business logic or data access
- It only cares about how to query the graph, and then how to structure results as requested

Coding GraphQL

- Organizations implement GraphQL using several strategies
  - Create a new services which use GraphQL combined with business logic and data access to provide services
  - Create new GraphQL services which wrap existing REST (and other) services and deliver data through the GraphQL interface (we will do that in our code review)

Coding GraphQL

- GraphQL is a standard, not an implementation
- In the webinar, we will use the Node.js implementation of GraphQL, but GraphQL implementations exist for many languages
  - Java
  - C#
  - Ruby
  - Python
  - And many more… http://graphql.org/code

Coding GraphQL

- GraphQL implementations come with a tool called GraphiQL
- GraphiQL is a web based query tool for GraphQL
- It provides syntax validation, query beautification, code completion and documentation of the GraphQL server's particular data structures (aka schema)
- Queries can be entered, variables can be specified, and the results are displayed

Coding GraphQL

- With GraphQL on the server side, clients can access the GraphQL server via HTTP
- Within a web browser, standard AJAX calls can be made
- The AJAX calls DO NOT conform to the REST Verb/URI pattern, but the calls to GraphQL are standard HTTP calls so using existing AJAX libraries works great, and even REST service clients can be used for testing and debugging

Coding GraphQL

- There are three kinds of GraphQL requests
  - Queries – retrieves application data as a graph of nodes
  - Mutations – changes application data (can update one or many nodes)
  - Introspections – retrieves graph schema (data structure of the various node types and their connections to each other)

Let's Get Started…

- We will explore GraphQL (as compared to a REST service), then review code for GraphQL
- Visual Studio Code will be used as the editor, but Node.js through Express and Webpack will be used to run the various transpilers and bundle our code, and serve the web pages
- JSON Server is being use to serve the REST service data

What is Relay?

- Relay is a framework for connecting GraphQL to client side code
- What problems does Relay solve?
  - Data requirements for views are defined using declarative GraphQL queries
  - The GraphQL queries are co-located with the views
  - Mutations are managed to ensure data consistency, optimistic updates and error checking

What is Relay?

- Relay is part implementation and part standard
- From a standards perspective, Relay defines the GraphQL structure to deliver data, and defines how containers are created on the client to access the Relay Store and deliver the GraphQL data to UI component

What is Relay?

- From this standards perspective
  - Relay does not care which implementation of GraphQL is used so long as the graph structure exposed by the server matches the Relay GraphQL standard
  - Relay does not care which component library is used so long as the component library knows how interact with the Relay Store
- In principle, Relay does not require React

What is Relay?

- In practice, React is required (but there are efforts to make Relay work with other libraries/frameworks such as Angular 2)
- In our code review, our implementation will use the implementation of GraphQL in Express using the Relay GraphQL standard for the structure of the graph
- The client-side code will be using React

What is the Relay GraphQL Standard?

- Relay requires four structures to be present in the GraphQL server
- First, the graph must have single root node
- Second, every node in the graph must have a unique id so that it can be queried directly
- Third, connections between nodes must have a special structure which supports paging through connections
- Fourth, a standard structure for mutations

What is the Relay GraphQL Standard?

- To support this standard any implementation of GraphQL (Java, C#, JavaScript, etc…) can be used, but the coding of the graph structure must follow this standard
- The **graphql-relay** NPM package provides library code to create these structures for the developer

React and Relay

- The NPM package **react-relay** provides client-side library code for connecting the React application to Relay
- It provides containers for connecting the React Component tree to the Relay Store
- It provides library code for configuring routes and mutations

Let's Get Started…

- We will explore Relay and review the code to support Relay both on the server-side and client-side

About Wintellect

consulting/training

who we are

Founded by top experts on Microsoft – Jeffrey Richter, Jeff Prosise, and John Robbins, along with a team of industry leaders – we pull out all the stops to help our customers achieve their goals through advanced software-based consulting and training solutions.

consulting

Wintellect helps you build better software, faster, tackling the tough projects and solving the software and technology questions that help you transform your business.

- Architecture, Analysis and Design
- Full lifecycle software development
- Debugging and Performance tuning
- Database design and development

training

Wintellect's courses are written and taught by some of the most renowned and respected names in the Microsoft programming industry.

- Learn from the best. Access the same training Microsoft's worldwide developers enjoy.
- Real world knowledge and solutions on both current and cutting edge technologies
- Flexibility in training options – onsite, virtual, on demand

Questions???

- To access the source code from today's webinar, download the code from here:
  - https://github.com/training4developers/react_graphql_relay_webinar
  - Email: eric@training4developers.com
- Any questions?