GraphQL with MongoDB for Investment Data Model

mongoDB®

exf

CONTENTS

mongoDB®

exf

GraphQL on MongoDB – Intelligent Data Delivery Meets World Class Data Platform

In the last couple of years, developers, startups, and enterprises alike are adopting GraphQL at a staggering rate. Companies like GitHub, Airbnb, Shopify, Expedia, and Trulia were early adopters of the technology and they have seen impressive results with it. Meanwhile, MongoDB continues its march as a world class data platform with document storage as its central core. This whitepaper brings together these two worlds to help organizations in their quest to modernize their data infrastructure to drive business results.

GraphQL

GraphQL is a query language for APIs and a runtime for fulfilling those queries with existing data. GraphQL provides a complete and understandable description of the data in API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools. Relationships and custom resolvers allow you to even further evolve in GraphQL API to access multiple data sources.

MongoDB Realm

The MongoDB Realm GraphQL API allows client applications to access data stored in a linked MongoDB cluster using any standard GraphQL client. Realm automatically creates GraphQL types for every linked collection that has a defined document schema and evaluates role-based permissions for all GraphQL requests.

Using Realm

- automatically generate JSON schema for MongoDB collections.
- create types and resolvers for that data.
- define custom query resolvers to access other databases or 3rd party APIs.
- test schema using the GraphQL interface inside the MongoDB Atlas UI.

A GraphQL server provides a client with a predefined schema – a model of the data that can be requested from the server. In other words, the schema serves as a middle ground between the client and the server while defining how to access the data.

Major Advantages of using a GraphQL API in an application instead of a REST API:

- Good fit for complex systems and microservices
- Fetching data with a single API call
- No over-fetching and under-fetching problems
- Validation and type checking out-of-the-box
- Autogenerating API documentation
- API evolution without versioning

The GraphQL API lets you access data that you have stored in a MongoDB Atlas cluster or data lake. GraphQL is a declarative, strongly-typed query language for client applications. Clients define the exact data shape and contents that they need in a single request which eliminates over-fetching problems and circumvents the need for multiple costly round trips to the server. The Realm Mobile Platform simplifies many repetitive tasks and solves major problems in the process of mobile application development.

ExfInsights – Exafluence's Platform for Financial Services Domain

ExfInsights is a cloud native, microservices based, open-source driven solution designed to accelerate Data Operations for Financial Services domain in an easy to use, transparent manner. It offers a meta data driven approach for system integration with improved data quality and gold copy creation. Business Analyst's tool set uses the latest predictive analytic technology stack of Big Data, data blending, non-structured databases, Machine Learning (ML), Natural Language Processing (NLP) and Artificial Intelligence (AI) ensembles.

Investment Data Model:

One of the most important components of ExfInsights, especially focused on Asset Management is proprietary 'Investment Data Model'. This model stores all relevant entities & attributes for buy side investment management firms and is built for data consumption, audit and integration with BI tools. A data warehouse is an organized store of data from different sources that is analyzed, visualized and reported for purposes of business intelligence, decision support, and legal or regulatory compliance. ExfInsights Investment Data Model gives solution addresses the problems investment managers face with change management, data quality and governance by handling all of data consolidation activities.

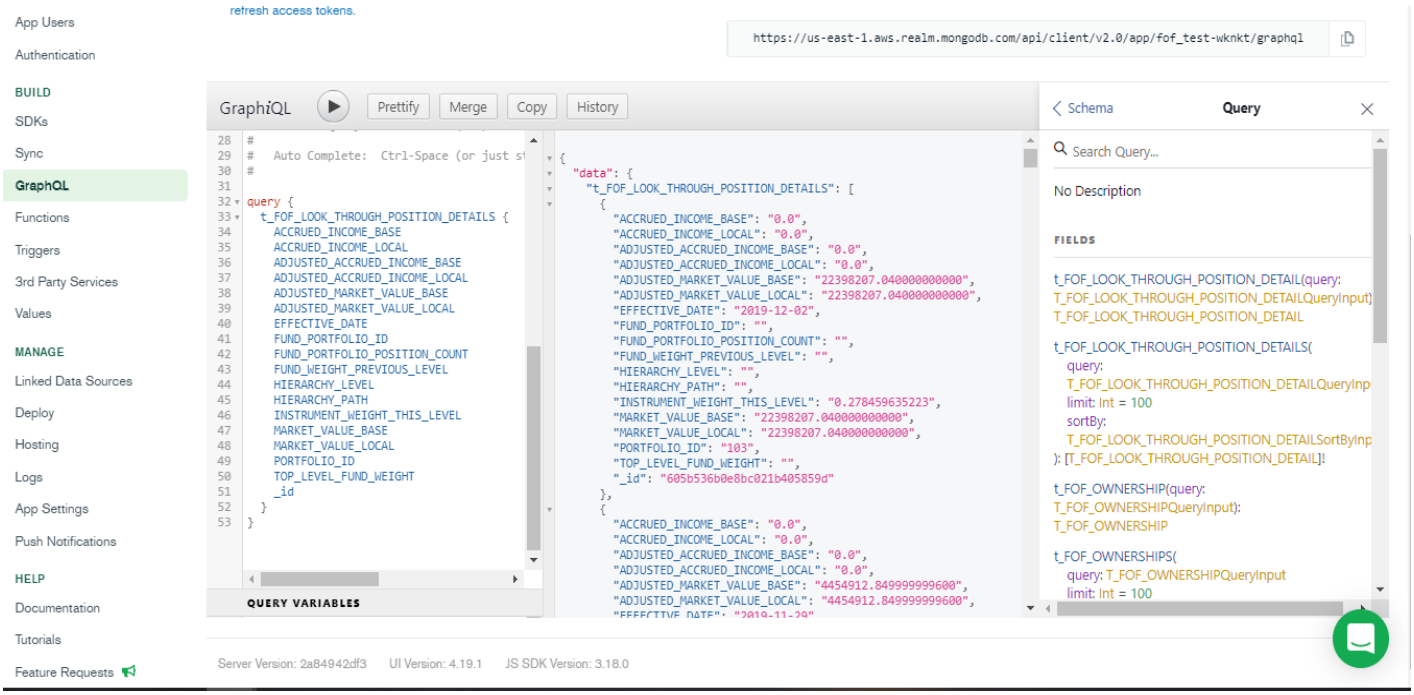
Investment Data Model uses a relational structure with entities connected by relationships defined by foreign keys. Since the Investment management domain is quite complex in terms of the number of entities and relations among them, the information delivery needs are typically satisfied by complex joins which results in poor performance and increase in total cost of ownership. To mitigate these challenges, effort has been taken to convert the relational structure into a document object model using MongoDB. MongoDB database has a flexible schema and provides the capability to store related datasets in a single collection which significantly improves performance. Also, since there is are many instances of unstructured data (Ex: Analyst reports, Manager statements in pdf formats) to be stored along with structured data, MongoDB is proving to be a great platform choice for this domain.

MongoDB Realm and the Investment Data Model:

Using **MongoDB Realm** to convert a relational data model to document data model and fetching exact data using **GraphQL**. In MongoDB Atlas, create a MongoDB Cluster with collections representing the individual entities (data domains). **Realm Application** is created and linked to the MongoDB Atlas cluster.

Data access rules: Realm enforces data collection rules for all incoming GraphQL requests. Realm evaluates a role for every document included in a GraphQL operation and only returns fields and documents that the Realm user has permission to see. GraphQL is strongly typed. Every GraphQL API conforms to a "schema." This schema data model serves as a contract between the client and the server to define what fields are mapped to what types and what kinds of queries a client can make.

Schema Generation: For the data in the collection, MongoDB Realm automatically creates the schema. Once the schema is generated, to ensure it is valid, select the Validate tab to Run Validation. Save, then Review & Deploy the changes. First GraphQL schema and endpoint is created. At this point, all that is needed to be done to surface the MongoDB Atlas data to client applications is completed. The endpoint can be tested using an API program like Postman (incase a front-end application is not available) and the data inside the Realm interface can be queried. The new schema can be tested in GraphQL IDE interface. Documentation for the endpoint can be automatically generated using the Documentation Explorer.



Relationships: Use schema to define relationships that connect each document in one local collection to one or more documents in a different collection on the linked MongoDB Atlas cluster. Create or define a relationship between two collections where Realm evaluates which documents are related to a given document based on a foreign key value. Relationship defines the foreign key fields.

Custom Resolvers: One of the many powerful aspects of GraphQL is the ability to resolve a single query to multiple backend resources. From the GraphQL Documentation. Each field on each type is backed by a function called the resolver which is provided by the GraphQL server developer. When a field is executed, the corresponding resolver is called to produce the next value. User calls a custom resolver, Realm executes the resolver function and returns the result, which must confirm to the resolver's Payload Type. Realm passes the function to any input data from the operation, if applicable. If the resolver is a computed property on a document type, Realm passes the function the specific document that the resolver was called on.

Realm automatically generates resolvers for the collection fields defined in schema. With custom resolvers, it bundles together different data sources - whether from other databases or 3rd party APIs - all under the same schema, and connects to frontend using Realm App ID and GraphQL endpoint.

Anonymous Authentication:

In Realm interface, the Providers tab allows users to log in anonymously. The Anonymous authentication provider allows users to log in to application without providing credentials. Realm provides a wide variety of built-in authentication options, such as email/password, Google, Facebook, and even custom authentications for using application.

About Exafluence

Exafluence is a next generation, Global Technology Solutions company focusing on solving complex business problems leveraging Big Data, Cloud Computing, Blockchain, Industrial IoT and Advanced Analytics - Machine Learning and Artificial Intelligence to enable our client's digital transformation imperatives through innovation, agility and customer centricity.