Developer Diary

Tuesday, 15 October 2024 9:22 AM

Requirements

☑ - The application should allow the querying of upcoming events by location;
☑ - The location of an event should not be updatable after it's been created;
☑ - An event cannot be updated or deleted after the date/time passed;
☑ - Metadata such as created and updated date should be stored and be available to query;
☑ - The listing endpoints should:
  - Require a Location; **get all events filtered by location?**
  - Be sortable by date/time and name both ways (ascending and descending);
  - Be paginated;
  - Be searchable by `name`.
☑ - The Location type should one of:
  - `class`
  - `1-on-1`
  - `workshop`
☑ - Locations should be previously created and referenced in the events by their ID field.
☑ - All Location fields are required except `tags`.
☑ - The Location `name` field should accept:
☑ - Alphanumerical characters;
☑ - Only `-` as a special character.

Questions -

1. **CRUD operations on Location?**
2. **How to reference Events in location? Add event under location schema?**
   a. **Ref location under events**

Test Cases - on resolvers

☑ - The application should allow the querying of upcoming events by location;
☑ - The location of an event should not be updatable after it's been created;
☑ - An event cannot be updated or deleted after the date/time passed;
☐ - Metadata such as created and updated date should be stored and be available to query;
☐ - The listing endpoints should:
  - Require a Location;
  - Be sortable by date/time and name both ways (ascending and descending);
  - Be paginated;
  - Be searchable by `name`.
☐ - The Location type should one of:

NOTE : How to check in unit test if class are these?
  - `class`
  - `1-on-1`
  - `workshop`

☑ - Locations should be previously created and referenced in the events by their ID field.
☐ - All Location fields are required except `tags`.
☐ - The Location `name` field should accept:
  - Alphanumerical characters;
  - Only `-` as a special character.

Note : Fields and validation of required fields/schema in integration/end to end testing?

Unit test vs end to end test
How to test resolvers?
    https://www.robinwieruch.de/graphql-resolver-testing/
    Create a mongo db instance? Mostly integration testing?
        Need to look into this, might take more time
    Mock models, eliminate mongodb

TODO:

1. Testing
   a. Unit tests for Utils
      i. Sorting
      ii. Pagination (Mock Chaining)
         1) https://stackoverflow.com/questions/54561550/jest-mocking-spying-on-mongoose-chained-find-sort-limit-skip-methods
      iii. searching
   b. Integration Tests between MongoDb and GraphQL
2. Pagination cursor
   a. Update using pageInfo and edge
3. Error handling
   a. https://testfully.io/blog/graphql-error-handling/

Javascript vs Typescript
- Javascript for faster development
- Typescript - how typing works between typescript and graphql and moongoose
  - Easier if we use interfaces for models/schema?

Event Management Service?

Things to note:

1. Scalability
2. Speed -
   - Implement keys and indexes for fast queries
     Caching
     Indexing
3. Modularisation
   a. How to structure code
      i. https://www.apollographql.com/blog/how-to-structure-graphql-server-code
4. Error Handling and Logging
5. Security?
   a. Env variables
6. Unit Testing https://www.apollographql.com/docs/apollo-server/testing/testing
7. Documentation?

Modularize for maintainabiliity, easier unit testing

1. Move pagination, sort, filter out of resolver to utils
2. Separate different schema/models
3. Connection -> Database?
   a. Make database extendable
   b. mongoDB is easier for smaller project like this

For validation of dates when inserting/updating:
    middleware mongoose vs checking in resolvers
    https://mongoosejs.com/docs/7.x/docs/middleware.html

Pagination
    Cursor vs offset(more straightforward
    Cursor - base off id
        mongoDB seems to somehow generate ID in relation to Date Time?
        Simplify for now
        Add pageInfo and edge for TODO

Filtering
    Implement filtering based on different fields and operators?
    Simple search for now, don't over engineer based on specs

Random Notes :

1. Tags : when updating, we upsert or update everything
   a. Update everything for now
2. How to store date/time mongoDB <-> GraphQL
   a. DATE