CSCC01 Summer 2022 System Design

CycList

Jun 30, 2022

Contents

CRC

Class name: app.js

Responsibility: Set up the backend server, use schema and resolver to handle each request

Collaborator: schema.js, resolver.js, auth.js

Under database folder:

{

Class name: user.js

Responsibility: set up a database for user in mongodb, provide a connection to mongodb

Collaborator: api/resolver/user.js

Class name: task.js

Responsibility: set up a database for task in mongodb, provide a connection to mongodb

Collaborator: api/resolver/task.js

Class name: tag.js

Responsibility: set up a database for tag(task group) in mongodb, provide a connection to mongodb

Collaborator: api/resolver/tag.js

}

Under schema folder:

{

Class name: schema.js

Responsibility: provide all functions name the backend currently have, include arguments and their type for each function, also provide the return type for each function.

Collaborator: app.js, resolver.js

}

Class name: auth.js

Responsibility: verify the token given from frontend, once verified, extract user id from this token and provide this user id for further action in backend

Collaborator: app.js

Under api/resolver folder

{

Class name: resolver.js

Responsibility: collect all function implementations that tag.js, task.js and user.js provide, so it can be used in app.js to solve each request. It should contain all functions which are listed in schema.js

Collaborator: api/resolver/task.js, api/resolver/user.js, api/resolver/tag.js, app.js, schema.js

Class name: user.js

Responsibility: provide the implementations of each function related to user in schema.js, follow the functions' format and provide all implementations to resolver.js

Collaborator: database/user.js, resolver.js, schema.js

Class name: task.js

Responsibility: provide the implementations of each function related to task in schema.js, follow the functions' format and provide all implementations to resolver.js

Collaborator: database/task.js, resolver.js, schema.js

Class name: tag.js

Responsibility: provide the implementations of each function related to tag in schema.js, follow the functions' format and provide all implementations to resolver.js

Collaborator: database/tag.js, resolver.js, schema.js

}

System Architecture Design

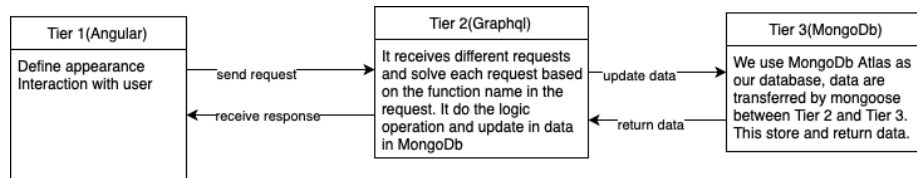Architecture: we use Three-tiered architecture where Tier 1 is our frontend server(using Angular), Tier 2 is our backend server(using Graphql) and Tier 3 is mongodb cloud service

Figure 1: system architecture