2020 Advanced Java Project

technical spec

quiz-test

08 May 2020

Author: Mengwei Yang

Menu

Overview:

The java project for a test. Admin could add some questions and add choices for each question. Then create a test name and assign the created questions to the matching exams. As for user, user could create an account and input their test name to attend a test. The created test is only including the multiple choices.

Goals:

a. The project includes:

1) Question and Choices creating page
2) An exam including different questions, choices, answers page
3) Student test Page

b. Product Requirements:

1) add and update questions
2) assign choices and answers to the questions
3) create user name and help user complete the test

c. Product Requirements:

1) Make sure all of the questions are multiple questions
2) admin could correct created choices
3) user could correct their answers.

Non-goals

I do not set the single questions and answers, because it is said that it is for multiple-choices tests.

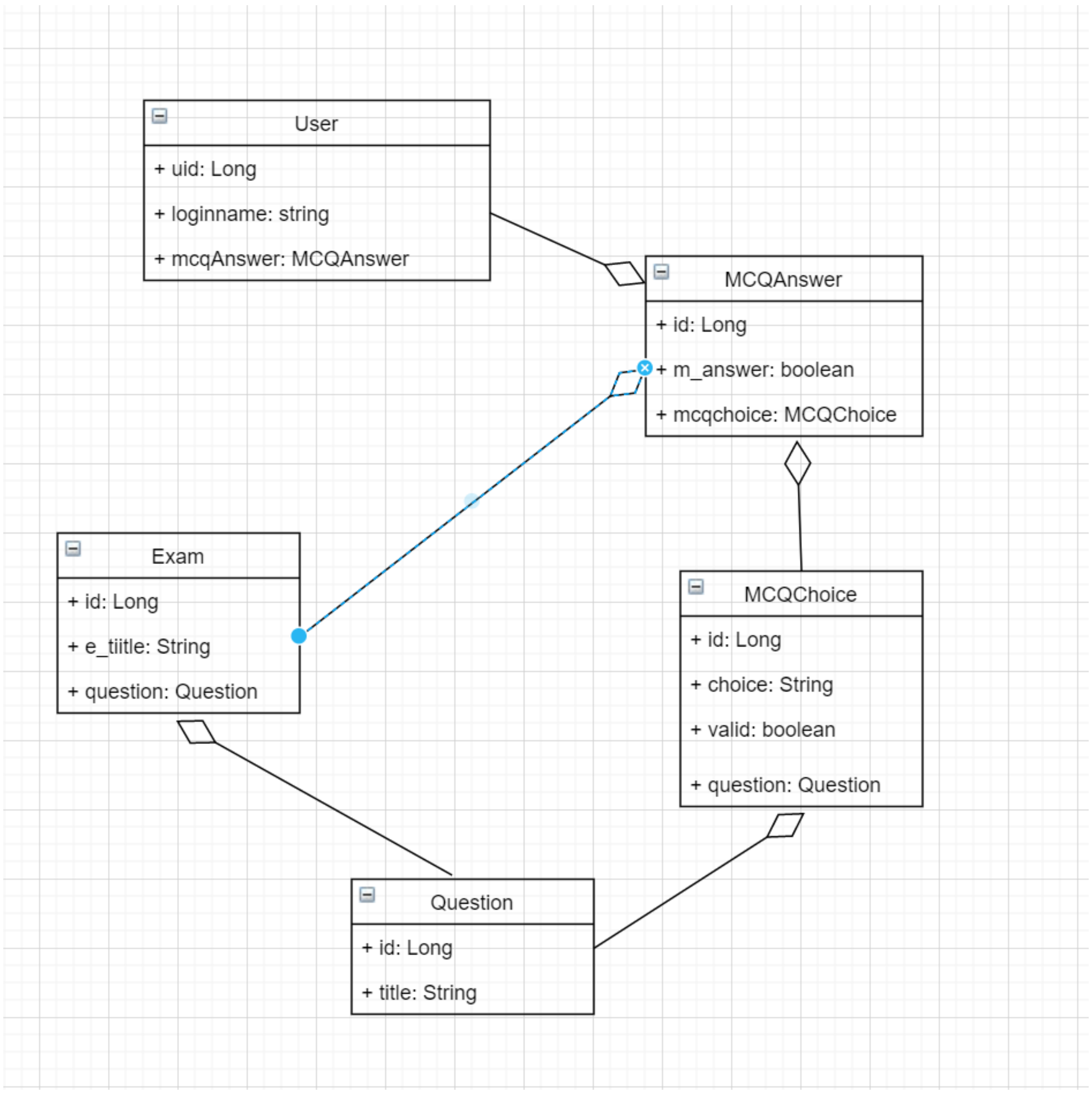
Approach

As for backend,

framework: JPA implementation with spring boot and Spring JPA Data with hibernate.

database: we use h2 online database, because it is easy to connect and operate.

the database structure shows:



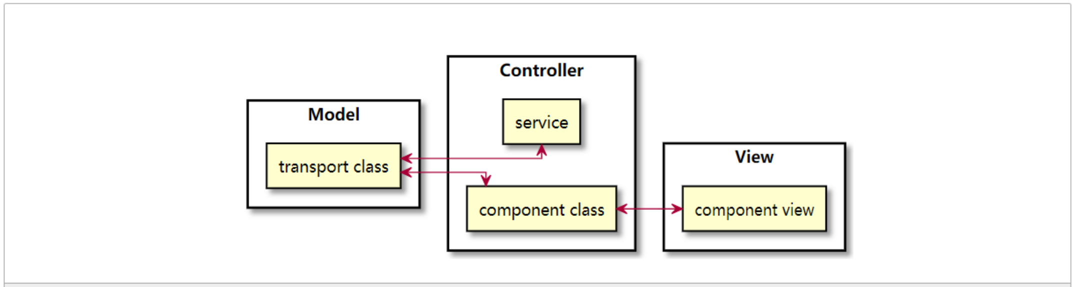
As for Frontend:

using angular2 and rest service to make a connection with backend.

Frontend Skill:

Angular key paradigms:

1). MVC pattern the way Angular apps are built strongly relies on the model-View-controller (MVC) pattern and Angular makes it easy to implement.



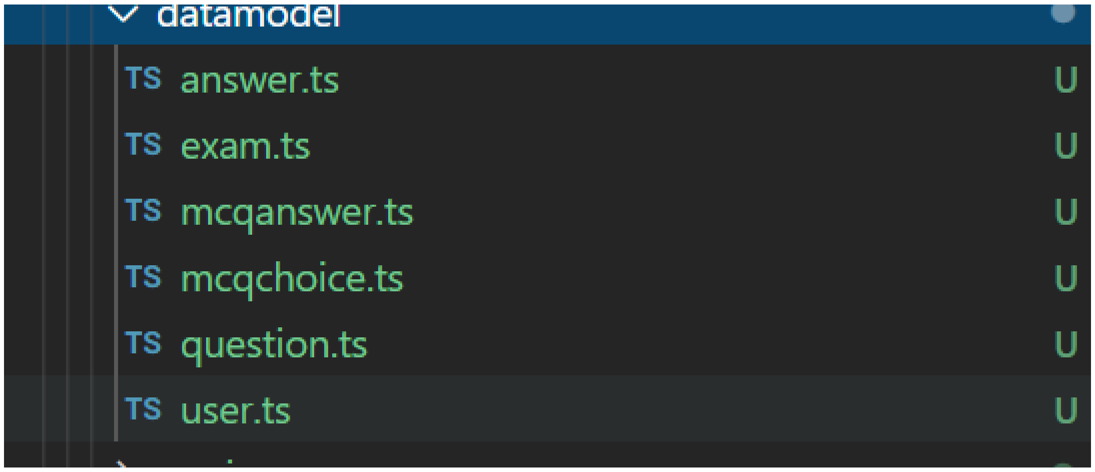
MVC with angular concept mappings

there is my project structure screen shoot:

———in the app folder: there are three models: components, datamodel, services, which correspond with view, controller, service in MVC framework respectively.

1). In the datamodel folder, according to the database structure, I set six parts

(but the answer is not used in this project, because it is related with multiple choices. Each ts.file have set their variables which is corresponded with the DTO in backend code.)

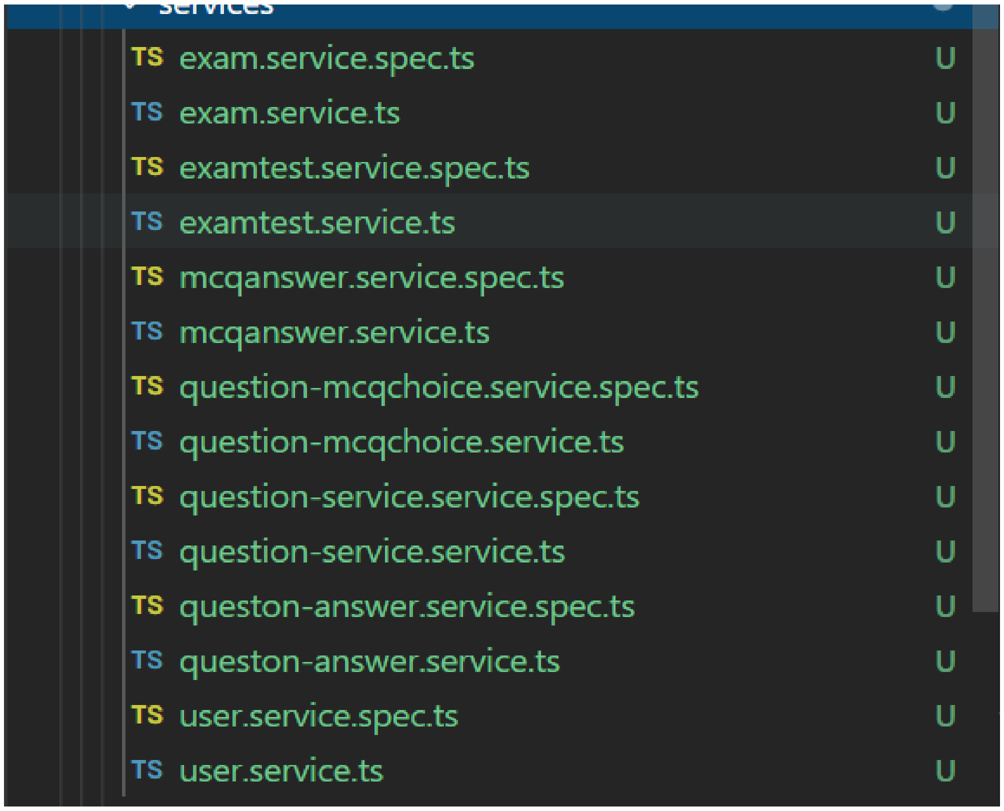


datamodel
TS answer.ts U
TS exam.ts U
TS mcqanswer.ts U
TS mcqchoice.ts U
TS question.ts U
TS user.ts U

2). in the service model, each ts.file uses the url to make a connection with the backend, getting the parameter from the html and according the URL to realize each functions .A REST API exposes interfaces on which actions can be made Typically, this interface would expose at an URL which the front-end could send requests with different http methods like

- `PUT` to create

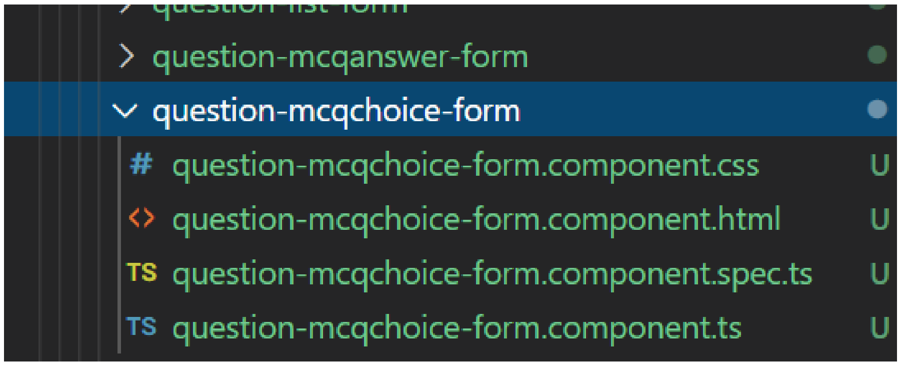
- `POST` to modify



services

TS exam.service.spec.ts U
TS exam.service.ts U
TS examtest.service.spec.ts U
TS examtest.service.ts U
TS mcqanswer.service.spec.ts U
TS mcqanswer.service.ts U
TS question-mcqchoice.service.spec.ts U
TS question-mcqchoice.service.ts U
TS question-service.service.spec.ts U
TS question-service.service.ts U
TS queston-answer.service.spec.ts U
TS queston-answer.service.ts U
TS user.service.spec.ts U
TS user.service.ts U

3). as for component folder, it also attributes by the datamodel, each entity has four parts: for example, question-choice-form, it has .css, .html. .spec.ts and ts. The css and html is for view (front-end) which are users to operate directly. and others, the ts file is to connect the http and transfer the service's function.



question-list-form
question-mcqanswer-form
question-mcqchoice-form
# question-mcqchoice-form.component.css U
question-mcqchoice-form.component.html U
TS question-mcqchoice-form.component.spec.ts U
TS question-mcqchoice-form.component.ts U

As for of technique backend

1. ORM (Object/Relational Mapping)

that allows us to query and manipulate date from any database using an object-oriented language.

2. JPA (java persistence API)

JPA is considered to be the standard industry approach for object to relational mapping, the persistent to make sure the object has to be save in a database.

3. Spring Data JPA:

in our data model, according to the database model, we create our entity. Each entity is corresponded with a table in the database and have its primary id and columns. using the oneToMany and ManytoOne to connect different entity.

4. DAO

adding dependencies in the pom.xml, Then we get a GenericDAO which has some general override function, like getQuery() to realize select from the h2 database, update, delete functions and so on.

Then each entityDAO extends the generic DAO to create their methods.

5. DTO

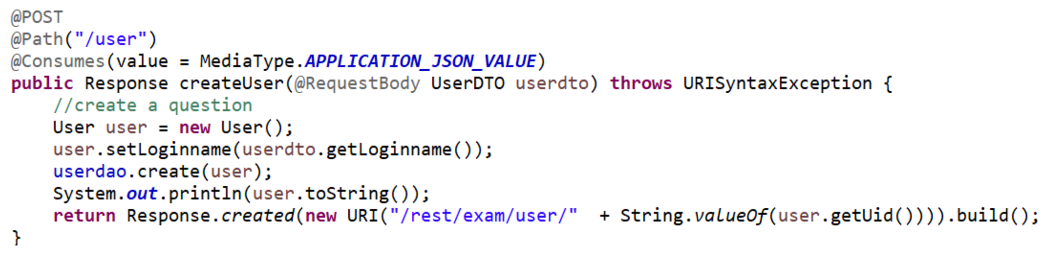
DTO (Data Transfer Object) It is a simple set of aggregated data that needs to be transferred across processes or network boundaries. It should not contain business logic and limit its behavior to activities such as internal consistency checks and basic verification.

6.Resources

In this package, we inject the needed entitydao , and using get, post methods from browser to get data URL.

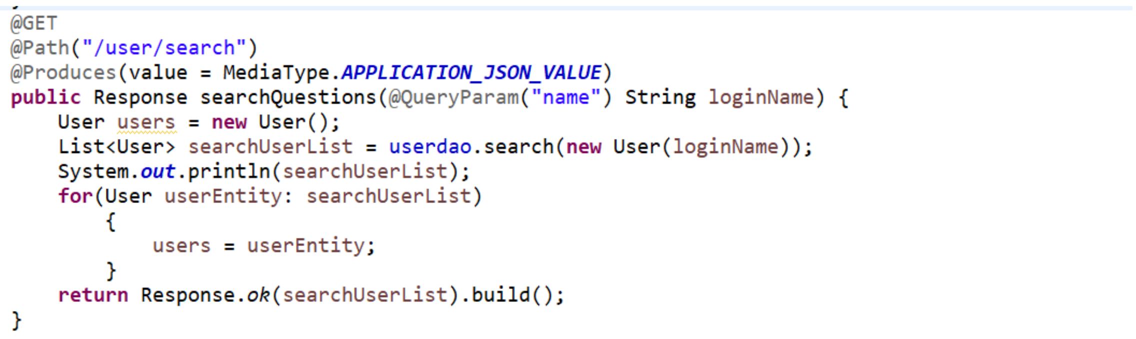
for example:

create a new user, using post



```java
@POST
@Path("/user")
@Consumes(value = MediaType.APPLICATION_JSON_VALUE)
public Response createUser(@RequestBody UserDTO userdto) throws URISyntaxException {
    //create a question
    User user = new User();
    user.setLoginname(userdto.getLoginname());
    userdao.create(user);
    System.out.println(user.toString());
    return Response.created(new URI("/rest/exam/user/"  + String.valueOf(user.getUid()))).build();
}
```

select some data from the database, using get



```java
@GET
@Path("/user/search")
@Produces(value = MediaType.APPLICATION_JSON_VALUE)
public Response searchQuestions(@QueryParam("name") String loginName) {
    User users = new User();
    List<User> searchUserList = userdao.search(new User(loginName));
    System.out.println(searchUserList);
    for(User userEntity: searchUserList)
        {
            users = userEntity;
        }
    return Response.ok(searchUserList).build();
}
```

Measuring Impact

when I create all of the entity, I will as a user to test the whole process and check whether the questions and answer will show or not, and if they are tight or not.

Other Considerations

At first , I decide to put the user into mcqanswerlist, to create @manytoone method ,but I think it is not easy ,it will make a mess, because the mcqanswer has the mcqchoice, and mcqchoice include questions ,Therefore, I put mcqanswer into userlist ,it is easy for me to create the logic method and show the final result in the browser.