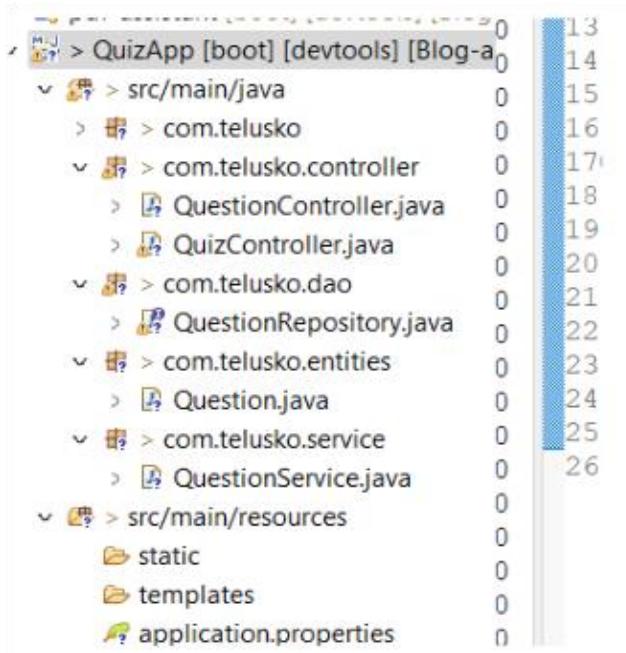


9.Quiz part 1

Project Structure



For QuizController

- **Controller class:** The class is annotated with `@RestController`, indicating it's a REST API controller.
- **Mapping and Endpoint:**
 1. The class is mapped to `"/quiz"` through the `@RequestMapping("quiz")`.
 2. The method `createQuiz` is mapped to a POST request via `@PostMapping("create")`.
- **Method Parameters:** The `createQuiz` method takes three query parameters:
 1. `category` (a String for the quiz category)
 2. `numQ` (an integer for the number of questions)
 3. `title` (a String for the quiz title)
- **Response:** The method responds with a `.ResponseEntity<String>` and returns the status `HttpStatus.OK` with a message "created".

Example

```
@RestController
@RequestMapping("quiz")
public class QuizController {

    // Endpoint to create a quiz using query parameters
    @PostMapping("create")
    public ResponseEntity<String> createQuiz(
        @RequestParam String category,
        @RequestParam int numQ,
        @RequestParam String title) {
        return new ResponseEntity<>("created", HttpStatus.OK);
    }
}
```

- **API Testing Screenshot:**

- The API is tested using a tool like Postman.
- The URL <http://localhost:8080/quiz/create> is being hit with query parameters `category=Packages`, `numQ=5`, and `title=JQuiz`.
- The response is a simple message "created" with a `200 OK` status.

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** <http://localhost:8080/quiz/create?category=Packages&numQ=5&title=JQuiz>
- Params:** A table showing the query parameters:

Key	Value	Description
category	Packages	
numQ	5	
title	JQuiz	
- Body:** A table showing the body parameters:

Key	Value	Description
- Response:** 200 OK | 320 ms | 170 B | `created`