

# Quiz Application Testing

## Creating a new question in a database

The screenshot displays a REST client interface with a POST request to `http://localhost:8080/api/question/create`. The request body is a JSON object with the following fields: `questionTitle`, `option1`, `option2`, `option3`, `option4`, `correctAnswer`, `difficulty`, and `category`. The response body is a JSON object that includes an `id` field (value 10) along with the same data as the request body.

```
POST http://localhost:8080/api/question/create

{
  "questionTitle": "Which of the following is not a Java access modifier?",
  "option1": "private",
  "option2": "protected",
  "option3": "friendly",
  "option4": "public",
  "correctAnswer": "friendly",
  "difficulty": "MEDIUM",
  "category": "JAVA"
}
```

```
{
  "id": 10,
  "questionTitle": "Which of the following is not a Java access modifier?",
  "option1": "private",
  "option2": "protected",
  "option3": "friendly",
  "option4": "public",
  "correctAnswer": "friendly",
  "difficulty": "MEDIUM",
  "category": "JAVA"
}
```

## Creating a new quiz

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/api/quiz/create?category=SPRINGBOOT&level=MEDIUM&title=second`
- Method:** `POST`
- Request URL:** `http://localhost:8080/api/quiz/create?category=JAVA&level=EASY&title=first`
- Params:** A table with 3 rows: 

Key	Value
category	JAVA
level	EASY
title	first
- Status:** 200 OK, Time: 498 ms, Size: 671 B
- Response Body (JSON):**

```
1 {
2   "id": 3,
3   "questions": [
4     {
5       "id": 3,
6       "questionTitle": "Which keyword is used to define a class in Java?",
7       "option1": "define",
8       "option2": "class",
9       "option3": "struct",
10      "option4": "new",
11      "correctAnswer": "class",
12      "difficulty": "EASY",
13      "category": "JAVA"
14    },
15    {
16      "id": 1,
17      "questionTitle": "What is JVM in Java?",
```

## Getting questions to the user

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/quiz/start/3`. The response is a JSON array of two quiz questions. The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, and Settings. The response status is 200 OK, with a time of 121 ms and a size of 491 B.

**Query Params**

Key	Value
Key	Value

**Body** | Cookies | Headers (5) | Test Results | Status: 200 OK | Time: 121 ms | Size: 491 B | Save Response

**Pretty** | Raw | Preview | Visualize | JSON |

```
1 {
2   {
3     "id": 0,
4     "questionTitle": "Which keyword is used to define a class in Java?",
5     "option1": "define",
6     "option2": "class",
7     "option3": "struct",
8     "option4": "new"
9   },
10  {
11    "id": 0,
12    "questionTitle": "What is JVM in Java?",
13    "option1": "Java Visual Machine",
14    "option2": "Java Virtual Method",
15    "option3": "Java Virtual Machine",
16    "option4": "Joint Visual Machine"
17  }
18 }
```

### Getting Score of the user

GET ⌵ `http://localhost:8080/api/quiz/submit/1`

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	Key	Value
	Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵ ↺

1 2

// sending wrong data it will throw exception

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/api/question/create`
- Method:** `POST`
- Body (JSON):**

```
{  "questionTitle": "What is a closure in JavaScript?",  "option1": "A way to define a class",  "option2": "A function having access to the parent scope",  "option3": "A built-in JavaScript library",  "correctAnswer": "A function having access to the parent scope",}
```
- Status:** `400 Bad Request`
- Time:** `1205 ms`
- Size:** `254 B`
- Response Body (JSON):**

```
{  "timestamp": "2025-08-09T03:35:25.493+00:00",  "status": 400,  "error": "Bad Request",  "path": "/api/question/create"}
```

## // swagger API

Servers

http://localhost:8080 - Generated server url ▾

### question-controller



PUT

/api/question/update/{id}



POST

/api/question/create



PATCH

/api/question/updatepatch/{id}



GET

/api/question/getquestions



GET

/api/question/getquestion/{id}



GET

/api/question/getQuestionsInPages



DELETE

/api/question/delete/{id}



### quiz-controller



POST

/api/quiz/create



GET

/api/quiz/submit/{id}



## //API-Doc

Pretty-print ☒

```
{
  "openapi": "3.1.0",
  "info": {
    "title": "OpenAPI definition",
    "version": "v0"
  },
  "servers": [
    {
      "url": "http://localhost:8080",
      "description": "Generated server url"
    }
  ],
  "paths": {
    "/api/question/update/{id}": {
      "put": {
        "tags": [
          "question-controller"
        ],
        "operationId": "updateQuestion",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int32"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/Question"
              }
            }
          },
          "required": true
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/json": {
                "schema": {
                  "$ref": "#/components/schemas/Question"
                }
              }
            }
          }
        }
      }
    }
  }
}
```