

1. adding new question

The screenshot shows a REST client interface. The top section displays a POST request to `http://localhost:8080/api/v1/question/addQuestion` with a JSON body:

```
1 {
2   "questionTitle": "What is java ?",
3   "option1": "Programming",
4   "option2": "Scripting",
5   "option3": "Query",
6   "option4": "None",
7   "correctAnswer": "Programming",
8   "category": "JAVA",
9   "level": "EASY"
10 }
```

The bottom section shows the response status: `200 OK` with headers `394 ms` and `357 B`. The response body is a JSON object:

```
1 {
2   "questionId": 6,
3   "questionTitle": "What is java ?",
4   "option1": "Programming",
5   "option2": "Scripting",
6   "option3": "Query",
7   "option4": "None",
8   "correctAnswer": "Programming",
9   "category": "JAVA",
10  "level": "EASY"
11 }
```

2. Get QuestionById

The screenshot shows a REST client interface. The top section displays a GET request to `http://localhost:8080/api/v1/question/getQuestionById/2`. The bottom section shows the response status: `200 OK` with headers `58 ms` and `376 B`. The response body is a JSON object:

```
1 {
2   "questionId": 2,
3   "questionTitle": "Which of these is not a Java primitive type?",
4   "option1": "int",
5   "option2": "float",
6   "option3": "String",
7   "option4": "boolean",
8   "correctAnswer": "String",
9   "category": "JAVA",
10  "level": "MEDIUM"
11 }
```

3. Update Question

The screenshot shows a REST client interface with a PATCH request to `http://localhost:8080/api/v1/question/updateQuestionById/2`. The request body is a JSON object with the following fields: `questionId` (2), `questionTitle` ("Which of these is not a Java primitive type?"), `option1` ("int"), `option2` ("float"), `option3` ("String"), `option4` ("boolean"), `correctAnswer` ("String"), `category` ("JAVA"), and `level` ("HARD"). The response status is 200 OK, with a response time of 46 ms and a body size of 374 B.

```
1  {
2    "level": "HARD"
3  }
```

```
1  {
2    "questionId": 2,
3    "questionTitle": "Which of these is not a Java primitive type?",
4    "option1": "int",
5    "option2": "float",
6    "option3": "String",
7    "option4": "boolean",
8    "correctAnswer": "String",
9    "category": "JAVA",
10   "level": "HARD"
11 }
```

4. getAllQuestions

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/v1/question/getAllQuestionsPage`. The response status is 200 OK, with a response time of 129 ms and a body size of 1.79 KB. The response body is a JSON array of question objects.

```
1  [
2    {
3      "questionId": 1,
4      "questionTitle": "Which keyword is used to inherit a class in Java?",
5      "option1": "this",
6      "option2": "extends",
7      "option3": "implements",
8      "option4": "super",
9      "correctAnswer": "extends",
10     "category": "JAVA",
11     "level": "EASY"
12   },
13   {
14     "questionId": 3,
15     "questionTitle": "Which annotation in Spring Boot is used to define a REST controller?",
16     "option1": "@Controller",
17     "option2": "@RestController",
18     "option3": "@Service",
19     "option4": "@Component",
20     "correctAnswer": "@RestController",
21     "category": "SPRING",
22     "level": "EASY"
23   }
24 ]
```

5. Custom Exception

HTTP <http://localhost:8080/api/v1/question/updateQuestionById/2> Save

PATCH ▼ <http://localhost:8080/api/v1/question/updateQuestionById/10> Send ▼

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body Cookies Headers (5) Test Results 404 Not Found 172 ms 262 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 {
2   "message": "Question with that ID is not present: 10",
3   "status": "QuestionNotFoundException"
4 }
```

6. Create quiz

POST ▼ <http://localhost:8080/api/v1/quiz/create?category=SQL&level=EASY&title=SQLBasicsQuiz> Send ▼

Params ● Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON** ▼ Beautify

1 {

Body Cookies Headers (5) Test Results 200 OK 106 ms 439 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 {
2   "id": 3,
3   "questions": [
4     {
5       "questionId": 4,
6       "questionTitle": "Which SQL statement is used to extract data from a database?",
7       "option1": "SELECT",
8       "option2": "GET",
9       "option3": "EXTRACT",
10      "option4": "OPEN",
11      "correctAnswer": "SELECT",
12      "category": "SQL",
13      "level": "EASY"
14    }
15  ],
16   "quizTitle": "SQLBasicsQuiz"
17 }
```

7. Submitquiz and adding score

POST ▼ http://localhost:8080/api/v1/quiz/submitQuiz/1

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON** ▼

```
1 [
2   { "id":1, "userAnswer": "extends" },
3   { "id":2, "userAnswer": "string" }
4 ]
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize **JSON** ▼

1 1

8. getquizByID

<http://localhost:8080/api/v1/quiz/getQuizById/1> save

GET ▼ http://localhost:8080/api/v1/quiz/getQuizById/1 **Send** ▼

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings **Cookies**

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 16 ms 320 B **Save Response** ▼

Pretty Raw Preview Visualize **JSON** ▼

```
1 {
2   "id": 1,
3   "questionTitle": "Which keyword is used to inherit a class in Java?",
4   "option1": "this",
5   "option2": "extends",
6   "option3": "implements",
7   "option4": "super"
8 }
9
10
```

9. loggers

```
WARN 11288 --- [Quiz-App] [nio-8080-exec-6] .m.a.ExceptionHandlerExceptionResolver : Resolved [com.wipro.shobhit.spring.quizapp.monorepo.exceptions.QuestionNotFoundException: Question with that ID is n
```

10. Swagger Ui

OpenAPI definition v0 OAS 3.1

/api-docs

Servers

http://localhost:8080 - Generated server url

quiz-controller

POST /api/v1/quiz/submitQuiz/{id}

POST /api/v1/quiz/create

GET /api/v1/quiz/getQuizById/{id}

question-controller

POST /api/v1/question/addNewQuestion

PATCH /api/v1/question/updateQuestionById/{id}

GET /api/v1/question/getQuestionById/{id}

GET /api/v1/question/getAllQuestionsPage

DELETE /api/v1/question/deleteQuestionById/{id}

Api Docs

```
localhost:8080/api-docs

Pretty-print

{
  "openapi": "3.1.0",
  "info": {
    "title": "OpenAPI definition",
    "version": "v0"
  },
  "servers": [
    {
      "url": "http://localhost:8080",
      "description": "Generated server url"
    }
  ],
  "paths": {
    "/api/v1/quiz/submitQuiz/{id}": {
      "post": {
        "tags": [
          "quiz-controller"
        ],
        "operationId": "submitQuiz",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int64"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/Response"
                }
              }
            }
          },
          "required": true
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Response"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```