

<http:PATCH http:DEL http://POST http:POST http:GET http://POST http:POST http:GET http://POST http:GET Untitle=POST http:GET http://POST http:POST http:GET http://>+

http://localhost:8080/api/v1/question/getQuestionByCategory/SPRINGBOOT

Save</>

GEThttp://localhost:8080/api/v1/question/getQuestionByCategory/SPRINGBOOTSend

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Bulk Edit
Key	Value	

BodyCookiesHeaders (5)Test ResultsStatus: 200 OKTime: 728 msSize: 835 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "id": 1,
3    "questionTitle": "Which annotation is used to mark a class as a Spring Boot application?",
4    "option1": "@Configuration",
5    "option2": "@Component",
6    "option3": "@SpringBootApplication",
7    "option4": "@EnableAutoConfiguration",
8    "correctAnswer": "@SpringBootApplication",
9    "difficultyLevel": "EASY",
10   "category": "SPRINGBOOT"
11 },
12 {
13   "id": 2,
14   "questionTitle": "What does @Autowired do?",
15   "option1": "Automatically creates a database connection",
16   "option2": "Automatically injects bean dependencies",
17   "option3": "Automatically starts the server",
18   "option4": "Automatically scans the classpath",
19   "correctAnswer": "Automatically injects bean dependencies",
20   "difficultyLevel": "EASY",
21 }
```

http://localhost:8080/api/v1/question/getAllQuestions

GET

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 74 ms

Size: 835 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "id": 1,
4     "questionTitle": "Which annotation is used to mark a class as a Spring Boot application?",
5     "option1": "@Configuration",
6     "option2": "@Component",
7     "option3": "@SpringBootApplication",
8     "option4": "@EnableAutoConfiguration",
9     "correctAnswer": "@SpringBootApplication",
10    "difficultyLevel": "EASY",
11    "category": "SPRINGBOOT"
12  },
13  {
14    "id": 2,
15    "questionTitle": "What does @Autowired do?",
16    "option1": "Automatically creates a database connection",
17    "option2": "Automatically injects bean dependencies",
18    "option3": "Automatically starts the server",
19    "option4": "Automatically scans the classpath",
20    "correctAnswer": "Automatically injects bean dependencies",
21    "difficultyLevel": "EASY",
22    "category": "SPRINGBOOT"
23  }
24 }
```

POST http://POST http://GET http://POST http://POST http://GET http://POST http://GET http://POST http://POST http://GET http://GET http://PATCH http://

http://localhost:8080/api/v1/question/updateQuestionById/1

PATCH http://localhost:8080/api/v1/question/updateQuestionById/1

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedorawbinaryJSON

```
1
2  "category": "SPRING"
3
```

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 178 msSize: 470 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "id": 1,
3    "questionTitle": "Which annotation is used to mark a class as a Spring Boot application?",
4    "option1": "@Configuration",
5    "option2": "@Component",
6    "option3": "@SpringBootApplication",
7    "option4": "@EnableAutoConfiguration",
8    "correctAnswer": "@SpringBootApplication",
9    "difficultyLevel": "EASY",
10   "category": "SPRING"
11 }
```

<

tp://

POST http://

GET http://

POST http://

POST http://

GET http://

POST http://

GET Untitled

POST http://

GET http://

POST http://

POST http://

GET http://

GET http://

PATCH http://

DEL http://

>

+

...

HTTP

http://localhost:8080/api/v1/question/deleteQuestionById/2

Save

</>

DELETE

▼

http://localhost:8080/api/v1/question/deleteQuestionById/2

Send

▼

⌘

⌘

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	key	Value	

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 78 ms

Size: 193 B

Save Response

▼

Pretty

Raw

Preview

Visualize

Text

▼

⌘

⌘

1

Question Deleted Successfully

tp://GET http://POST http://POST http://GET http://POST http://GET UnfiledPOST http://GET http://POST http://POST http://GET http://GET http://PATCH http://DEL http://POST http://> +see

http://localhost:8080/api/v1/quiz/create?category=SPRING&difficultyLevel=EASY&response="Quiz"

Save</>

POST

http://localhost:8080/api/v1/quiz/create?category=SPRING&difficultyLevel=EASY&response="Quiz"

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	category	SPRING	
<input checked="" type="checkbox"/>	difficultyLevel	EASY	
<input checked="" type="checkbox"/>	response	"Quiz"	
	Key	Value	

Body

Cookies

Headers (5)

Test Results

Status: 200 OKTime: 259 msSize: 514 BSave Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "id": 2,
3    "response": "\"Quiz\"",
4    "question": [
5      {
6        "id": 1,
7        "questionTitle": "Which annotation is used to mark a class as a Spring Boot application?",
8        "option1": "@Configuration",
9        "option2": "@Component",
10       "option3": "@SpringBootApplication",
11       "option4": "@EnableAutoConfiguration",
12       "correctAnswer": "@SpringBootApplication",
13       "difficultyLevel": "EASY",
14       "category": "SPRING"
15     }
16   ]
17 }
```

<

32//

POST http://

POST http://

GET http://

POST http://

GET Untitled

POST http://

GET http://

POST http://

POST http://

GET http://

GET http://

PATCH http

DEL http://

POST http://

GET http://

>

+

...

http://localhost:8080/api/v1/quiz/getQuizById/1

Save

</>

GET

http://localhost:8080/api/v1/quiz/getQuizById/1

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 68 ms

Size: 386 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {}
2
3  {
4    "id": 1,
5    "questionTitle": "Which annotation is used to mark a class as a Spring Boot application?",
6    "option1": "@Configuration",
7    "option2": "@Component",
8    "option3": "@SpringBootApplication",
9    "option4": "@EnableAutoConfiguration"
10 }
```

200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.

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

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

JSON

1

2

3

4

5

6

[{"id": "1", "response": "Quiz"}]

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

0

Status: 200 OK

Time: 60 ms

Size: 165 B

Save Response


```
Console X
Quiz_Application - QuizApplication (1) [Spring Boot App] C:\Program Files\Java\jdk-17\bin\javaw.exe (09-Aug-2025, 9:24:16am elapsed: 0:02:22) [pid: 648]
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown
2025-08-09T09:24:29.201+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set
2025-08-09T09:24:29.348+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for p
2025-08-09T09:24:29.758+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If applicable,
2025-08-09T09:24:30.110+05:30 WARN 648 --- [Quiz_Application] [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by defa
2025-08-09T09:24:30.212+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] o.s.v.b.OptionalValidatorFactoryBean : Failed to set up a Bean Validation provide
2025-08-09T09:24:31.208+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2025-08-09T09:24:31.294+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with co
2025-08-09T09:24:31.312+05:30 INFO 648 --- [Quiz_Application] [ restartedMain] c.w.lalitha.monorepo.QuizApplication : Started QuizApplication in 10.724 seconds
2025-08-09T09:24:49.477+05:30 INFO 648 --- [Quiz_Application] [nio-8080-exec-3] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dis
2025-08-09T09:24:49.481+05:30 INFO 648 --- [Quiz_Application] [nio-8080-exec-3] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-08-09T09:24:49.571+05:30 INFO 648 --- [Quiz_Application] [nio-8080-exec-3] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
2025-08-09T09:24:49.571+05:30 INFO 648 --- [Quiz_Application] [nio-8080-exec-3] c.w.l.m.controller.quizController : Fetching quiz questions for quizId=1
Hibernate:
select
  q1_0.id,
  q1_0.response
from
  quiz q1_0
where
  q1_0.id=?
Hibernate:
select
  q1_0.quiz_id,
  q1_1.id,
  q1_1.category,
  q1_1.correct_answer,
  q1_1.difficulty_level,
  q1_1.option1,
  q1_1.option2,
  q1_1.option3,
  q1_1.option4,
  q1_1.question_title
from
```

OpenAPI definition

v0

OAS 3.0

/v3/api-docs

Servers

http://localhost:8080 - Generated server url

quiz-controller		^
POST	/api/v1/quiz/submitQuiz/{id}	v
POST	/api/v1/quiz/create	v
GET	/api/v1/quiz/getQuizById/{id}	v
question-controller		^
POST	/api/v1/question/addQuestion	v

GET	/api/v1/question/getQuestionByCategory/{category}	▼
GET	/api/v1/question/getAllQuestions	▼
DELETE	/api/v1/question/deleteQuestionById/{id}	▼

Schemas	^
Response	>
Question	>
Quiz	>
QuestionWrapper	>

Pretty-print ☐

```
{
  "openapi": "3.0.1",
  "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": "int32"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/Response"
                }
              }
            }
          },
          "required": true,
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "*/*": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/quiz/create": {
      "post": {
        "tags": [
          "quiz-controller"
        ],
        "operationId": "createQuiz",
        "parameters": [
          {
            "name": "category",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "difficultyLevel",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "response",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/Quiz"
                }
              }
            }
          },
          "required": true,
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "*/*": {
                  "schema": {
                    "type": "string"
                  }
                }
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/addQuestion": {
      "post": {
        "tags": [
          "question-controller"
        ],
        "operationId": "addQuestion",
        "parameters": [
          {
            "name": "category",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "difficultyLevel",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "response",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/Question"
                }
              }
            }
          },
          "required": true,
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "*/*": {
                  "schema": {
                    "type": "string"
                  }
                }
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/updateQuestionById/{id}": {
      "patch": {
        "tags": [
          "question-controller"
        ],
        "operationId": "patchUpdateQuestionById",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int64"
            }
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "#/components/schemas/Question"
                }
              }
            }
          },
          "required": true,
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "*/*": {
                  "schema": {
                    "type": "string"
                  }
                }
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/quiz/getQuizById/{id}": {
      "get": {
        "tags": [
          "quiz-controller"
        ],
        "operationId": "getQuizQuestions",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int32"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/QuestionWrapper"
                  }
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/getQuestionById/{id}": {
      "get": {
        "tags": [
          "question-controller"
        ],
        "operationId": "getQuestionById",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int64"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/getQuestionByCategory/{category}": {
      "get": {
        "tags": [
          "question-controller"
        ],
        "operationId": "getQuestionByCategory",
        "parameters": [
          {
            "name": "category",
            "in": "path",
            "required": true,
            "schema": {
              "type": "string",
              "enum": [
                "JAVA",
                "SPRING",
                "GIT",
                "SPRINGBOOT"
              ]
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Question"
                  }
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/deleteQuestionById/{id}": {
      "delete": {
        "tags": [
          "question-controller"
        ],
        "operationId": "deleteQuestionById",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "integer",
              "format": "int64"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/question/getAllQuestions": {
      "get": {
        "tags": [
          "question-controller"
        ],
        "operationId": "getAllQuestions",
        "parameters": [
          {
            "name": "category",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string",
              "enum": [
                "JAVA",
                "SPRING",
                "GIT",
                "SPRINGBOOT"
              ]
            }
          },
          {
            "name": "difficultyLevel",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string",
              "enum": [
                "EASY",
                "MEDIUM",
                "HARD"
              ]
            }
          },
          {
            "name": "response",
            "in": "query",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Question"
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "components": {
    "schemas": {
      "Response": {
        "type": "integer",
        "format": "int32"
      },
      "Quiz": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": {
              "type": "integer",
              "format": "int64"
            },
            "questionTitle": {
              "type": "string"
            },
            "option1": {
              "type": "string"
            },
            "option2": {
              "type": "string"
            },
            "option3": {
              "type": "string"
            },
            "option4": {
              "type": "string"
            },
            "correctAnswer": {
              "type": "string"
            },
            "difficultyLevel": {
              "type": "string",
              "enum": [
                "EASY",
                "MEDIUM",
                "HARD"
              ]
            },
            "category": {
              "type": "string",
              "enum": [
                "JAVA",
                "SPRING",
                "GIT",
                "SPRINGBOOT"
              ]
            }
          }
        }
      },
      "Question": {
        "type": "object",
        "properties": {
            "id": {
              "type": "integer",
              "format": "int32"
            },
            "getResponse": {
              "type": "string"
            },
            "question": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/Question"
              }
            },
            "QuestionWrapper": {
              "type": "object",
              "properties": {
                "id": {
                  "type": "integer",
                  "format": "int64"
                },
                "questionTitle": {
                  "type": "string"
                },
                "option1": {
                  "type": "string"
                },
                "option2": {
                  "type": "string"
                },
                "option3": {
                  "type": "string"
                },
                "option4": {
                  "type": "string"
                }
              }
            }
          }
      }
    }
  }
}
```