

API DOCUMENTATION

```
localhost:8080/v3/api-docs

Pretty-print

{"openapi":"3.0.1","info":{"title":"OpenAPI definition","version":"v0"},"servers":[{"url":"http://localhost:8080","description":"Generated server url"}],"paths":{"questions/get/{id}":{"put":{"tags":["question-controller"],"operationId":"getUpdateQuestion","parameters":[{"name":"id","in":"path","required":true,"schema":{"type":"integer","format":"int64"}}],"requestBody":{"content":{"application/json":{"schema":{"$ref":"#/components/schemas/Question"}}},"required":true},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"$ref":"#/components/schemas/Question"}}}}}}},"questions/add":{"post":{"tags":["question-controller"],"operationId":"addQuestions","requestBody":{"content":{"application/json":{"schema":{"$ref":"#/components/schemas/Question"}}},"required":true},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"$ref":"#/components/schemas/Question"}}}}}}},"api/quizzes/submit/{id}":{"post":{"tags":["quiz-controller"],"operationId":"submitQuiz","parameters":{"$ref":"#/components/schemas/Response"}}},"required":true},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"integer","format":"int64"}}}}}}},"api/quizzes/create":{"post":{"tags":["quiz-controller"],"operationId":"createQuiz","parameters":[{"name":"category","in":"query","required":true,"schema":{"type":"string"}},{"name":"level","in":"query","required":true,"schema":{"type":"string"}},{"name":"title","in":"query","required":true,"schema":{"type":"string"}}],"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"$ref":"#/components/schemas/Quiz"}}}}}}},"questions/patch/{id}":{"patch":{"tags":["question-controller"],"operationId":"getPatch","parameters":[{"name":"id","in":"path","required":true,"schema":{"type":"integer","format":"int64"}}],"requestBody":{"content":{"application/json":{"schema":{"$ref":"#/components/schemas/Question"}}},"required":true},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"$ref":"#/components/schemas/Question"}}}}}}},"questions/getPage":{"get":{"tags":["question-controller"],"operationId":"getQuestion","parameters":[{"name":"pageable","in":"query","required":true,"schema":{"$ref":"#/components/schemas/Pageable"}}],"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"$ref":"#/components/schemas/PageQuestion"}}}}}}},"questions/difficulty/{difficultyLevel}":{"get":{"tags":["question-controller"],"operationId":"getByDifficulty","parameters":[{"name":"difficultyLevel","in":"path","required":true,"schema":{"type":"string","enum":["EASY","MEDIUM","DIFFICULT"]}}],"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"array","items":{"$ref":"#/components/schemas/Question"}}}}}}},"questions/all":{"get":{"tags":["question-controller"],"operationId":"getQuestionAll","responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"array","items":{"$ref":"#/components/schemas/Question"}}}}}}},"questions/Category/{category}":{"get":{"tags":["question-controller"],"operationId":"getByCategory","parameters":{"name":"category","in":"path","required":true,"schema":{"type":"string","enum":["JAVA","PYTHON","SPRING_BOOT","DATABASE","WEB_TECH","APTITUDE"]}}},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"array","items":{"$ref":"#/components/schemas/Question"}}}}}}},"api/quizzes/getQuizById/{id}":{"get":{"tags":["quiz-controller"],"operationId":"getQuizQuestions","parameters":{"name":"id","in":"path","required":true,"schema":{"type":"integer","format":"int64"}}},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"array","items":{"$ref":"#/components/schemas/QuestionWrapper"}}}}}}},"questions/delete/{id}":{"delete":{"tags":["question-controller"],"operationId":"deleteQuestions","parameters":{"name":"id","in":"path","required":true,"schema":{"type":"integer","format":"int64"}}},"responses":{"200":{"description":"OK","content":{"*/":{"schema":{"type":"string"}}}}}}},"components":{"schemas":{"Question":{"type":"object","properties":{"id":{"type":"integer","format":"int64"},"questionTitle":{"type":"string"},"optionA":{"type":"string"},"optionB":{"type":"string"},"optionC":{"type":"string"},"optionD":{"type":"string"},"correctAnswer":{"type":"string"},"difficultyLevel":{"type":"string","enum":["EASY","MEDIUM","DIFFICULT"]},"category":{"type":"string","enum":["JAVA","PYTHON","SPRING_BOOT","DATABASE","WEB_TECH","APTITUDE"]}}},"Response":{"type":"object","properties":{"id":{"type":"integer","format":"int64"},"response":{"type":"string"}}},"Quiz":{"type":"object","properties":{"id":{"type":"integer","format":"int64"},"title":{"type":"string"},"questions":{"type":"array","items":{"$ref":"#/components/schemas/Question"}}},"Pageable":{"type":"object","properties":{"page":{"minimum":0,"type":"integer","format":"int32"},"size":{"minimum":1,"type":"integer","format":"int32"},"sort":{"type":"array","items":{"type":"string"}}},"PageQuestion":{"type":"object","properties":{"totalPages":{"type":"integer","format":"int32"},"totalElements":{"type":"integer","format":"int64"},"size":{"type":"integer","format":"int32"},"content":{"type":"array","items":{"$ref":"#/components/schemas/Question"}}},"number":{"type":"integer","format":"int32"},"sort":{"type":"array","items":{"$ref":"#/components/schemas/SortObject"}}},"first":{"type":"boolean"},"last":{"type":"boolean"},"numberOfElements":{"type":"integer","format":"int32"},"pageable":{"$ref":"#/components/schemas/PageableObject"},"empty":{"type":"boolean"},"PageableObject":{"type":"object","properties":{"offset":{"type":"integer","format":"int64"},"sort":{"type":"array","items":{"$ref":"#/components/schemas/SortObject"}}},"paged":{"type":"boolean"},"unpaged":{"type":"boolean"},"pageNumber":{"type":"integer","format":"int32"},"pageSize":{"type":"integer","format":"int32"},"SortObject":{"type":"object","properties":{"direction":{"type":"string"},"nullHandling":{"type":"string"},"ascending":{"type":"boolean"},"property":{"type":"string"},"ignoreCase":{"type":"boolean"}}},"QuestionWrapper":{"type":"object","properties":{"id":{"type":"integer","format":"int64"},"questionTitle":{"type":"string"},"optionA":{"type":"string"},"optionB":{"type":"string"},"optionC":{"type":"string"},"optionD":{"type":"string"}}}}}}}
```

ons

postman tutorial / New Request

Save

POST http://localhost:8080/questions/add

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "questionTitle": "Which SQL command is used to retrieve data?",
3   "optionA": "GET",
4   "optionB": "FETCH",
5   "optionC": "SELECT",
6   "optionD": "SHOW",
7   "correctAnswer": "SELECT",
8   "difficultyLevel": "EASY",
9   "category": "DATABASE"
10 }
11
12
13
14
```

JD, test & variable

Body Cookies Headers (5) Test Results

200 OK • 404 ms • 376 B

JSON Preview Visualize

```
1 {
2   "id": 6,
3   "questionTitle": "Which SQL command is used to retrieve data?",
4   "optionA": "GET",
5   "optionB": "FETCH",
6   "optionC": "SELECT",
7   "optionD": "SHOW",
8   "correctAnswer": "SELECT",
9   "difficultyLevel": "EASY",
10  "category": "DATABASE"
11 }
```

2) Get Question By Id

ions

postman tutorial / New Request

Save

PUT http://localhost:8080/questions/get/3

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "questionTitle": "Which SQL command is used to retrieve data?",
3   "optionA": "GET",
4   "optionB": "FETCH",
5   "optionC": "SELECT",
6   "optionD": "SHOW",
7   "correctAnswer": "SELECT",
8   "difficultyLevel": "EASY",
9   "category": "DATABASE"
10 }
11
```

UD, test & variable

Body Cookies Headers (5) Test Results

200 OK • 14 ms • 376 B

JSON Preview Visualize

```
1 {
2   "id": 3,
3   "questionTitle": "Which SQL command is used to retrieve data?",
4   "optionA": "GET",
5   "optionB": "FETCH",
6   "optionC": "SELECT",
7   "optionD": "SHOW",
8   "correctAnswer": "SELECT",
9   "difficultyLevel": "EASY",
10  "category": "DATABASE"
11 }
```

3) Delete Question By Id

The screenshot shows the Postman interface for a new request. The method is set to **DELETE** and the URL is `http://localhost:8080/questions/delete/5`. The **Body** tab is selected, and the content type is **none**. The message "This request does not have a body" is displayed. The **Test Results** tab shows a successful response with a status of **200 OK** and a response body of `1 Question deleted successfully`.

4) Patching

The screenshot shows the Postman interface for a new request. The method is set to **PATCH** and the URL is `http://localhost:8080/questions/patch/3`. The **Body** tab is selected, and the content type is **JSON**. The request body is a JSON object:

```
{  "questionTitle": "Which SQL command is used to retrieve data?",  "optionA": "GET",  "optionB": "FETCH",  "optionC": "SELECT",  "optionD": "SHOW",  "correctAnswer": "SELECT",  "difficultyLevel": "EASY",  "category": "DATABASE"}
```

. The **Test Results** tab shows a successful response with a status of **200 OK**, a response time of **12 ms**, and a response size of **376 B**. The response body is a JSON object:

```
{  "id": 3,  "questionTitle": "Which SQL command is used to retrieve data?",  "optionA": "GET",  "optionB": "FETCH",  "optionC": "SELECT",  "optionD": "SHOW",  "correctAnswer": "SELECT",  "difficultyLevel": "EASY",  "category": "DATABASE"}
```

GET ALL QUESTIONS

postman tutorial / New Request

GET http://localhost:8080/questions/all

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "questionTitle": "Which SQL command is used to retrieve data?",
3   "optionA": "GET",
4   "optionB": "FETCH",
5   "optionC": "SELECT",
6   "optionD": "SHOW",
7   "correctAnswer": "SELECT",
8   "difficultyLevel": "EASY",
9   "category": "DATABASE"
10 }
11
```

Body Cookies Headers (5) Test Results

200 OK • 111 ms • 804 B

JSON Preview Visualize

```
1 [
2   {
3     "id": 1,
4     "questionTitle": "Which SQL command is used to retrieve data?",
5     "optionA": "GET",
6     "optionB": "FETCH",
7     "optionC": "SELECT",
8     "optionD": "SHOW",
9     "correctAnswer": "SELECT",
10    "difficultyLevel": "EASY",
11    "category": "DATABASE"
12  },
13 ]
```

PAGINATION

postman tutorial

New Request

GET http://localhost:8080/questions/getPage

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "questionTitle": "Which SQL command is used to retrieve data?",
3   "optionA": "GET",
4   "optionB": "FETCH",
5   "optionC": "SELECT",
6   "optionD": "SHOW",
7   "correctAnswer": "SELECT",
8   "difficultyLevel": "EASY",
9   "category": "DATABASE"
10 }
11
```

Body Cookies Headers (5) Test Results

200 OK • 49 ms • 1.09 KB

JSON Preview Visualize

```
1 content : [
2   {
3     "id": 1,
4     "questionTitle": "Which SQL command is used to retrieve data?",
5     "optionA": "GET",
6     "optionB": "FETCH",
7     "optionC": "SELECT",
8     "optionD": "SHOW",
9     "correctAnswer": "SELECT",
10    "difficultyLevel": "EASY",
11    "category": "DATABASE"
12  },
13 ]
```

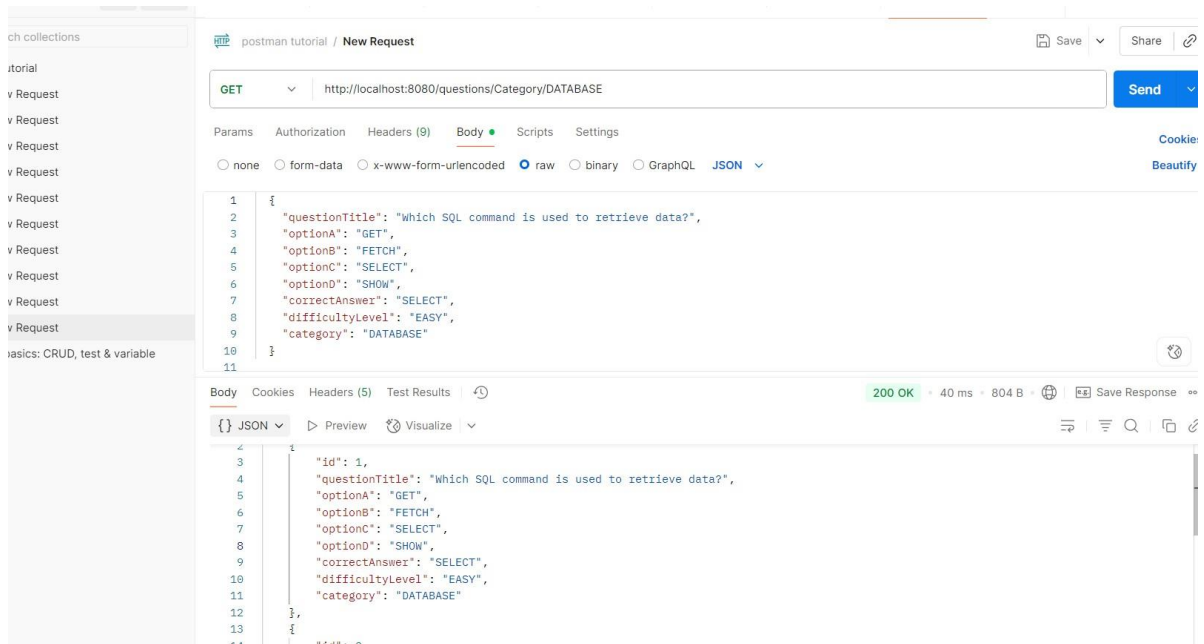
Find and replace Console

Postbot Runner Start Proxy Cookies

Search

ENG

GET DATA BY CATEGORY



postman tutorial / New Request

GET http://localhost:8080/questions/Category/DATABASE

Params Authorization Headers (9) Body Scripts Settings

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

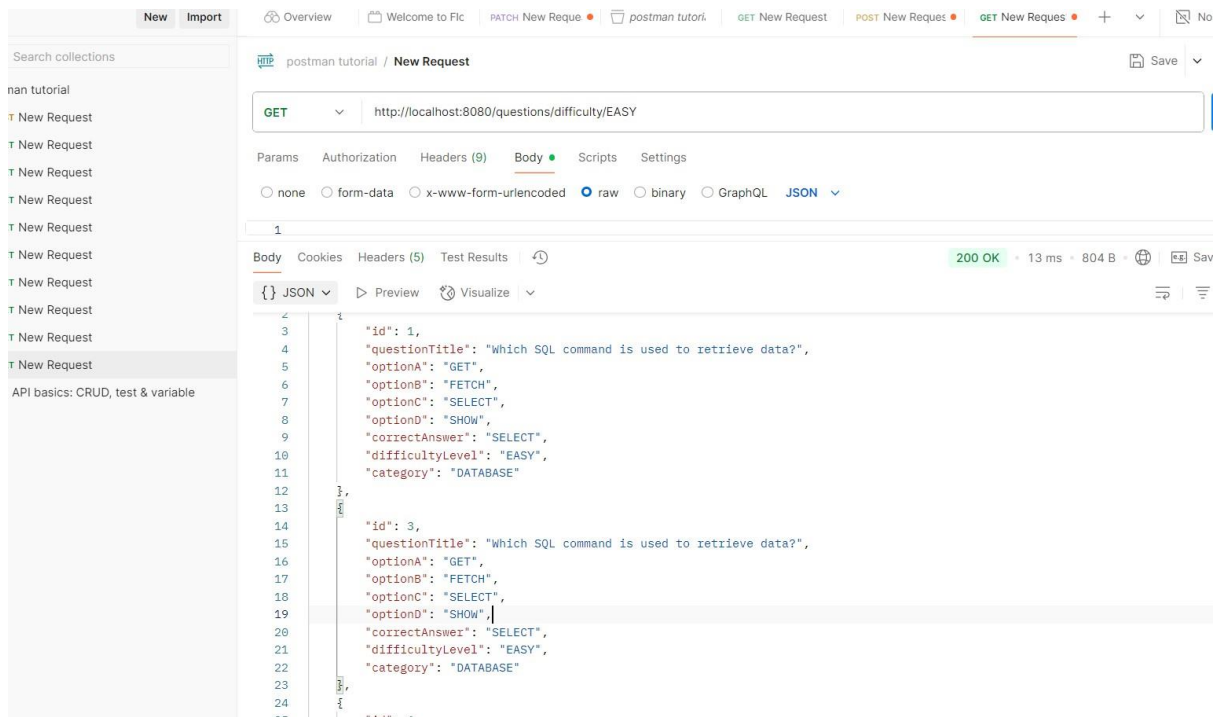
```
1 {
2   "questionTitle": "Which SQL command is used to retrieve data?",
3   "optionA": "GET",
4   "optionB": "FETCH",
5   "optionC": "SELECT",
6   "optionD": "SHOW",
7   "correctAnswer": "SELECT",
8   "difficultyLevel": "EASY",
9   "category": "DATABASE"
10 }
11
```

200 OK • 40 ms • 804 B • Save Response

JSON Preview Visualize

```
1 [
2   {
3     "id": 1,
4     "questionTitle": "Which SQL command is used to retrieve data?",
5     "optionA": "GET",
6     "optionB": "FETCH",
7     "optionC": "SELECT",
8     "optionD": "SHOW",
9     "correctAnswer": "SELECT",
10    "difficultyLevel": "EASY",
11    "category": "DATABASE"
12  },
13 ]
14
```

GET QUESTION BY DIFFICULTY



New Import Overview Welcome to Flc PATCH New Reque postman tutori GET New Request POST New Request GET New Request + No

Search collections

nan tutorial

T New Request

T New Request

T New Request

T New Request

T New Request

T New Request

T New Request

T New Request

T New Request

API basics: CRUD, test & variable

postman tutorial / New Request

GET http://localhost:8080/questions/difficulty/EASY

Params Authorization Headers (9) Body Scripts Settings

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

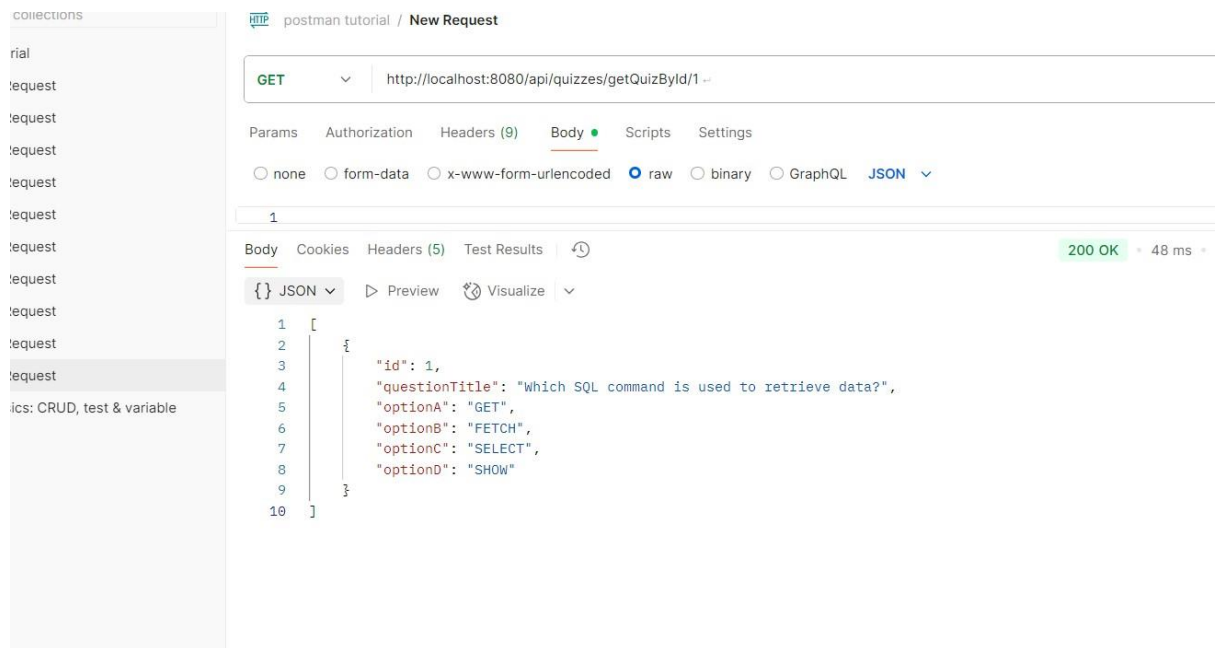
1

Body Cookies Headers (5) Test Results

200 OK • 13 ms • 804 B • Save

JSON Preview Visualize

```
1 [
2   {
3     "id": 1,
4     "questionTitle": "Which SQL command is used to retrieve data?",
5     "optionA": "GET",
6     "optionB": "FETCH",
7     "optionC": "SELECT",
8     "optionD": "SHOW",
9     "correctAnswer": "SELECT",
10    "difficultyLevel": "EASY",
11    "category": "DATABASE"
12  },
13   {
14     "id": 3,
15     "questionTitle": "Which SQL command is used to retrieve data?",
16     "optionA": "GET",
17     "optionB": "FETCH",
18     "optionC": "SELECT",
19     "optionD": "SHOW",
20     "correctAnswer": "SELECT",
21     "difficultyLevel": "EASY",
22     "category": "DATABASE"
23   },
24 ]
25
```



GET QUIZ BY ID

Postman interface showing a new request configuration. The request is a POST to `http://localhost:8080/api/quizzes/submit/1`. The body is raw JSON, containing an array of quiz objects:

```
1 [
2   {
3     "id": 1,
4     "response": "FETCH"
5   },
6   {
7     "id": 2,
8     "response": "C"
9   },
10  {
11    "id": 6,
12    "response": "B"
13  }
14 ]
15
```

The response status is 200 OK, 12 ms, 165 B.

CALCULATE SCORE

CREATE QUIZ

Postman interface showing a new request configuration. The request is a POST to `http://localhost:8080/api/quizzes/create?category=Java&level=Easy&title=Basic Quiz`. The body is raw JSON, containing a quiz object:

```
1 {
2   "id": 3,
3   "title": "Basic Quiz\n\n",
4   "questions": []
5 }
```

The response status is 200 OK, 100 ms, 212 B.

LOGGINGS

```
    q1_0.question_title
  from
    question q1_0
  where
    q1_0.id=?
2025-08-08T19:11:08.362+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-2] c.w.s.q.controller.QuestionController : /getpage is done successfull
Hibernate:
select
  q1_0.id,
  q1_0.category,
  q1_0.correct_answer,
  q1_0.difficulty_level,
  q1_0.optiona,
  q1_0.optionb,
  q1_0.optionc,
  q1_0.optiond,
  q1_0.question_title
from
  question q1_0
limit
  ?, ?
2025-08-08T19:11:08.376+05:30 WARN 13724 --- [Quiz-App] [nio-8080-exec-2] rations$PageModule$WarningLoggingModifier : Serializing PageImpl instances as-is is not supported, meaning that th
For a stable JSON structure, please use Spring Data's PagedModel (globally via @EnableSpringDataWebSupport(pageSerializationMode = VIA_DTO))
or Spring HATEOAS and Spring Data's PagedResourcesAssembler as documented in https://docs.spring.io/spring-data/commons/reference/repositories/core-extensions.html#core.web.pageabl
2025-08-08T19:12:00.280+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-3] c.w.s.q.controller.QuestionController : /getCategory is done
Hibernate:
select
  q1_0.id,
  q1_0.category,
  q1_0.correct_answer,
  q1_0.difficulty_level,
  q1_0.optiona,
  q1_0.optionb,
  q1_0.optionc,
  q1_0.optiond,
  q1_0.question_title
from
  question q1_0
where
  q1_0.category=?
2025-08-08T19:12:37.485+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-4] c.w.s.q.controller.QuestionController : getDifficulty is done
Hibernate:
```

SWAGGER-UI

Servers

http://localhost:8080 - Generated server uri

question-controller

PUT

/questions/get/{id}

POST

/questions/add

PATCH

/questions/patch/{id}

GET

/questions/getPage

GET

/questions/difficulty/{difficultyLevel}

GET

/questions/all

GET

/questions/Category/{category}

DELETE

/questions/delete/{id}

quiz-controller

SWAGGER-UI

Servers

http://localhost:8080 - Generated server url

question-controller

PUT

/questions/get/{id}

POST

/questions/add

PATCH

/questions/patch/{id}

GET

/questions/getPage

GET

/questions/difficulty/{difficultyLevel}

GET

/questions/all

GET

/questions/Category/{category}

DELETE

/questions/delete/{id}

quiz-controller

```
2025-08-08T19:02:44.624+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-2] c.w.s.q.controller.QuestionController : this question is coming from Question(id=null, questionTitle=Which SQL command is used to
Hibernate:
insert
into
question
(category, correct_answer, difficulty_level, optiona, optionb, optionc, optiond, question_title)
values
(?, ?, ?, ?, ?, ?, ?, ?)
Hibernate:
select
q1_0.id,
q1_0.category,
q1_0.correct_answer,
q1_0.difficulty_level,
q1_0.optiona,
q1_0.optionb,
q1_0.optionc,
q1_0.optiond,
q1_0.question_title
from
question q1_0
where
q1_0.id=?
Hibernate:
delete
from
question
where
id=?
2025-08-08T19:04:10.175+05:30 WARN 13724 --- [Quiz-App] [nio-8080-exec-4] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method
2025-08-08T19:04:21.021+05:30 WARN 13724 --- [Quiz-App] [nio-8080-exec-5] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.http.converter.HttpMessageNotReadableException: Required r
2025-08-08T19:04:30.207+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-6] c.w.s.q.controller.QuestionController : putmapping is done
Hibernate:
select
q1_0.id,
q1_0.category,
q1_0.correct_answer,
q1_0.difficulty_level,
```

```

where
id=?
2025-08-08T19:04:53.241+05:30 WARN 13724 --- [Quiz-App] [io-8080-exec-10] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpReque
2025-08-08T19:05:26.512+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-9] c.w.s.q.controller.QuestionController : putmapping is done
Hibernate:
select
q1_0.id,
q1_0.category,
q1_0.correct_answer,
q1_0.difficulty_level,
q1_0.optiona,
q1_0.optionb,
q1_0.optionc,
q1_0.optiond,
q1_0.question_title
from
question q1_0
where
q1_0.id=?
2025-08-08T19:07:40.724+05:30 INFO 13724 --- [Quiz-App] [nio-8080-exec-7] c.w.s.q.controller.QuestionController : get method done successfully
Hibernate:
select
q1_0.id,
q1_0.category,
q1_0.correct_answer,
q1_0.difficulty_level,
q1_0.optiona,
q1_0.optionb,
q1_0.optionc,
q1_0.optiond,
q1_0.question_title
from
question q1_0
Hibernate:
select
q1_0.id,
q1_0.category,
q1_0.correct_answer,
q1_0.difficulty_level,
q1_0.optiona,
q1_0.optionb,
q1_0.optionc,
q1_0.optiond,
```