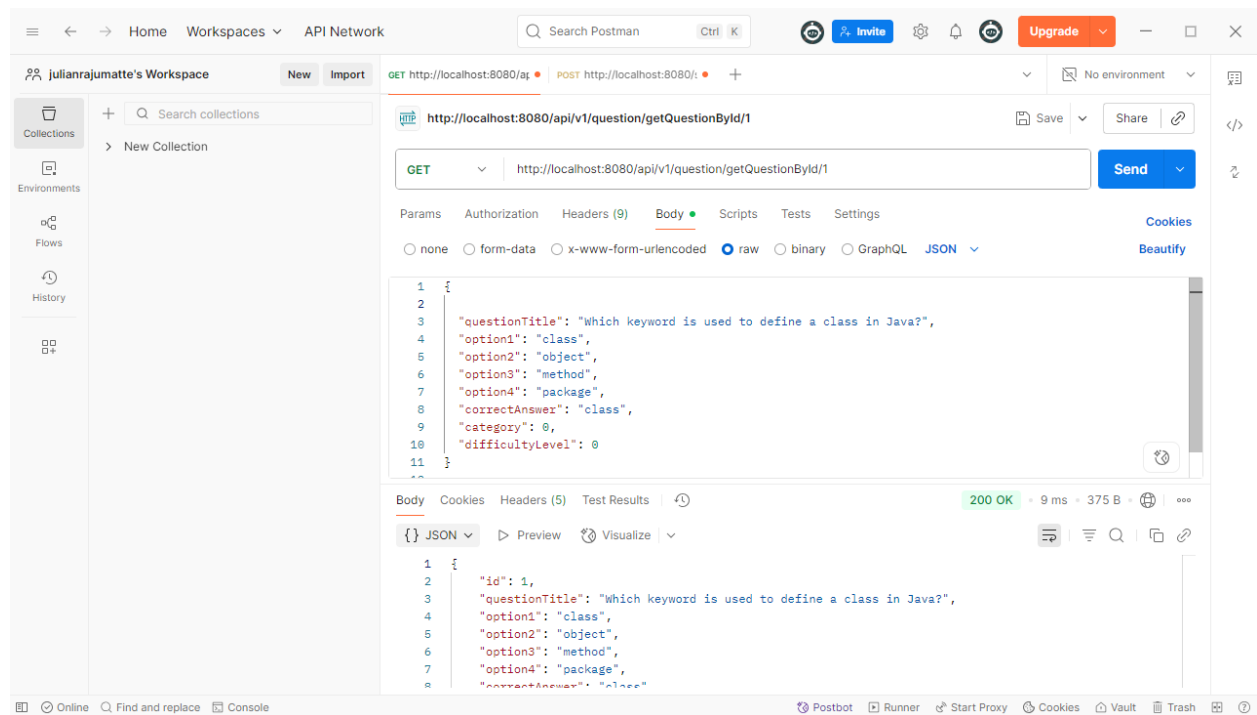


## GET:



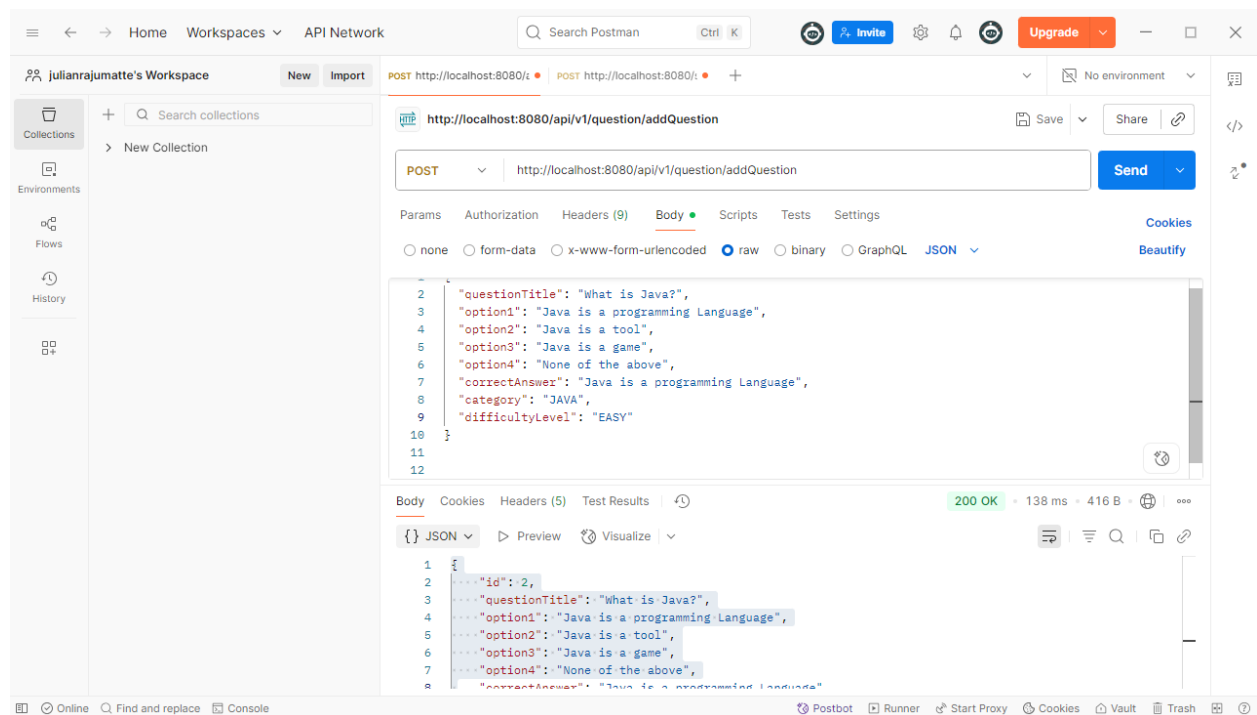
A screenshot of the Postman API client interface showing a GET request. The URL bar contains `http://localhost:8080/api/v1/question/getQuestionById/1`. The request method is set to GET. The body tab is selected, displaying a JSON object with the following structure:

```
1 {
2   "questionTitle": "Which keyword is used to define a class in Java?",
3   "option1": "class",
4   "option2": "object",
5   "option3": "method",
6   "option4": "package",
7   "correctAnswer": "class",
8   "category": 0,
9   "difficultyLevel": 0
10 }
11
```

The response tab is also selected, showing a 200 OK status with a response time of 9 ms and a body size of 375 B. The response body is a JSON object:

```
1 {
2   "id": 1,
3   "questionTitle": "Which keyword is used to define a class in Java?",
4   "option1": "class",
5   "option2": "object",
6   "option3": "method",
7   "option4": "package",
8   "correctAnswer": "class"
9 }
```

## Post:



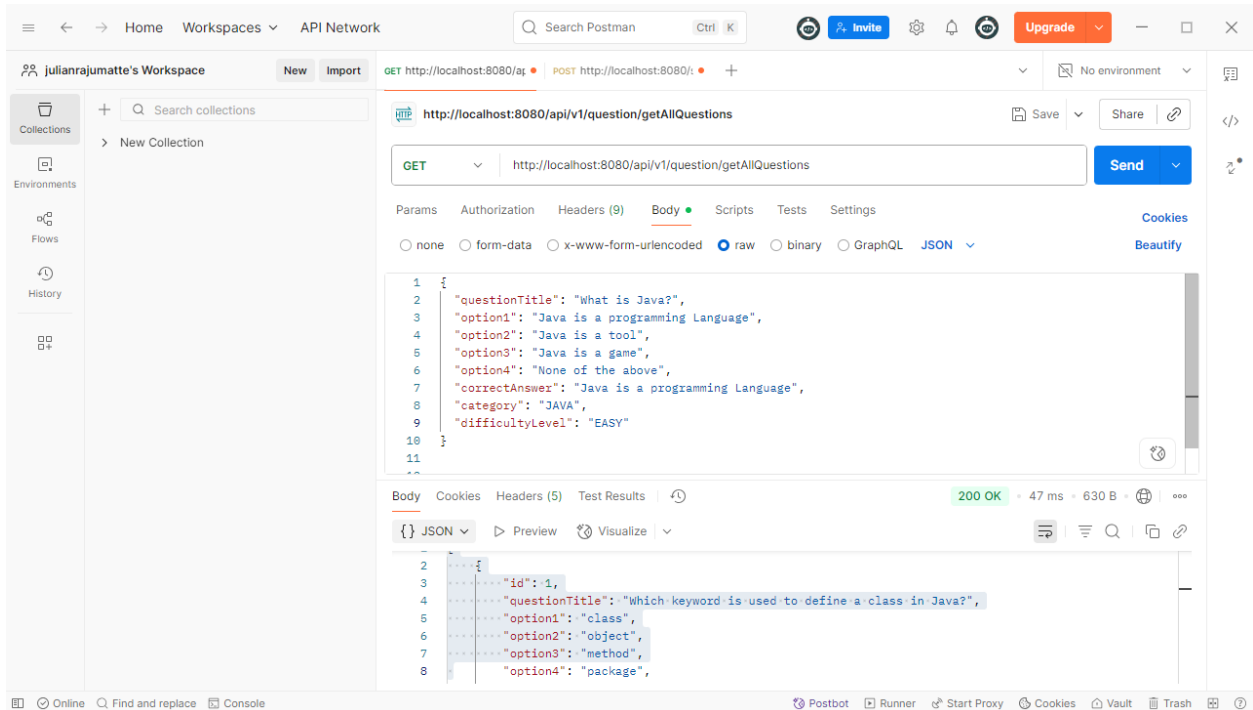
A screenshot of the Postman API client interface showing a POST request. The URL bar contains `http://localhost:8080/api/v1/question/addQuestion`. The request method is set to POST. The body tab is selected, displaying a JSON object with the following structure:

```
1 {
2   "questionTitle": "What is Java?",
3   "option1": "Java is a programming Language",
4   "option2": "Java is a tool",
5   "option3": "Java is a game",
6   "option4": "None of the above",
7   "correctAnswer": "Java is a programming Language",
8   "category": "JAVA",
9   "difficultyLevel": "EASY"
10 }
11
```

The response tab is also selected, showing a 200 OK status with a response time of 138 ms and a body size of 416 B. The response body is a JSON object:

```
1 {
2   "id": 2,
3   "questionTitle": "What is Java?",
4   "option1": "Java is a programming Language",
5   "option2": "Java is a tool",
6   "option3": "Java is a game",
7   "option4": "None of the above",
8   "correctAnswer": "Java is a programming Language"
9 }
```

## GetAllQuestions:

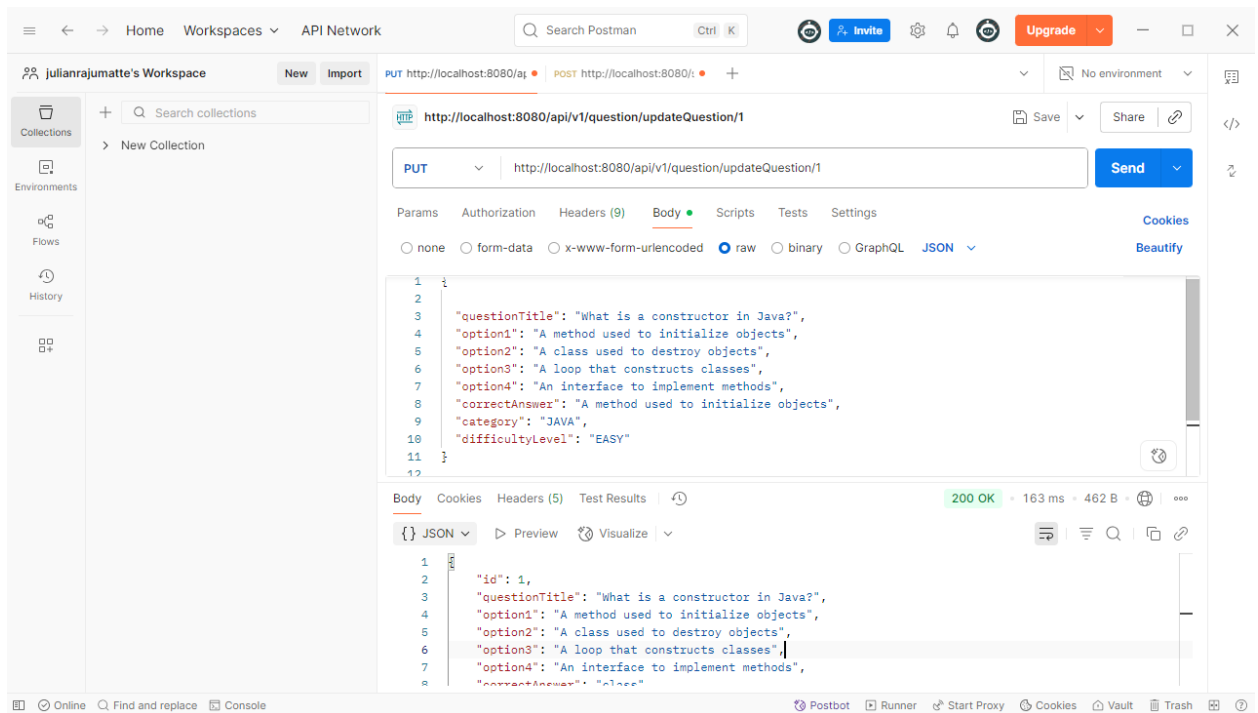


The screenshot shows the Postman interface for a workspace named "julianrajumatte's Workspace". The selected request is a GET request to `http://localhost:8080/api/v1/question/getAllQuestions`. The response is a 200 OK status with a response time of 47 ms and a body size of 630 B. The response body is a JSON array containing one question object.

```
1 {
2   "questionTitle": "What is Java?",
3   "option1": "Java is a programming Language",
4   "option2": "Java is a tool",
5   "option3": "Java is a game",
6   "option4": "None of the above",
7   "correctAnswer": "Java is a programming Language",
8   "category": "JAVA",
9   "difficultyLevel": "EASY"
10 }
```

```
1 {
2   "id": 1,
3   "questionTitle": "Which keyword is used to define a class in Java?",
4   "option1": "class",
5   "option2": "object",
6   "option3": "method",
7   "option4": "package",
8 }
```

## UpdateQuestion:

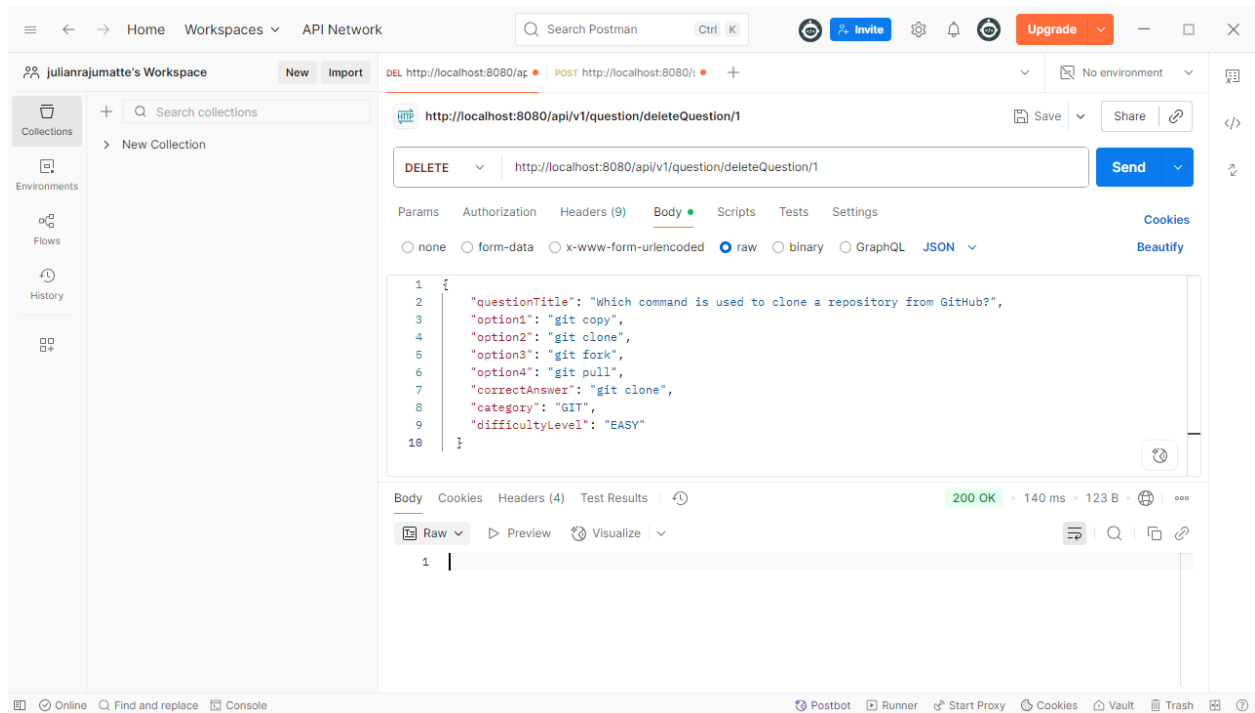


The screenshot shows the Postman interface for a workspace named "julianrajumatte's Workspace". The selected request is a PUT request to `http://localhost:8080/api/v1/question/updateQuestion/1`. The response is a 200 OK status with a response time of 163 ms and a body size of 462 B. The response body is a JSON object representing the updated question.

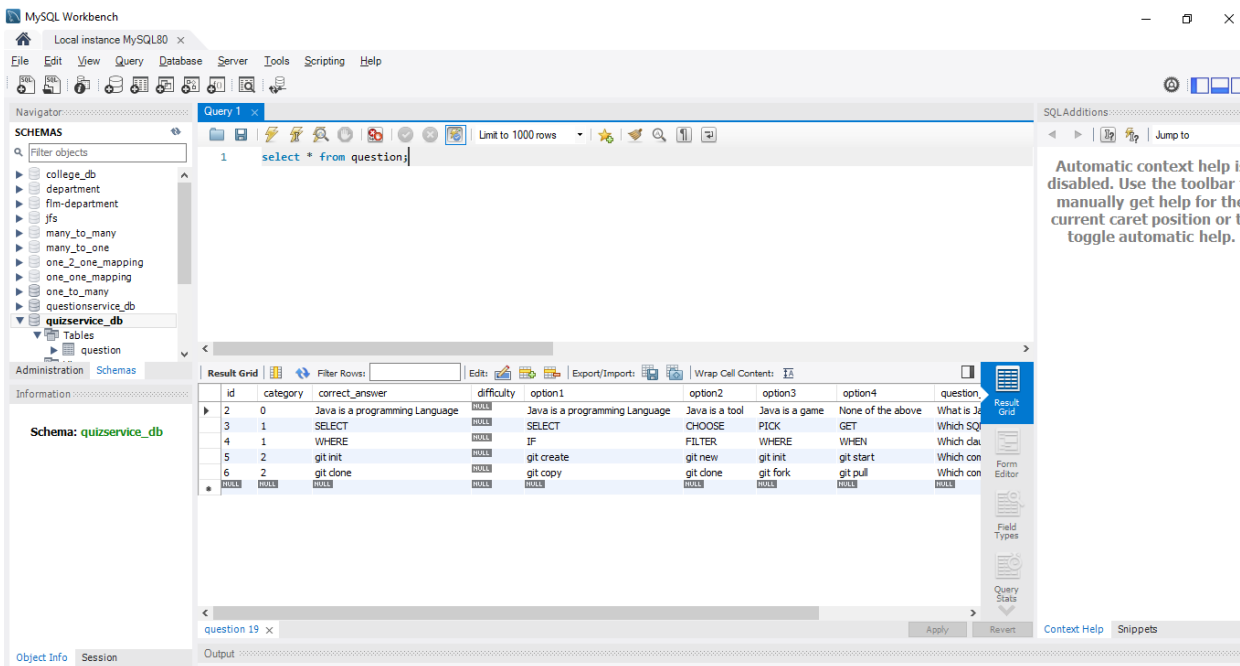
```
1 {
2   "questionTitle": "What is a constructor in Java?",
3   "option1": "A method used to initialize objects",
4   "option2": "A class used to destroy objects",
5   "option3": "A loop that constructs classes",
6   "option4": "An interface to implement methods",
7   "correctAnswer": "A method used to initialize objects",
8   "category": "JAVA",
9   "difficultyLevel": "EASY"
10 }
```

```
1 {
2   "id": 1,
3   "questionTitle": "What is a constructor in Java?",
4   "option1": "A method used to initialize objects",
5   "option2": "A class used to destroy objects",
6   "option3": "A loop that constructs classes",
7   "option4": "An interface to implement methods",
8   "correctAnswer": "A method used to initialize objects",
9   "category": "JAVA",
10  "difficultyLevel": "EASY"
11 }
```

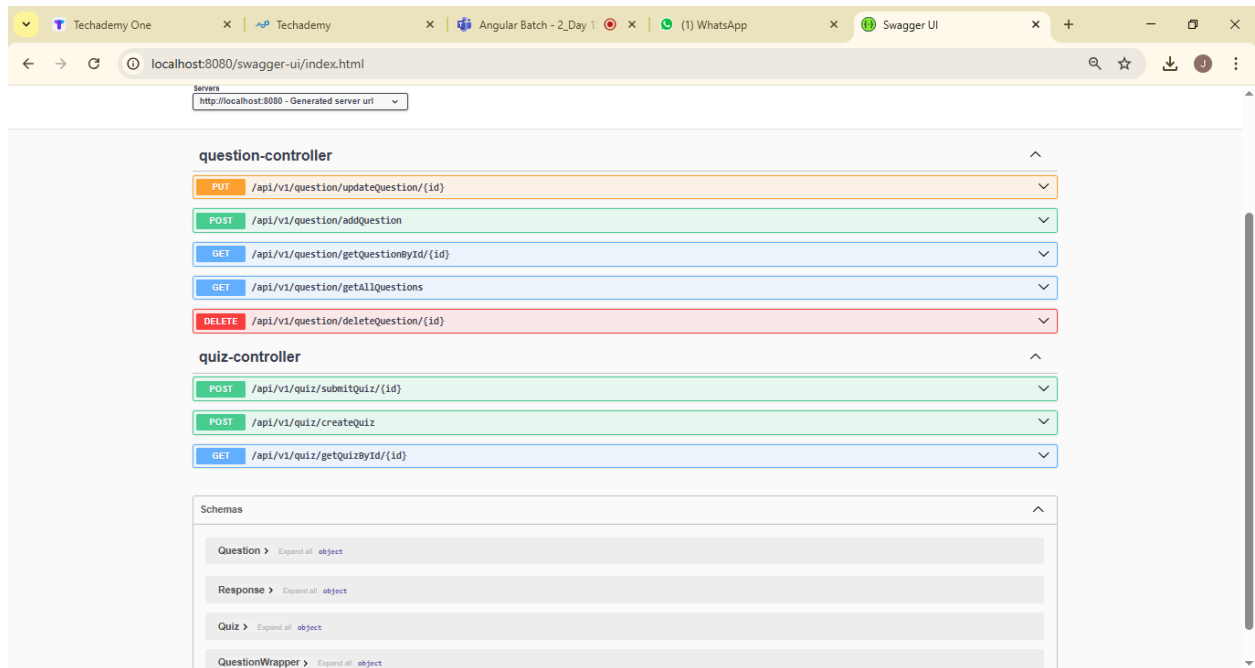
## Delete Question:



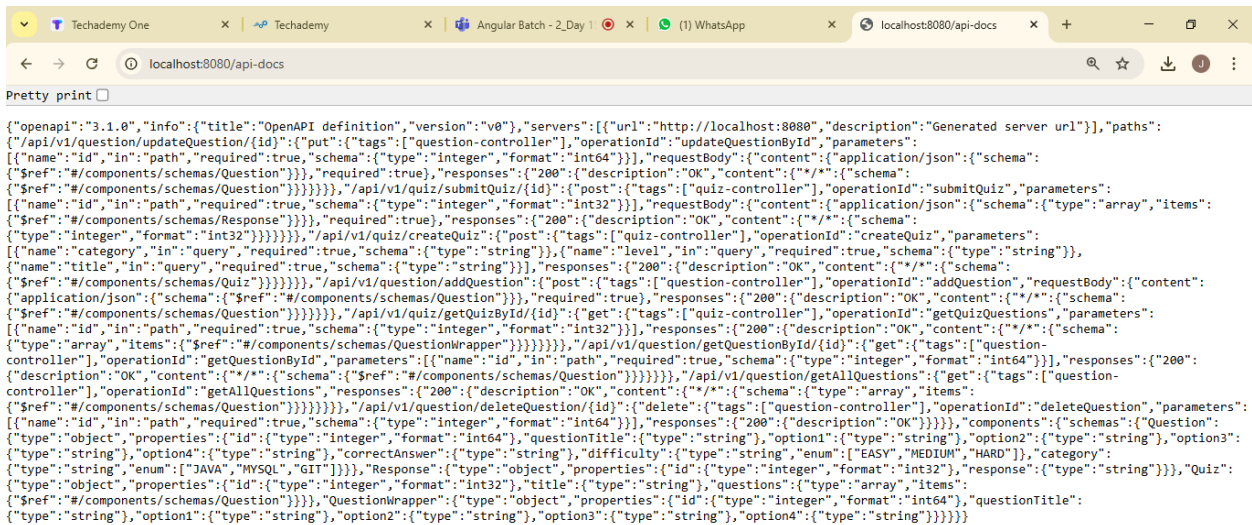
## MySQL Database:



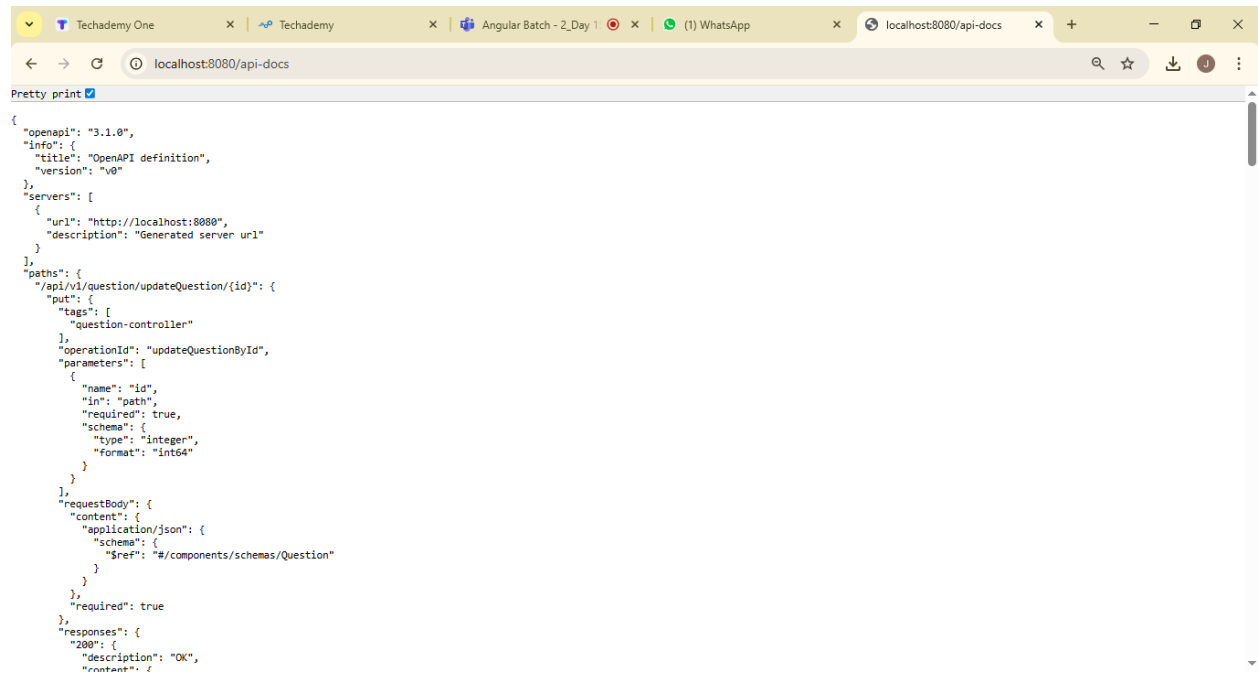
## Swagger:



## Api-Docs:



## Pretty Print:



The screenshot shows a web browser with multiple tabs. The active tab is titled 'localhost:8080/api-docs'. The address bar shows the URL 'localhost:8080/api-docs'. Below the address bar, there is a 'Pretty print' button. The main content area displays a JSON-formatted OpenAPI definition. The definition includes an 'openapi' version of '3.1.0', an 'info' object with title 'OpenAPI definition' and version 'v0', a 'servers' array with a single server at 'http://localhost:8080', and a 'paths' object. The 'paths' object contains a single path '/api/v1/question/updateQuestion/{id}' with a 'put' method. The method details include a 'tags' array with 'question-controller', an 'operationId' of 'updateQuestionById', a 'parameters' array with a path parameter 'id' of type 'integer' and format 'int64', a 'requestBody' object with a JSON schema that references '#/components/schemas/Question', and a 'responses' object with a '200' status code and a description 'OK'.

```
{
  "openapi": "3.1.0",
  "info": {
    "title": "OpenAPI definition",
    "version": "v0"
  },
  "servers": [
    {
      "url": "http://localhost:8080",
      "description": "Generated server url"
    }
  ],
  "paths": {
    "/api/v1/question/updateQuestion/{id}": {
      "put": {
        "tags": [
          "question-controller"
        ],
        "operationId": "updateQuestionById",
        "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": {}
          }
        }
      }
    }
  }
}
```