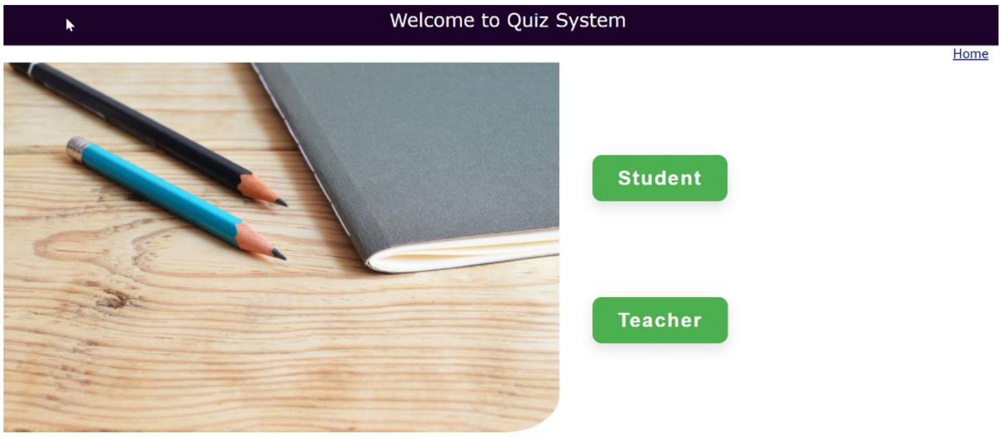
# Advanced Java Report

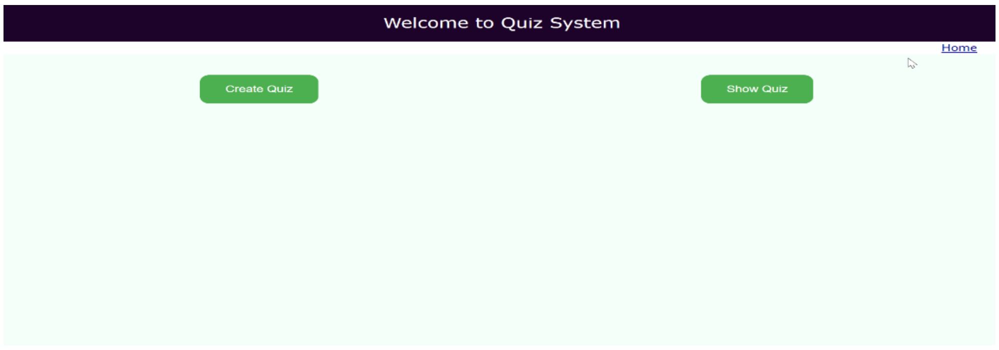
Submitted By: Naresh Kumar Jaiswal
Submitted To: Thomas Broussard Sir

## Starting from Frontend,

**User is welcome by the login page or Option page for Student or Teacher.**

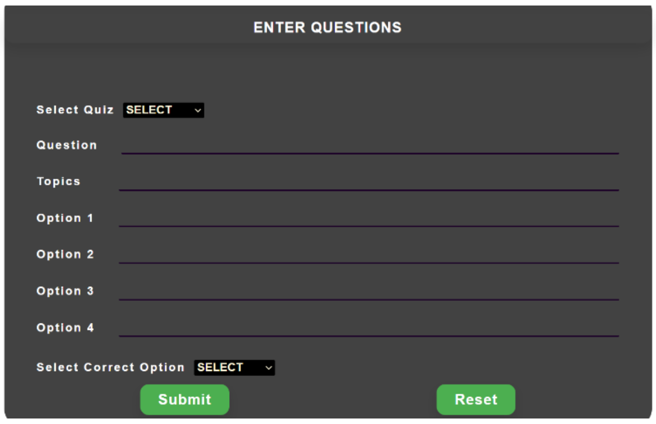


**If a user chose to be Teacher, he will be redirected to another page.**

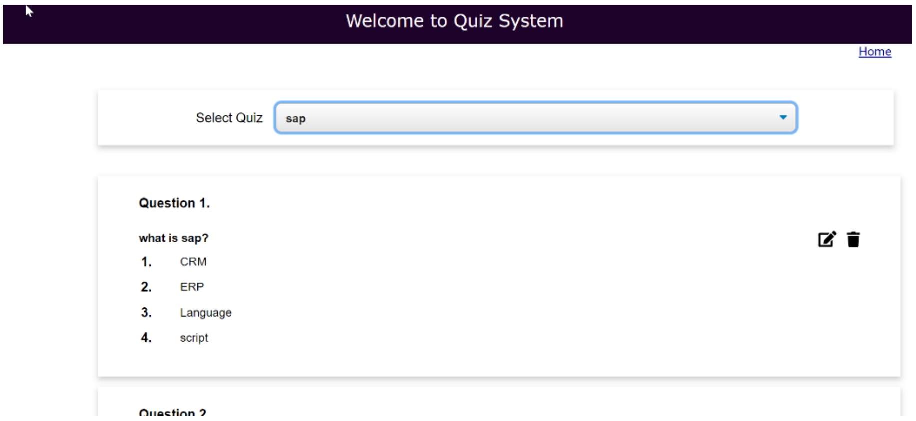


**He can choose create question/Quiz or show created quiz. Lets cho**

**Lets Choose Create quiz/question. Here Teacher can Add questions**



.



Welcome to Quiz System

Home

Select Quiz sap

Question 1.

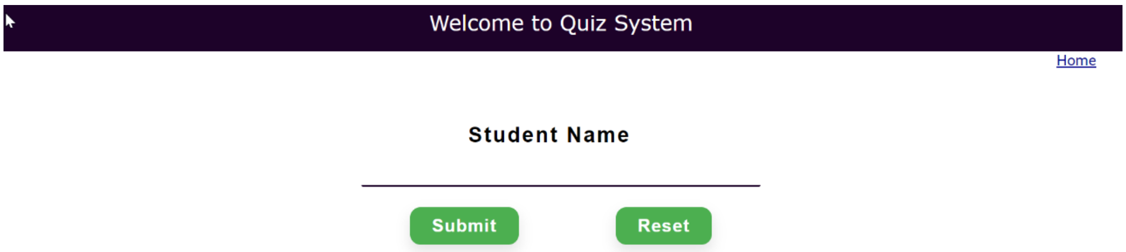
what is sap?

1. CRM
2. ERP
3. Language
4. script

Question 2

**Here Teacher can edit or delete the created question.**

**Now Student can start the quiz.**

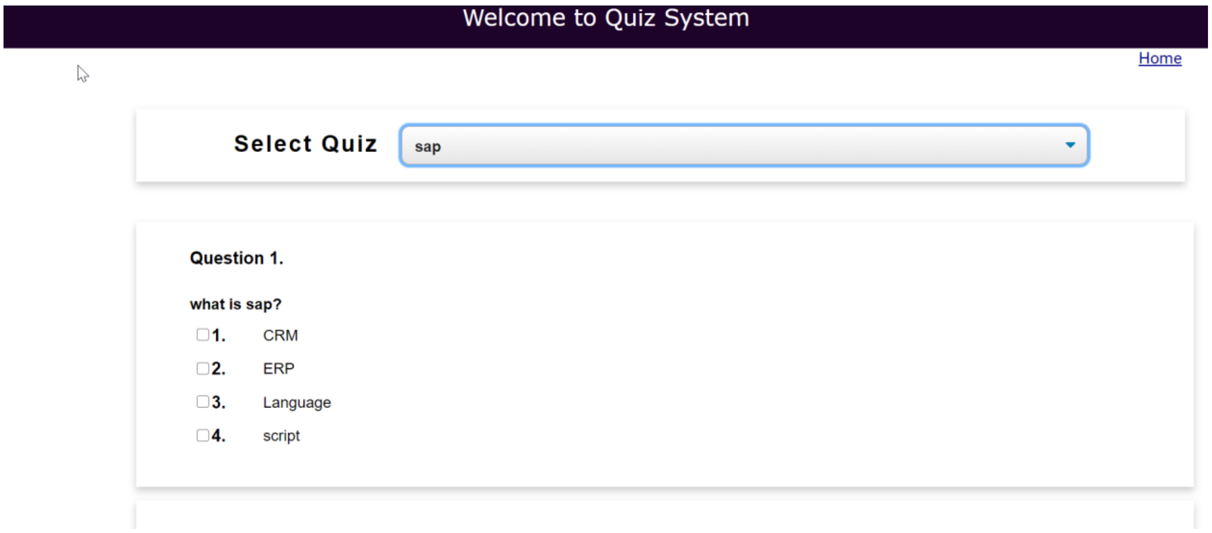


Welcome to Quiz System

Home

Student Name

**Sample of Quiz Exam questions**



Welcome to Quiz System

Home

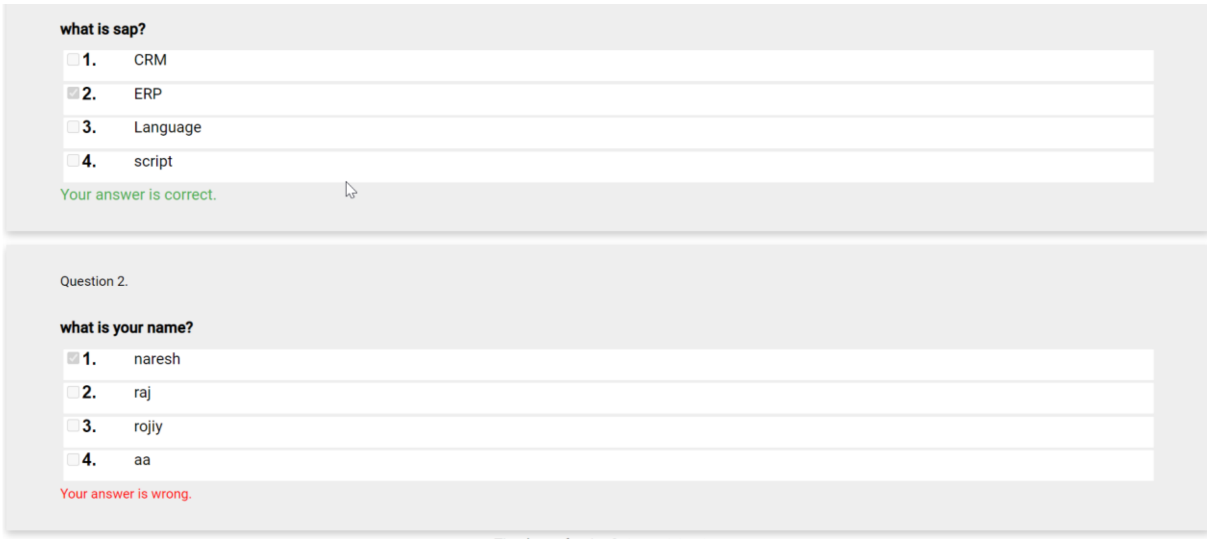
Select Quiz sap

Question 1.

what is sap?

☐1. CRM
☐2. ERP
☐3. Language
☐4. script

**After submitting student got the Result.**



**what is sap?**

1. CRM
[✓ 2.] ERP
3. Language
4. script

Your answer is correct.

Question 2.

**what is your name?**

[✓ 1.] naresh
2. raj
3. rojiy
4. aa

Your answer is wrong.

## Now Backend,

I have used JAVA , SPRING , JPA and H2 for database.
API, s used are:

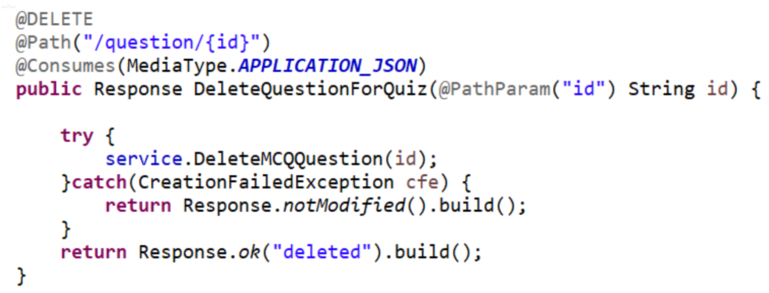
### 1. Creating questions.

```java
@PUT
@Path("/{id}/questions")
@Consumes(MediaType.APPLICATION_JSON)
public Response updateQuestionForQuiz(@PathParam("id") String id, QuestionDTO dto) {
    if (dto == null) {
        return Response.status(400, "no question was provided").entity("bad request").build();
    }

    try {
        service.UpdateMCQQuestion(dto);
    }catch(CreationFailedException cfe) {
        return Response.notModified().build();
    }

    return Response.created(URI.create(String.valueOf("quiz/"+id+"/question/"+ dto.getQuestionId()))).build()
}
```

### 2. Updating question.

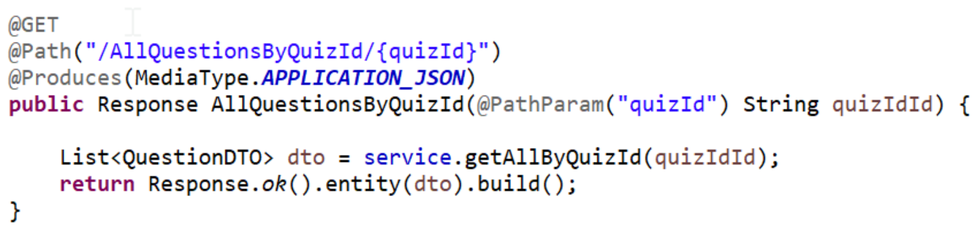
```java
@PUT
@Path("/{id}/questions")
@Consumes(MediaType.APPLICATION_JSON)
public Response updateQuestionForQuiz(@PathParam("id") String id, QuestionDTO dto) {
    if (dto == null) {
        return Response.status(400, "no question was provided").entity("bad request").build();
    }

    try {
        service.UpdateMCQQuestion(dto);
    }catch(CreationFailedException cfe) {
        return Response.notModified().build();
    }

    return Response.created(URI.create(String.valueOf("quiz/"+id+"/question/"+ dto.getQuestionId()))).build()
}
```

## 3. Deleting question



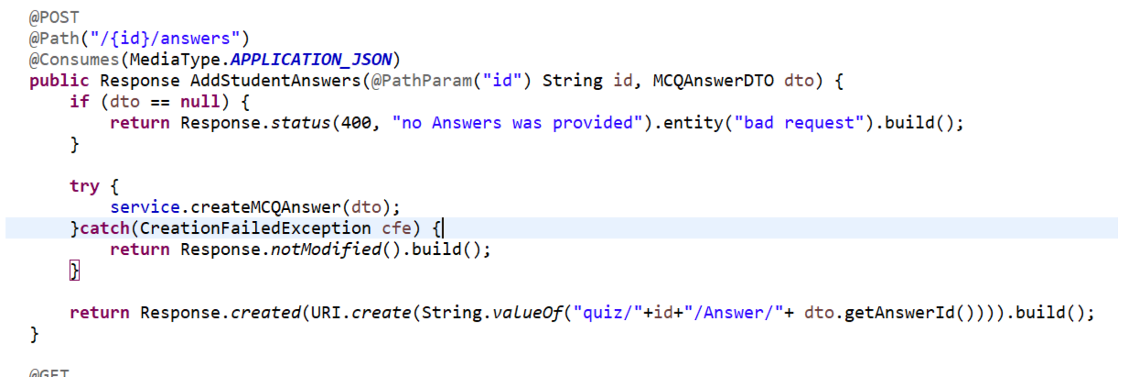
```
@DELETE
@Path("/question/{id}")
@Consumes(MediaType.APPLICATION_JSON)
public Response DeleteQuestionForQuiz(@PathParam("id") String id) {

    try {
        service.DeleteMCQQuestion(id);
    }catch(CreationFailedException cfe) {
        return Response.notModified().build();
    }
    return Response.ok("deleted").build();
}
```

## 4. Get Question List according to Quiz id



```
@GET
@Path("/AllQuestionsByQuizId/{quizId}")
@Produces(MediaType.APPLICATION_JSON)
public Response AllQuestionsByQuizId(@PathParam("quizId") String quizIdId) {

    List<QuestionDTO> dto = service.getAllByQuizId(quizIdId);
    return Response.ok().entity(dto).build();
}
```

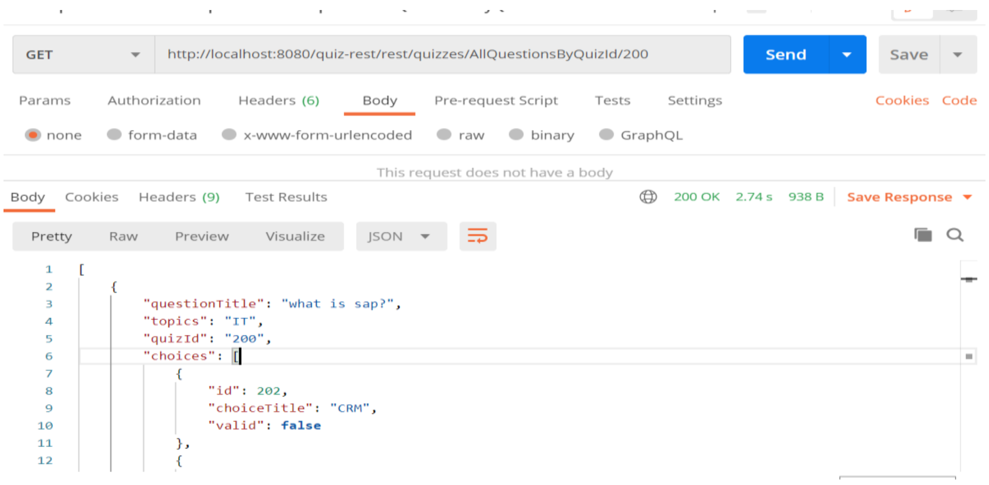
## 5. Save Student Quiz answers



```
@POST
@Path("/{id}/answers")
@Consumes(MediaType.APPLICATION_JSON)
public Response AddStudentAnswers(@PathParam("id") String id, MCQAnswerDTO dto) {
    if (dto == null) {
        return Response.status(400, "no Answers was provided").entity("bad request").build();
    }

    try {
        service.createMCQAnswer(dto);
    }catch(CreationFailedException cfe) {
        return Response.notModified().build();
    }

    return Response.created(URI.create(String.valueOf("quiz/"+id+"/Answer/"+ dto.getAnswerId()))).build();
}

@GET
```

## Now PostMan Api Request and Respose.

1. Get all Questions By QuizId



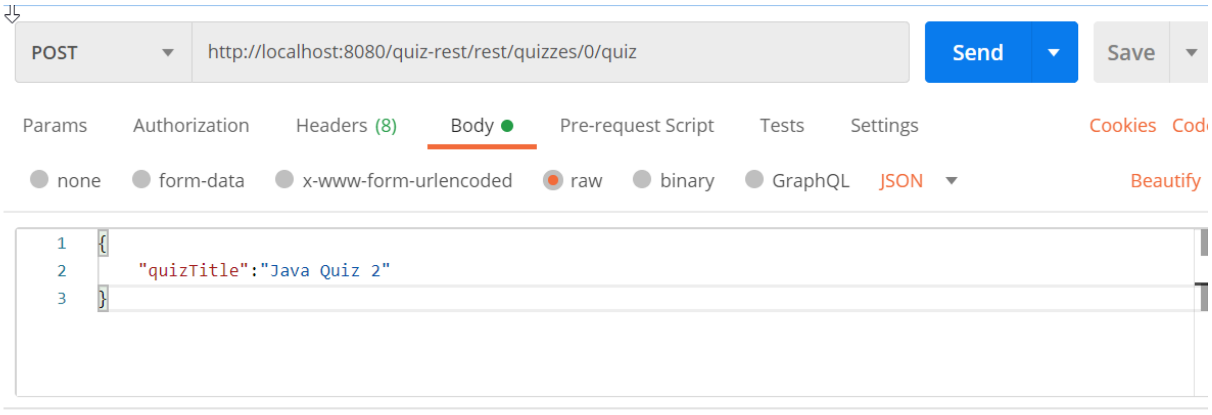
GET http://localhost:8080/quiz-rest/rest/quizzes/AllQuestionsByQuizId/200

Body: none — This request does not have a body

200 OK 2.74 s 938 B

```
[
    {
        "questionTitle": "what is sap?",
        "topics": "IT",
        "quizId": "200",
        "choices": [
            {
                "id": 202,
                "choiceTitle": "CRM",
                "valid": false
            },
            {
```

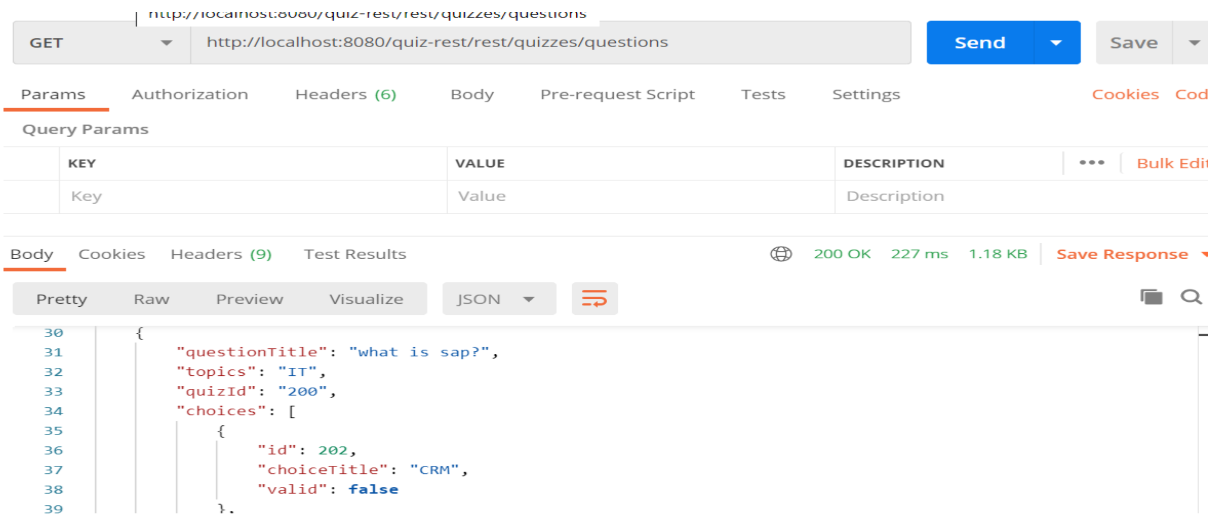
2. Create Quiz.



POST http://localhost:8080/quiz-rest/rest/quizzes/0/quiz

```
{
    "quizTitle":"Java Quiz 2"
}
```

3. Get All Questions



GET http://localhost:8080/quiz-rest/rest/quizzes/questions

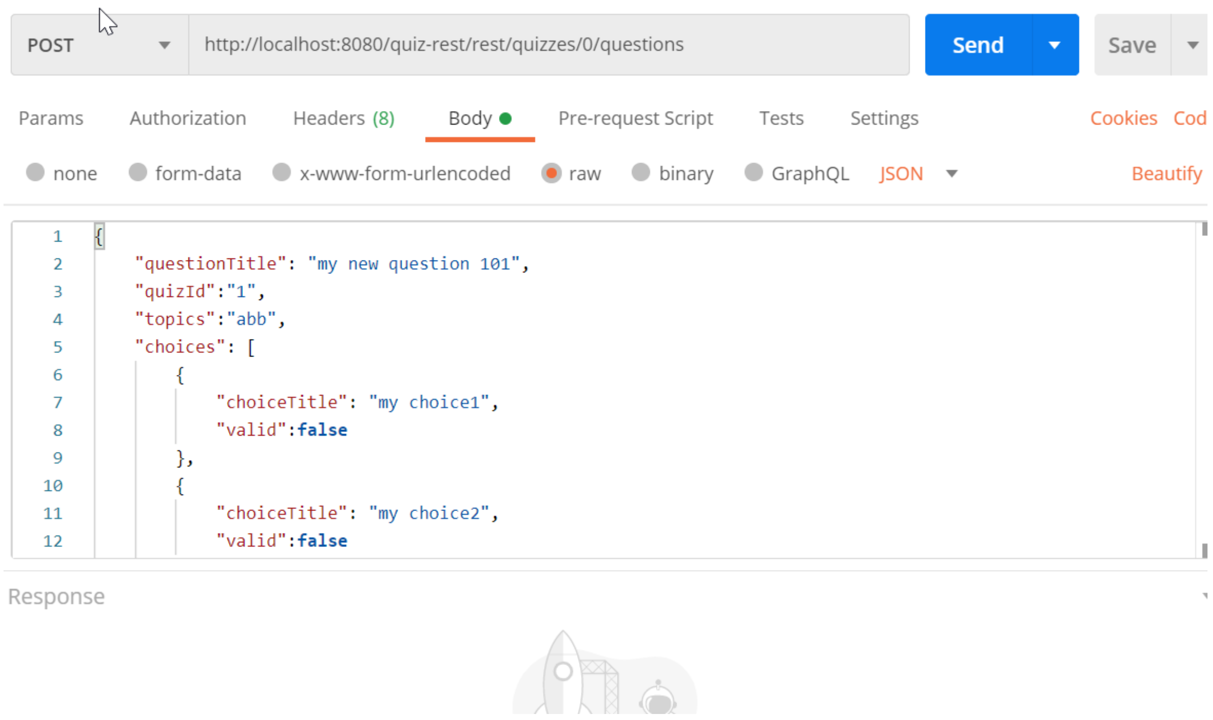
Query Params

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

200 OK 227 ms 1.18 KB

```
    {
        "questionTitle": "what is sap?",
        "topics": "IT",
        "quizId": "200",
        "choices": [
            {
                "id": 202,
                "choiceTitle": "CRM",
                "valid": false
            },
```

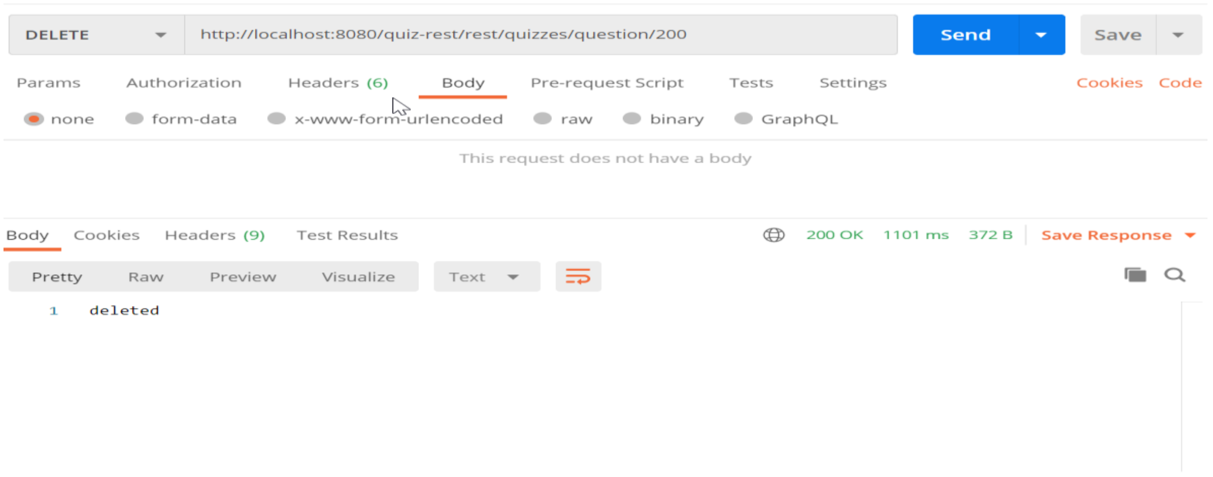
4. Create Questions.



http://localhost:8080/quiz-rest/rest/quizzes/0/questions

```
{
    "questionTitle": "my new question 101",
    "quizId":"1",
    "topics":"abb",
    "choices": [
        {
            "choiceTitle": "my choice1",
            "valid":false
        },
        {
            "choiceTitle": "my choice2",
            "valid":false
```

5. Delete Questions.



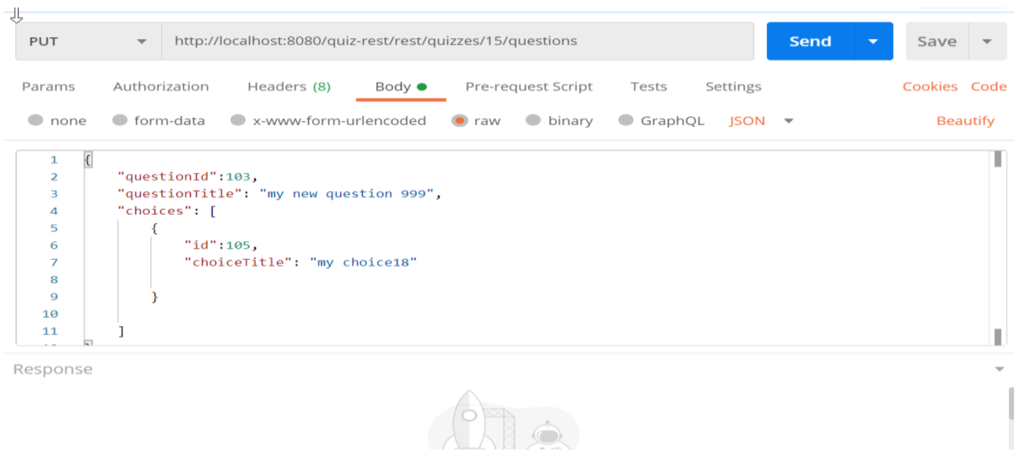
http://localhost:8080/quiz-rest/rest/quizzes/question/200

This request does not have a body

200 OK 1101 ms 372 B

```
deleted
```

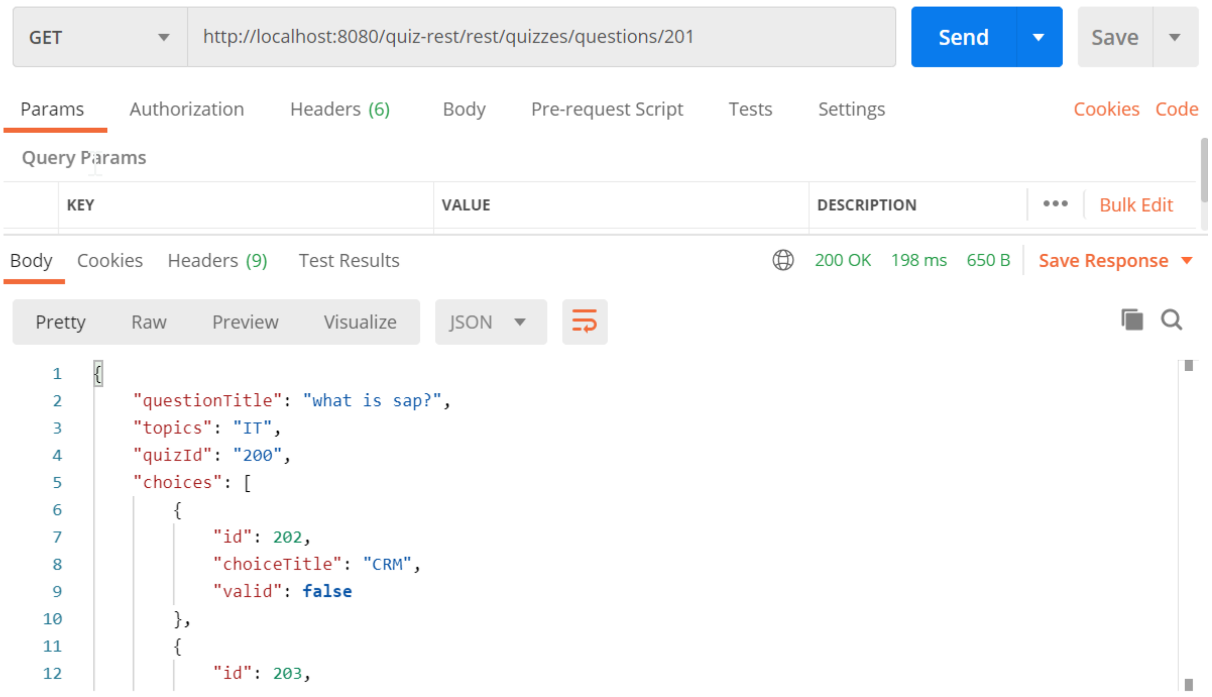
6. Update Questions.



http://localhost:8080/quiz-rest/rest/quizzes/15/questions

```
{
    "questionId":103,
    "questionTitle": "my new question 999",
    "choices": [
        {
            "id":105,
            "choiceTitle": "my choice18"

        }
    ]
```

7. Get Question By ID

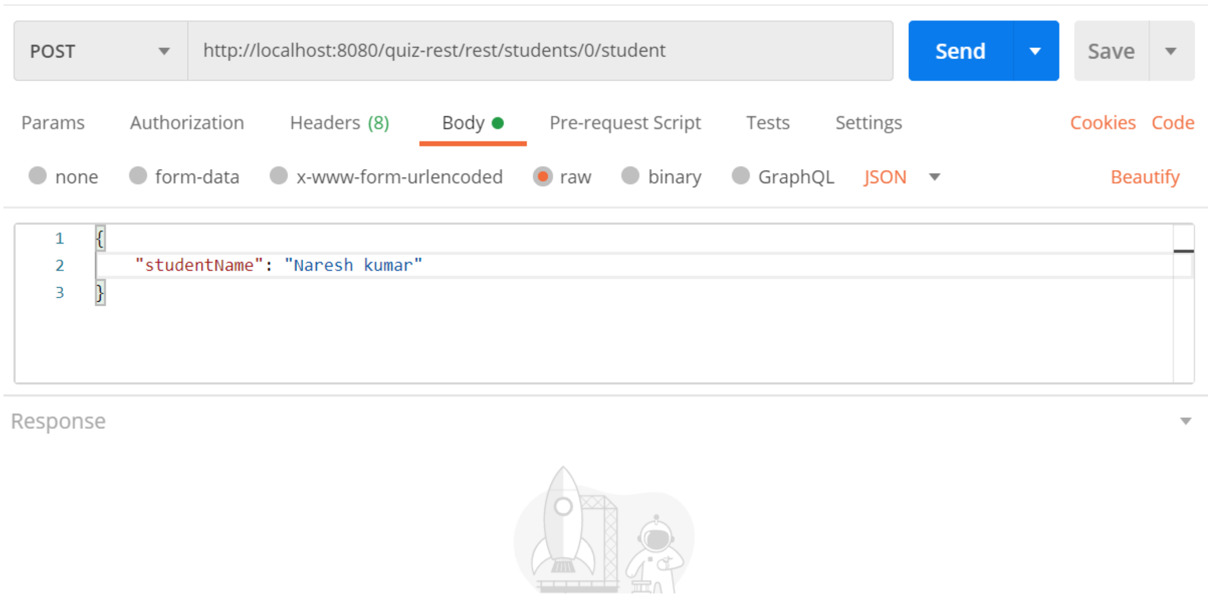


GET http://localhost:8080/quiz-rest/rest/quizzes/questions/201

200 OK 198 ms 650 B

```
{
    "questionTitle": "what is sap?",
    "topics": "IT",
    "quizId": "200",
    "choices": [
        {
            "id": 202,
            "choiceTitle": "CRM",
            "valid": false
        },
        {
            "id": 203,
```

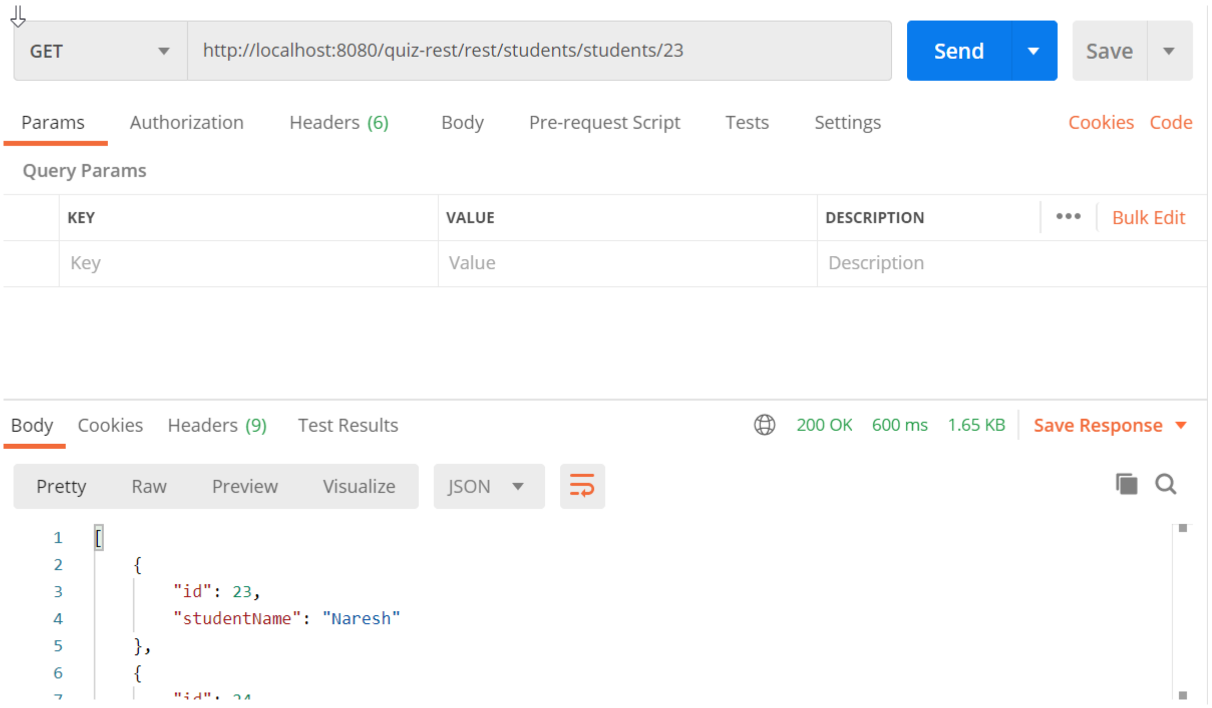
8. Add Student



POST http://localhost:8080/quiz-rest/rest/students/0/student

```
{
    "studentName": "Naresh kumar"
}
```

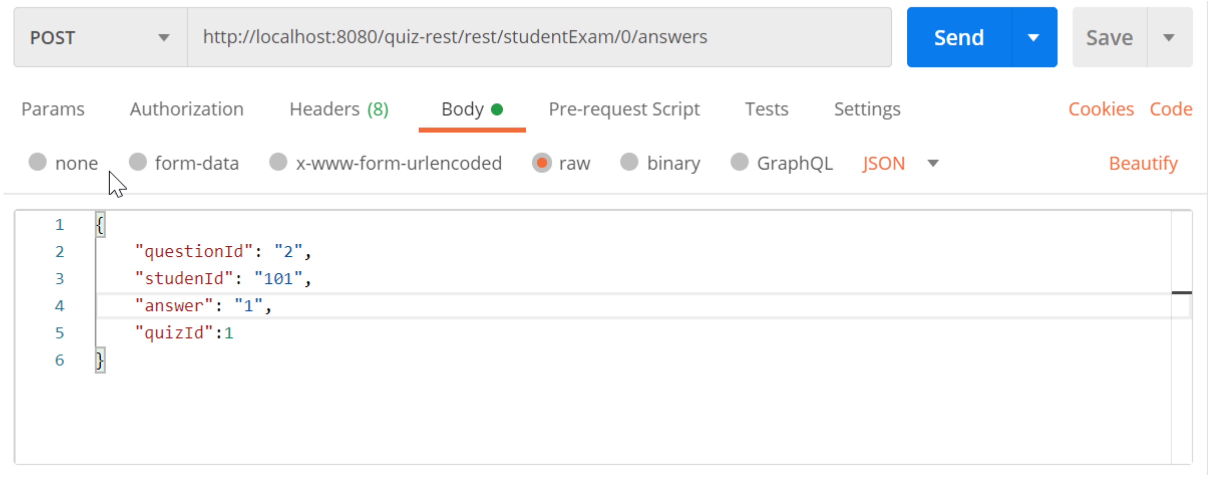
9. Get Student.



http://localhost:8080/quiz-rest/rest/students/students/23

```
[
    {
        "id": 23,
        "studentName": "Naresh"
    },
    {
        "id": 24
```

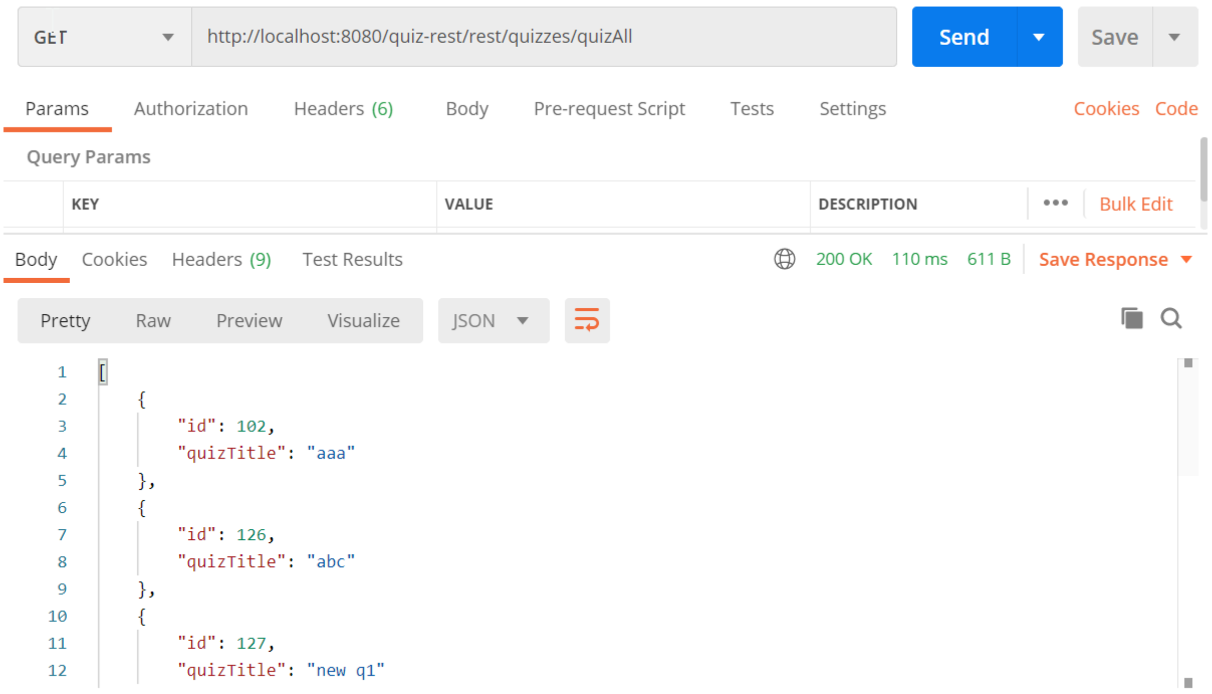
10. Save Student Exam Answers.



http://localhost:8080/quiz-rest/rest/studentExam/0/answers

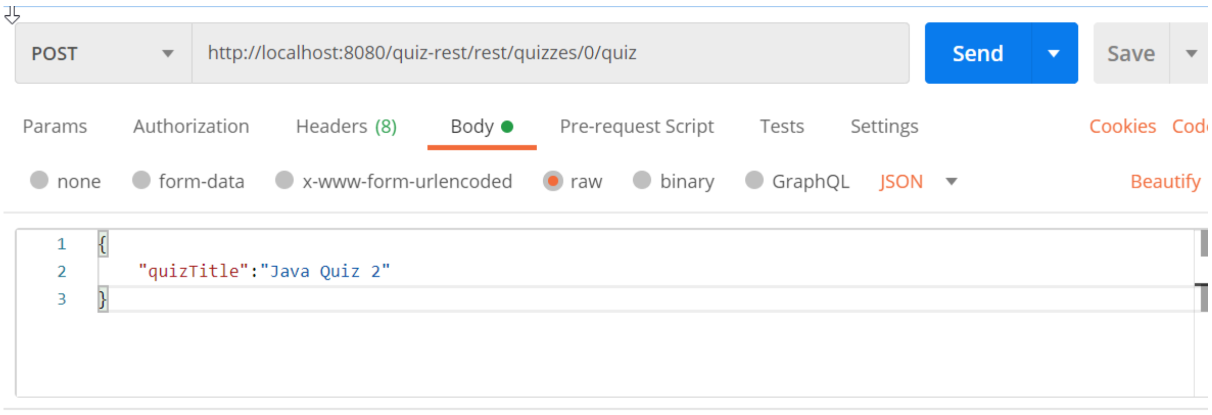
```
{
    "questionId": "2",
    "studenId": "101",
    "answer": "1",
    "quizId":1
}
```

11. Get AllQuiz


GET http://localhost:8080/quiz-rest/rest/quizzes/quizAll

```
[
    {
        "id": 102,
        "quizTitle": "aaa"
    },
    {
        "id": 126,
        "quizTitle": "abc"
    },
    {
        "id": 127,
        "quizTitle": "new q1"
```

12. Add Quiz.


POST http://localhost:8080/quiz-rest/rest/quizzes/0/quiz

```
{
    "quizTitle":"Java Quiz 2"
}
```

THANK YOU