

# Online Quiz Portal Using REST APIs

## 1. User Register

Postman interface showing a successful POST request to `http://localhost:8136/quizapp/userRegister`. The request body is a JSON object with the following fields:

```
1 {
2   "emailid": "userrev@gmail.com",
3   "password": "userrev123",
4   "phno": "8989895656"
5 }
```

The response is a single word: `registered`. The status is 200 OK, Time: 757 ms, Size: 166 B.

Postman interface showing a successful POST request to `http://localhost:8136/quizapp/userRegister`. The request body is a JSON object with the following fields:

```
1 {
2   "emailid": "revanth@gmail.com",
3   "password": "revanth123",
4   "phno": "8989892323"
5 }
```

The response is a single word: `registered`. The status is 200 OK, Time: 32 ms, Size: 166 B.

## 2. User Login

Postman interface showing a successful login request.

Request URL: `http://localhost:8136/quizapp/userLogin`

Method: **POST**

Body (JSON):

```
1 {
2   "emailid": "revanth@gmail.com",
3   "password": "revanth123"
4 }
```

Status: 200 OK, Time: 33 ms, Size: 172 B

Response (JSON):

```
1 login successfull
```

Postman interface showing an unsuccessful login request.

Request URL: `http://localhost:8136/quizapp/userLogin`

Method: **POST**

Body (JSON):

```
1 {
2   "emailid": "user@gmail.com",
3   "password": "revanth123"
4 }
```

Status: 200 OK, Time: 14 ms, Size: 170 B

Response (JSON):

```
1 User not found
```

### 3. Admin Login

Postman interface showing a successful POST request to `http://localhost:8136/quizapp/adminLogin`. The request body is a JSON object:

```
1 {
2   "username": "admin",
3   "password": "admin123"
4 }
```

The response status is 200 OK, Time: 16 ms, Size: 169 B. The response body is:

```
1 Welcome admin
```

Postman interface showing a failed POST request to `http://localhost:8136/quizapp/adminLogin`. The request body is a JSON object:

```
1 {
2   "username": "adminRev",
3   "password": "admin123"
4 }
```

The response status is 200 OK, Time: 17 ms, Size: 175 B. The response body is:

```
1 invalid Credentials
```

## 4. Add Questions

Postman interface showing a POST request to `http://localhost:8136/quizapp/addQuestions`. The request is configured with the following body (JSON):

```
1 {
2   "quest": "Who is well known as father of Java ?",
3   "opt1": "Guido van Rossum",
4   "opt2": "James Gosling",
5   "opt3": "Dennis Ritchie",
6   "opt4": "Bjaine Stroustrup",
7   "ans": 2
8 }
```

The response status is 200 OK, Time: 62 ms, Size: 170 B. The response body is displayed as "1 question added".

Postman interface showing a POST request to `http://localhost:8136/quizapp/addQuestions`. The request is configured with the following body (JSON):

```
1 {
2   "quest": "AOP Stands for ?",
3   "opt1": "Aspect Oriented Programming",
4   "opt2": "Any Object Programming",
5   "opt3": "Asset Oriented Programming",
6   "opt4": "Asset Oriented Protocol",
7   "ans": 1
8 }
```

The response status is 200 OK, Time: 32 ms, Size: 170 B. The response body is displayed as "1 question added".

## 5. Add Quiz

Postman interface showing a successful POST request to `http://localhost:8136/quizapp/addQuiz`. The request body is a JSON object with the following structure:

```
1 {
2   "subject": "Java",
3   "title": "Java Basics",
4   "quizno": 2,
5   "qid": 1
6 }
```

The response body is `quiz added`. The status is 200 OK, Time: 175 ms, Size: 166 B.

Postman interface showing a successful POST request to `http://localhost:8136/quizapp/addQuiz`. The request body is a JSON object with the following structure:

```
1 {
2   "subject": "Spring Boot",
3   "title": "Spring Boot Basics",
4   "quizno": 1,
5   "qid": 3
6 }
```

The response body is `quiz added`. The status is 200 OK, Time: 30 ms, Size: 166 B.

## 6. Take Test

The screenshot shows the Postman interface for a REST client. The top bar includes an 'Explore' button, a search bar, and utility buttons like 'Invite', 'Upgrade', and 'No Environment'. The main workspace displays a POST request to `http://localhost:8136/quizapp/takeTest`. The request body is a JSON object with the following structure:

```
1 {
2   "userid": {
3     "uid": 1
4   },
5   "quizid": {
6     "quid": 4
7   },
8   "questionid": {
9     "qid": 3
10  },
11  "testans": "1"
12 }
```

The response status is 200 OK, with a time of 38 ms and a size of 164 B. The response body is displayed in the 'Body' tab, showing the text 'submitted'.

The screenshot shows the Postman interface for a REST client. The top bar includes an 'Explore' button, a search bar, and utility buttons like 'Invite', 'Upgrade', and 'No Environment'. The main workspace displays a POST request to `http://localhost:8136/quizapp/takeTest`. The request body is a JSON object with the following structure:

```
1 {
2   "userid": {
3     "uid": 2
4   },
5   "quizid": {
6     "quid": 3
7   },
8   "questionid": {
9     "qid": 1
10  },
11  "testans": "3"
12 }
```

The response status is 200 OK, with a time of 26 ms and a size of 164 B. The response body is displayed in the 'Body' tab, showing the text 'submitted'.

## 7. Get Result

Explore Search Postman Invite Upgrade

GET http://localhost:8136/q + ... No Environment

http://localhost:8136/quizapp/getresult Save

GET http://localhost:8136/quizapp/getresult Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 67 ms Size: 245 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "email": "userrev@gmail.com",
4     "marks": 2
5   },
6   {
7     "email": "revanth@gmail.com",
8     "marks": 1
9   }
10 ]
```

## 8. Get Admin Result

Explore Search Postman Invite Upgrade

GET http://localhost:8136/q + ... No Environment

http://localhost:8136/quizapp/getAdminResult Save

GET http://localhost:8136/quizapp/getAdminResult Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 43 ms Size: 245 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "email": "userrev@gmail.com",
4     "marks": 2
5   },
6   {
7     "email": "revanth@gmail.com",
8     "marks": 1
9   }
10 ]
```

## 9. Quiz Info

Explore

Search Postman

Invite

Upgrade

GET http://localhost:8136/q

No Environment

Save

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Body

Cookies

Headers (5)

Test Results

Status: 200 OK Time: 656 ms Size: 241 B Save as Example

Pretty Raw Preview Visualize JSON

Activate Windows  
Go to Settings to activate Windows.



## 10. View Quiz

The screenshot shows the Postman interface with a GET request to `http://localhost:8136/quizapp/viewQuiz`. The request is successful, returning a 200 OK status with a response time of 14 ms and a size of 208 B. The response body is displayed in the 'Body' tab, showing a JSON array of two quiz items.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body** (Pretty)

```
[{"description": "Java Basics", "score": 1}, {"description": "Spring Boot Basics", "score": 1}]
```

## 11. Get All Test

Explore Search Postman Invite Upgrade

GET http://localhost:8136/q + ... No Environment

HTTP http://localhost:8136/quizapp/getAllTest Save Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 75 ms Size: 2.34 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "tid": 1,
3   "userid": {
4     "uid": 1,
5     "emailid": "userrev@gmail.com",
6     "password": "userrev123",
7     "phno": 8989895656
8   },
9 },
10  "quizid": {
11    "quid": 4,
12    "title": "Spring Boot Basics",
13    "quizno": 1,
14    "subject": "Spring Boot",
15    "qid": {
16      "qid": 3,
17      "quest": "AOP Stands for ?",
18      "opt1": "Aspect Oriented Programming",
19      "opt2": "Any Object Programming",
20      "opt3": "Asset Oriented Programming",
21      "opt4": "Asset Oriented Protocol",
22      "ans": 1
23    }
24  },
25  "questionid": {
26    "qid": 3,
27    "quest": "AOP Stands for ?",
```

Activate Windows  
Go to Settings to activate Windows.

Runner Capture requests Cookies Trash

## 12. View All Quizes

The image shows the Postman application interface. At the top, there's a search bar and navigation icons. The main area displays a GET request to `http://localhost:8136/quizapp/viewAllQuiz`. The response is a JSON array with two quiz objects. The first quiz is titled "Java Basics" and the second is titled "Spring Boot Basics".

```
1 [
2   {
3     "quid": 3,
4     "title": "Java Basics",
5     "quizno": 2,
6     "subject": "Java",
7     "qid": {
8       "qid": 1,
9       "quest": "Who is well known as father of Java ?",
10      "opt1": "Guido van Rossum",
11      "opt2": "James Gosling",
12      "opt3": "Dennis Ritchie",
13      "opt4": "Bjarne Stroustrup",
14      "ans": 2
15    }
16  },
17  {
18    "quid": 4,
19    "title": "Spring Boot Basics",
20    "quizno": 1,
21    "subject": "Spring Boot",
22    "qid": {
23      "qid": 3,
24      "quest": "AOP Stands for ?",
25      "opt1": "Aspect Oriented Programming",
26      "opt2": "Any Object Programming",
27      "opt3": "Asset Oriented Programming",
28    }
29  }
30 ]
```

## 13. Tables :

- Admin

The image shows a SQL query being executed in a database client. The query is:

```
1 • create database quizapp;
2 • use quizapp;
3 • select * from admin;
4
```

The result grid shows the following data:

	id	password	username
▶	1	admin123	admin
✱	NULL	NULL	NULL

- **User**

```
1 • create database quizapp;
2 • use quizapp;
3 • select * from user;
4
```

Result Grid				
	uid	emailid	password	phno
1	userrev@gmail.com	userrev123	8989895656	
2	revanth@gmail.com	revanth123	8989892323	
	NULL	NULL	NULL	NULL

- **Quiz**

```
1 • create database quizapp;
2 • use quizapp;
3 • select * from quiz;
4
```

	quid	quizno	subject	title	qid_qid
▶	3	2	Java	Java Basics	1
	4	1	Spring Boot	Spring Boot Basics	3
✱	NULL	NULL	NULL	NULL	NULL

- **Question**

```
1 • create database quizapp;
2 • use quizapp;
3 • select * from question;
4
```

qid	ans	opt1	opt2	opt3	opt4	quest
1	2	Guido van Rossum	James Gosling	Dennis Ritchie	Bjarne Stroustrup	Who is well known as father of Java ?
2	2	Object-oriented	Use of pointers	Portable	Dynamic and Extensible	Which one of the following is not a Java feature ?
3	1	Aspect Oriented Programming	Any Object Programming	Asset Oriented Programming	Asset Oriented Protocol	AOP Stands for ?

- Test

```
1 • create database quizapp;  
2 • use quizapp;  
3 • select * from test;  
4
```

<					
Result Grid					
Filter Rows:					
Edit:					
	tid	testans	questionid_qid	quizid_quiz	userid_uid
▶	1	1	3	4	1
	2	2	1	3	1
	3	3	1	3	2
	4	1	3	4	2
*	NULL	NULL	NULL	NULL	NULL