

API Specification Doc

Quiz API

| Version | Date | Author | Description |
|---------|-------------|--------------------|----------------------------------|
| 1.0 | 10-Mar-2019 | Gomaa Abed Mohamed | Initial draft |
| 1.1 | 18-Mar-2019 | Amr Mohamed Fahmy | Added version number in response |

Index

1. Get categories

[Request](#)

[Response](#)

2. get score

[Request](#)

[Response](#)

[Conventions](#)

[Status Codes](#)

Methods

1. get categories

Get All Categories of the Available Quizes

Request

| Method | URL |
|--------|-----------------|
| POST | Quiz/Categories |

| Type | Params | Values |
|--------------|---------------------|------------------|
| HEAD POST | auth_key version | string number |

auth_key

The `auth_key` that was given in response to `/api/login`

version

The current version of internal recipe database. Each time when updates are pulled from the server through the web service, the internal database version is incremented.

Response

| Status | Response |
|--------|--|
| 200 | <p>Response will be an object containing the list of recipes (array) as well as the updated recipe database. Each item in the recipe array has the following structure.</p> <pre>[{ "quiz_id": 10, "title": "Data structure", "categoryId": 1 }]</pre> |

| | |
|-----|---|
| | An example response is:- <pre>{ "recipes": [[{ "quiz_id": 10, "title": "Data structure", "categoryId": 1 }], [{ "quiz_id": 12, "title": "OOP", "categoryId": 2 }]], "version": "4" }</pre> |
| 400 | {"error": "Please specify database version."} |
| 400 | {"error": "Invalid database version."} |
| 401 | {"error": "Invalid API key."} |
| 500 | {"error": "Something went wrong. Please try again later."} |

2. Get Score

Get the score of some user of specific quiz by its title and userId

Request

| Method | URL |
|--------|---------------------------------------|
| POST | Quiz/users?UserId =userId&QuizTitle=t |

| Type | Params | Values |
|------|-----------|--------|
| HEAD | UserId | String |
| POST | QuizTitle | String |
| POST | version | number |

version

The current version of internal database. Each time when updates are pulled from the API, the internal database version increases.

Response

| Status | Response |
|--------|---|
| 200 | An array containing the ID's of recipes to delete is given Example response:- <pre>{ "userId":1, "QuizId" :23, "QuizTitle":C++, "Score" : 7.5 , "version":"5"}</pre> |
| 400 | <pre>{"error":"invalid userID."}</pre> |
| 400 | <pre>{"error":"invalid Quiz Title."}</pre> |
| 401 | <pre>{"error":"Invalid Auth key."}</pre> |
| 500 | <pre>{"error":"Something went wrong. Please try again later."}</pre> |

Glossary

Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as **[optional]**
- The type of values accepted for a *request* parameter are shown the the values column like this **[10|<any number>]**. The | symbol means *OR*. If the parameter is **[optional]**, the default value is shown in blue bold text, as **10** is written in **[10|<any number>]**.

Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client's part

5XX - Error occurred in server's part

| Status Code | Description |
|-------------|---|
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |

| | |
|-----|--------------------------|
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |