Kevin The Hack Buster

API documentation and Database Schema

Version: 1.0

Please note that this is still an early release, and the final version will vary.

Table of Contents

Introduction

Basic information:

1. API is a standalone NodeJS application.
2. After development it must be protected with SSL certificate.
3. API offers open, publicly available part and protected administrator area.
4. Admin's part requires JWT authentication.
5. For this documentation domain is "kevinhackbuster.com". It will change after deployment.

Problems for further discussion:

Problem 1:

Will web panel use the same domain as this API?

Problem 2:

Access to API should be realized with subdomain or subdirectory?

Example subdirectory connection:

https://kevinhackbuster.com/api/example

Example subdomain connection:

https://api.kevinhackbuster.com/example

Problem 3:

Do we need support for open questions in this project? In this state API is ready for closed questions with one or multiple correct answers.

Problem 4:

How advanced telemetry we need?

Problem 5:

How are we planning to test our project for the production?

Problem 6:

Determine how many questions the quiz consists of?

API Endpoints:

Public part

The public part of the API is intended to provide the game with questions and check user responses. It also collects usage data.

0. To access any given API route, the public key is required to be sent via POST:

```
{
    "publicKey": "apiCurrentSecretPublicKey"
}
```

1. To receive random question from API:
   https://api.kevinhackbuster.com/randomquestion

result:

```
{
    "questionId": 3,
    "type": "single",
    "question": "Does anyone even read this?",
    "answers": [
        "Yes",
        "No"
    ]
}
```

2. To receive quiz questions IDs array:
   https://api.kevinhackbuster.com/generatequiz

result:

```
{
    "quizQuestionsIds": [1, 5, 13, 4, 12]
}
```

3. To get specific question make request with id of a question:
   https://api.kevinhackbuster.com/question/{question_id}

result:

```
{
    "questionId": 5,
    "type": "single",
    "question": "What city is the capital of Italy?",
    "answers": [
        "Rome",
        "Warsaw",
        "Paris",
        "London"
    ]
}
```

4. To validate question, send id of an answer with given structure (counting from 0):
   https://api.kevinhackbuster.com/question/validate
   POST request:

```
{
    "publicKey": "apiCurrentSecretPublicKey",
    "questionId": 5,
    "answer": 0,
}
```

result:

```
{
    "ans": "correct" // or "incorrect"
}
```

Admin part

0. **To access administration part of API user simply must make API calls without any additional parameter in POST requests. After successful login client will be given an http only cookie that contains token with required credentials.**

User authentication:

1. To login user need to make POST request with login and password.
   https://api.kevinhackbuster.com/administration/login
   POST request:

```
{
    "login": "admin",
    "password": "TopSecret"
}
```

   result:

```
{
    "status": 0 // or 1
}
```

2. To logout administrator simply must make following request. Server will unset http only cookie with token making access to api no longer available.
   https://api.kevinhackbuster.com/administration/logout

3. To edit user admin must send credentials.
   https://api.kevinhackbuster.com/administration/edituser
   POST request:

```
{
    "login": "newLogin",
    "password": "newPassword"
}
```

   result:

```
{
    "status": 0 // or 1
}
```

Handling questions:

1. To get all questions in database:
   https://api.kevinhackbuster.com/administration/questions

2. To add new question:
   https://api.kevinhackbuster.com/administration/questions/add

PUT request:

```
{
    "type": "single",
    "question": "How are you?",
    "answers": [
        "Good",
        "Great",
        "Bad",
        "Therrible"
    ],
    "correct": 1
}
```

result:

```
{
    "status": 0 // or 1
}
```

3. To remove question:
   https://api.kevinhackbuster.com/administration/questions/delete

DELETE request:

```
{
    "questionId": 4
}
```

result:

```
{
    "status": 0 // or 1
}
```

Statistics:

It is possible to make precise routes to specific statistics but because of the project vision only admin's web panel will render statistics therefore it can get .json file with entire telemetry data from a single request.

https://api.kevinhackbuster.com/administration/statistics

result:

the result should be discussed in further API development.

Database structure

Overview:

Database: ***Amazon DynamoDB***
Database type: ***NoSQL***

Note: Due to NoSQL nature of database, I am unable to provide a detailed table schema, but I will try to describe the structure and purpose of the tables.

Tables:

In this state API requires 4 tables:

1. PublicKey
2. Users
3. Questions
4. Telemetry

Tables descriptions:

PublicKey table contains only one element. This is a value that protects the public part of the API to prevent spam and DDoS attacks from third party software that would like to access the question database without permission or send requests that will generate false telemetry information. It is also a useful tool to end support for older releases of game.

Users table for now will only include one user who will be an administrator and will use the web panel. It was created to make the application more future proof.

Questions table contains questions, as the name suggests. In this version it can store closed questions with two or more answers and one or more correct answers.

Telemetry table is designed to store statistics data about application.