Dog Quiz Application

Introduction

This is the documentation for the DogQuiz API (DQAPI).
With this DQAPI you can register users, login, generate a dog quiz and keep track of the score.

Prefix

All Resources need to be prefixed with: "http://localhost:3333/"

Resource formats

The resource format is JSON (application/json).

Resources:

Note: The POST-request always checks if the right Content-type is applied. If you put a content-type besides "application/json" you will get an Error.
Example:

headers: { "Content-Type": "application/json" } = correct
headers: { "Content-Type": "html/text" } = incorrect

Resource: Register (POST)
Resource URL: api/register.php
This POST-request allows you to add a new user to the register. The body of the request must have the structure:

```
{
   username: string,
   password: string
}
```

If the username is already in use, the server will send a response with the status 409 (Conflict) and a message.
If the password is too short (needs to be 4 characters or more) the server will send a response code 400 (Bad Request) and a message.
If either or one of the username and password are empty, the server will send a response with status 400 (Bad Request) and a message.

If everything checks out, the server response with status 201 response and an object:

```
{
   username: string,
   password: string,
   points: 0 (number),
}
```

Resource: Log in (POST)
Resource URL: api/login.php
This POST-request allows you to check a pair of credentials (username, password) by the registered users. The body of the request must have the structure:

```
{
   username: string,
   password: string
}
```

If the username and password does not exist in the register, the server will send a response with status 404 (Not found).
If either or one of the username and password are empty, the server will send a response with status 400 (Bad Request) and a message.

If everything checks out the server will send an response with status 200 and an object:

```
{
   username: string,
   password: string,
   points: current points(number),
}
```

Resource: Quiz (GET)
Resource URL: api/quiz.php
This GET-request allows you to get an object with a random dog picture and an array with random dog names with the keys true or false.

If the request method isn't GET the server will send an response with status 400 (Bad Request) and a message.
If the folder/file with images doesn't exist, the server will send a response with status 404(Not found) and a message.

If everything checks out the server will send an response with status 200 and an object:

```
{
   image: string,
   alternatives: [
      {
         correct: false or true,
         name: string,
      },
      {...},
      {...},
      {...},
   ]
}
```

Resource: Points (POST)
Resource URL: api/points.php

This POST-request allows you to add a point to the current logged in user. The body of the request must have the structure:

```
{
   username: string,
   password: string,
   points: how many points you want to add (number),
}
```

If you are missing any of the keys in the body structure, the server will send an response with status 400 (Bad Request) and a message.

If everything checks out the server will send an response with status 200 and an object:

```
{
   points: users current points (number),
}
```

Resource: Points (GET)
Resource URL: api/points.php
This GET-request allows you to get an array of the five users with the highest score in the order highest score to the least score of the five users.

If everything checks out the server will send an response with status 200 and an array with objects:

```
[
   {
      username: string,
      points: users current points
   }
   {...}
   {...}
   {...}
   {...}
]
```

Errors

If the read fails for any reason (for example, non existing url path), the API will respond with a standard error response.