Objective

The objective of this workshop is to practice designing RESTful web services. In particular, this workshop will also focus on the GET and the POST method and designing resource names.

Setup

a. Open `workshop01` folder in your IDE. Open `main.js` file in the editor.
b. Execute

```
npm install
```

in `workshop01` directory to install the required modules

Workshop

You are given a Node application which interacts with a boardgame database. Implement the following REST endpoint and features.

The database functionalities are found in the `database.js` file.

`GET /games`

Return an array of game resources URL in a JSON array.

Use `findAllGames()` to get the result. The function takes 2 positional parameters

- offset – set which game to start retrieving the data from. Defaults to 0
- limit – limit the number of games returned. Default to 10

Allow a way for the HTTP request to configure these 2 parameters. If any of these values are not specified in the request, then use the default value.

You should return the result in the following format

```
[ "/game/<game_id_1>", "/game/<game_id_2>", … ]
```

`GET /game/<game_id>`

Eg. `GET /game/abc123` – return the boardgame details with `game_id` as `abc123`.

Use `findGameById(gameId)` to find the game from the database.

The request should return the following status code

- 404 – if the `game_id` does not exist
- 200 – otherwise

Return the following JSON payload according to the status code

- 404

```
{ "error": "Game <game_id> not found" }
```

- 200 – result from `findGameById()`

`POST /comment`

Post a comment for a game. The request should send the content type as JSON payload with the following properties

- `user` (string) – user name
- `rating` (integer) – your rating of the game, 1 (worst) to 10 (best)
- `c_text` (string) – your comments
- `gid` (number) – the game id that you are commenting on

Use the function `insertComment(comment)` where the `comment` parameter is the above object. The request should return the following status code.

- 400 – if any of the following conditions is true
  - Any of the above properties is missing, or
  - If the `rating` property is not within the specified rating range, or
  - the comment text is empty, or
  - the game id does not exist
- 200 – successfully posted the new comment

Return the following JSON payload according to the status code

- 400

```
{ "error": "Your error message" }
```

- 200

```
{ "id": "<comment_id>" }
```

Optional Workshop

Content Negotiation

Allow the request

`GET /game/<game_id>`

to return result as CSV (comma separated values). The response content should be in the following format

```
id,name,year,ranking,users_rated,url,image
1,Die Macher,1986,223,4777,https://…,https://…
```

with a 200 status code.

Return a 415 status code if the requested media is not CSV or JSON with the following format

```
{ "error": "Media <media> is not supported" }
```

Partial Results

Allow the response of

`GET /games`

to advertise support for partial request. Implement the `HEAD` method to return the HTTP Range header.

Modify the `GET /games` request handler to support partial request.

Rate Limit

Throttle the **`GET /games`** endpoint to 3 request per 5 seconds.

Submission

When you have completed this workshop perform a push to your remote repository.

```
git add .
git commit -m 'workshop01'
git push origin master
```