Objective

The objective of this workshop is to learn how to query a document database

Setup

a. Import the given JSON document into a new MongoDB database. Call the database `boardgames` and the collection from `game.json` and `comment.json`. Call the collection `games` and `comments` respectively
b. Create a SpringBoot application with the following dependencies
   i. Spring Boot Dev Tools
   ii. Spring Web
   iii. Thymeleaf
   iv. Spring Data MongoDB
   v. JSON-P

Workshop

You are to create the following REST resources

a. Browsing the games by name

```
GET /games
Accept: application/json
```

The request accepts the following optional query parameters
`limit` – limits the number of games returned. The default value 25
`offset` – specify offset of the first row to be returned. The default is 0
The endpoint returns the result in the following JSON object

```
{
  games: [ <array of games> ],
  offset: <offset or default value>,
  limit: <limit or default value>,
  total: <total number of games>,
  timestamp: <result timestamp>
}
```

Each element of the `games` array has the following structure

```
{
  game_id: <ID field>,
  name: <Name field>
}
```

b. Browsing the games by rank

```
GET /games/rank
Accept: application/json
```

Games are returned based on their rank (`ranking` field) in ascending order. The endpoint has the same query parameters as a. above.
The return result is also the same as a.

c. Get game details

```
GET /game/<game_id>
Accept: application/json
```

Returns the details of a game base on its `_id` (`<game_id>` is equals to `_id` field). The following JSON object is returned for a game

```
{
  game_id: <ID field>,
  name: <Name field>,
  year: <Year field>,
  ranking: <Rank field>,
  average: <Average field>,
  users_rated: <Users rated field>,
  url: <URL field>,
  thumbnail: <Thumbnail field>,
  timestamp: <result timestamp>
}
```

The REST endpoint should handle non-existence game ids.