Objective

The objective of this workshop is to create a simple RESTful API that will persist data in to a backend data store. JSON will be used as the data exchange format between the client and the server

Setup

a. Create a new repository for this workshop
b. Create a new Redis database. You may have to delete or clear your previous database if you have previously create one
c. You will be provided with a JSON dataset
d. Generate a SpringBoot application with the usual dependencies
e. Add JSON-P library to your application

Workshop

Task 1

Write a REST endpoint that will insert 1 document (in JSON) into the data store. The end point is as follows

```
POST /api/boardgame
```

Determine what key to use when you save to Redis.

Once the document has been inserted return at 201 status code with the following JSON payload

```
{ "insert_count": 1, "id": <Redis key>  }
```

Task 2

Write a REST endpoint that will retrieve a given board game. The following HTTP request performs the query

```
GET /api/boardgame/<boardgame id>
```

The boardgame should be return as a JSON document.

If the boardgame is not found, return a 404 status and an appropriate error object.

Task 3

Write a REST endpoint that will update a document

```
PUT /api/boardgame/<boardgame id>
```

The above REST endpoint takes the payload from the request body and attempts to update the data stored in Redis with the corresponding key `<boardgame id>`.

If `<boardgame id>` does not exists, the endpoint should return a 400 status code and an appropriate error object.

If the update was successful, return a 200 status code and the following payload

```
{ "update_count": <count>, "id": <Redis key>  }
```

where `<count>` is the number of documents you have updated.

This endpoint takes an optional parameter, `upsert`. If `upsert` is set to `true`, then the endpoint should perform and insert if the <boardgame id> does not exist.

Assume that if query string, `upsert` is specified, then it is set to `false`.

Task 4

Deploy to Heroku.

Submission

When you have completed the workshop, commit and push your code to your Github repository.