ASSIGNMENT TWO – API DOCUMENTATION AND REPORT

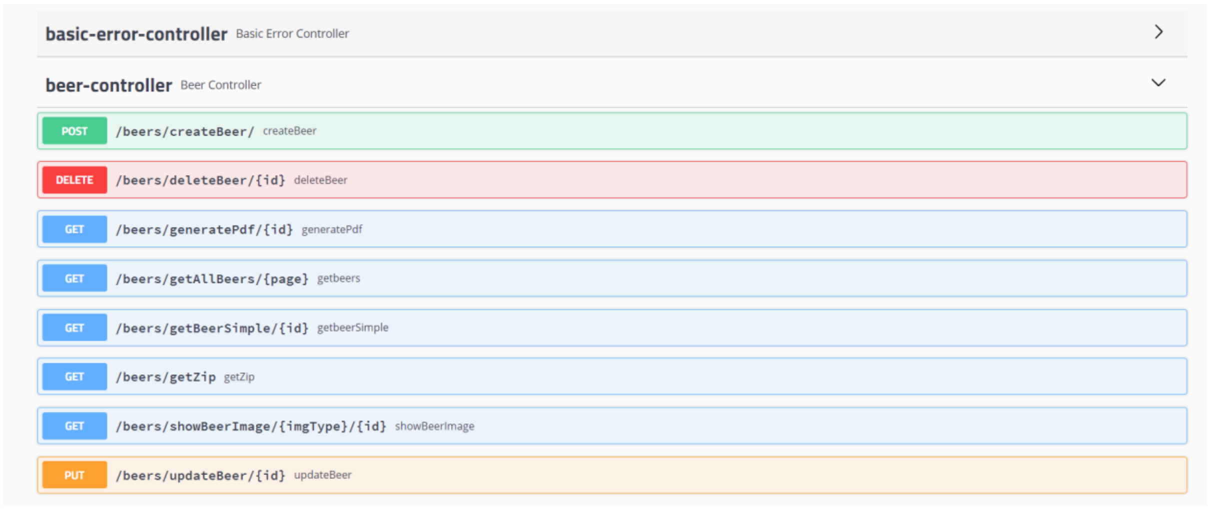
K00235601 – Lukasz Dębicki

MARCH 22, 2022
TECHNOLOGICAL UNIVERSITY OF LIMERICK

Contents

Documentation

Link: http://localhost:8888/swagger-ui.html#/



basic-error-controller Basic Error Controller

beer-controller Beer Controller

POST `/beers/createBeer/` createBeer

DELETE `/beers/deleteBeer/{id}` deleteBeer

GET `/beers/generatePdf/{id}` generatePdf

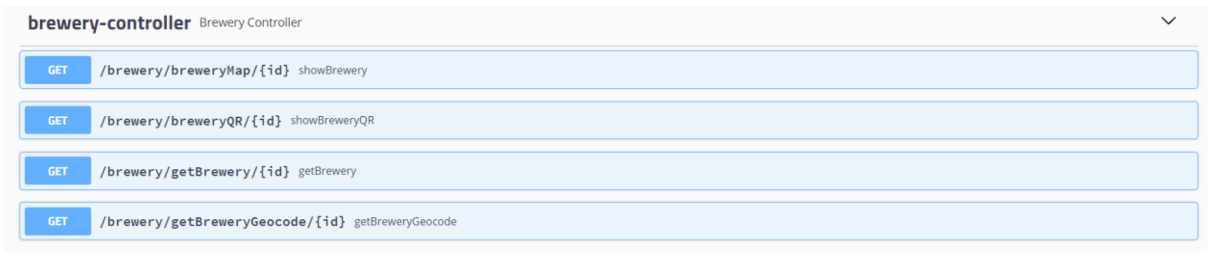
GET `/beers/getAllBeers/{page}` getbeers

GET `/beers/getBeerSimple/{id}` getbeerSimple

GET `/beers/getZip` getZip

GET `/beers/showBeerImage/{imgType}/{id}` showBeerImage

PUT `/beers/updateBeer/{id}` updateBeer



brewery-controller Brewery Controller

GET `/brewery/breweryMap/{id}` showBrewery

GET `/brewery/breweryQR/{id}` showBreweryQR

GET `/brewery/getBrewery/{id}` getBrewery

GET `/brewery/getBreweryGeocode/{id}` getBreweryGeocode

Self-Evaluation

- Uses CrudRepository interface for pagination without the use of the PagingAndSortingRepository interface, which would interfere with the HateOAS WebMvcLinkBuilder class methods such as ‘methodOn’. The Slice object used allows the user to specify the page to be displayed and by default, I set the number of beers to be shown on one page to 10. My implementation of Slice is superior in the sense that it avoids triggering a count query to calculate the overall number of pages as that might be expensive and only knows whether a next or previous Slice is available.

- Implements the use of two controllers instead of just one like in assignment one. This allows better code readability, assortment of functionality and better error handling but increases code complexity in smaller applications where one controller may suffice.
- Zip, PDF and QR functionality is split into separate classes rather than being in a mapping method. This enables code reusability in the case that more methods in the API require this functionality in the future.
- Each method is annotated with ‘consumes’ and ‘produces’, and a value of JSON, which ensures the system only ever inputs and outputs JSON values as per assignment requirements. This in turn improves error handling and ensures unexpected behaviour is caught by the error handling existing in the API.
- Model classes in the system implement Bean validation, which are used for validating user inputs. This stops unexpected or invalid values from being saved in the database, reducing the risk of malicious attacks and redundant data.
- ResponseEntity is used to control the content of an API response back to the client requesting the API call, allowing the developer to have full control over the response, however the disadvantage is code complexity because Spring by default creates a default ResponseEntity regardless of if it is specified or not, which may not always be accurate.
- Beer images are returned from the API as a byte array, allowing the body of the ResponseEntity to display an image to the client, where otherwise sending images directly from the system directory is not allowed.
- The brewery geocode method implements Google Maps and plots the exact co-ordinates of the selected brewery on the map.
- Brewery QR method returns a BufferedImage to the client, which allows the displaying of the QR code as an image rather than a JSON string. The QR code uses the MECARD format, which attempts to add relevant details to the phones contact list upon scanning. The large advantage of the MECARD format is its ease of use and documentation freely available online.

- The Zip functionality works flawlessly with one small issue, where an extra empty folder is created in the zip file alongside the beer image folders. After some investigation and trial and error, I suspect the zip4j dependency itself is at fault.
- The update beer functionality with PutMapping updates the last modification date with the actual current date of when the API function was executed.

Benchmarking and Enhancements

Using this API as an example, the benefits that this API bring to a project are:

Integration: API's allow content to be embedded from any site or application more easily, which guarantees more fluid information delivery and an integrated user experience. GlobalBeers could confidently use this API for their web enabled application or future desktop-based applications.

Lightweight: One of the main benefits of mainly REST API's is that they rely on the HTTP standard, which means its format-agnostic and you can use XML, JSON, HTML, etc. This makes REST API's fast and lightweight, which is necessary for mobile app projects, IOT devices etc. GlobalBeers would save capital on processing costs and training.

Efficiency: When access is provided to an API, the content generated can be published automatically and is available for every channel. It allows it to be shared and distributed more easily. GlobalBeers would benefit from this with increased ease of use for development.

More scope: With an API, an application layer can be created which can be used to distribute information and services to new audiences, which can be personalized to create custom user experiences. GlobalBeer benefits by having reusable functionality without extra development costs of new software.

Independence: REST API's client and server sides are independent. The REST protocol separates the data storage and the UI from the server, which means developers can work on different areas of a project independently. GlobalBeer development costs and time would decrease due to multiple developers working on tasks simultaneously.

Scalability and flexibility: REST API's can be scaled quickly primarily due to the separation between the client and the server, in the case that traffic heightens to the API, crashing wouldn't occur.

There are three enhancements I would make to this API:

1. Enable Google Maps billing to remove the ‘For developmental use only’ watermark always placed on the map
2. Add the ability to filter beers to the getAllBeers method in the API, with the extra parameters selecting the property of the beer class and by sort by either ‘ascending’ or ‘descending’
3. Implement the Google Drive API into this API to allow the ZIP file of beer images to be automatically saved to the online drive