Advanced Enterprise App Dev

Assignment 2

Michael Hickey

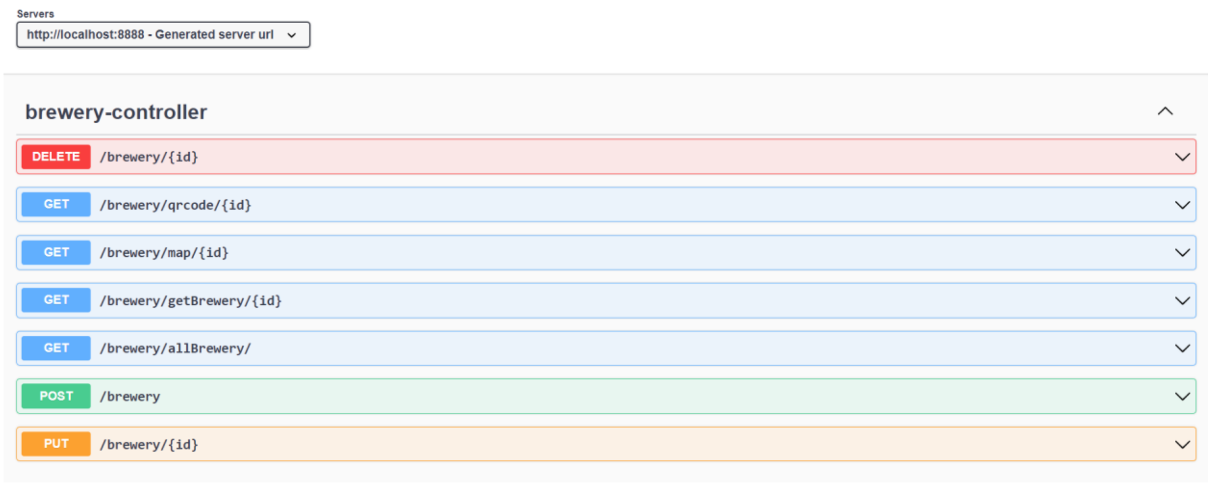
K00243046

Software Development

Year 4

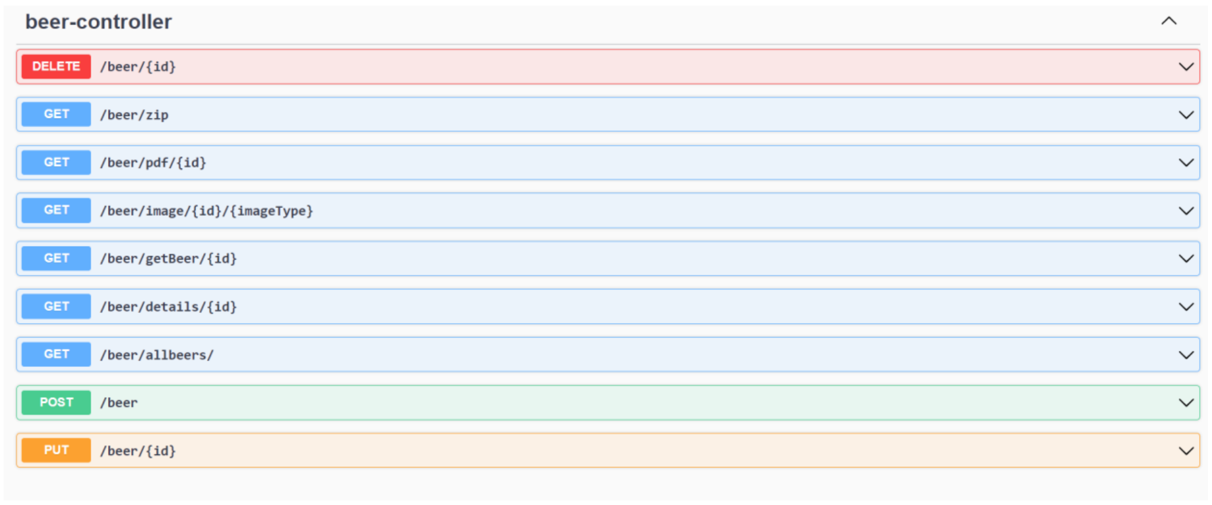
1.0 Documentation

Link to documentation when running project = http://localhost:8888/swagger-ui/index.html#/

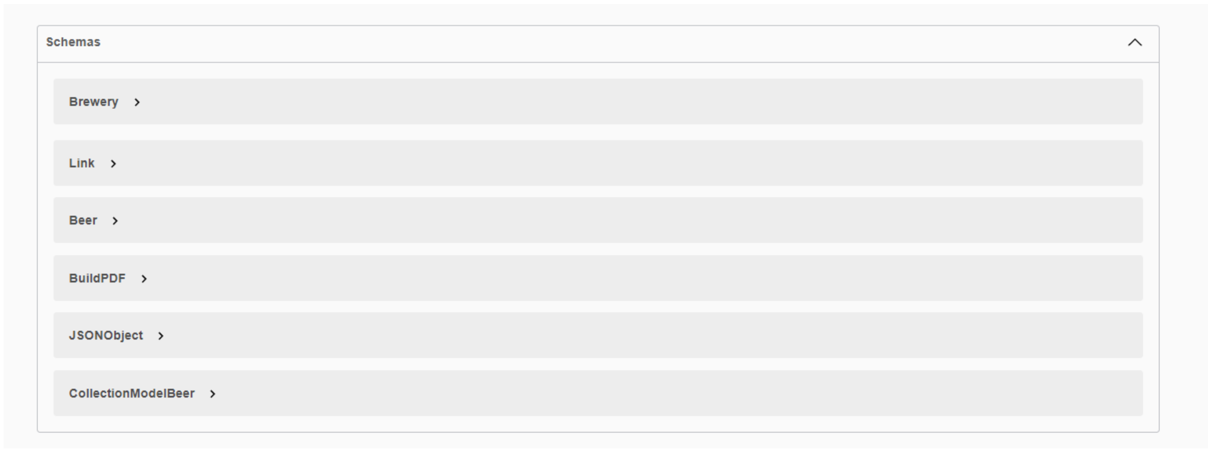

Servers
http://localhost:8888 - Generated server url

brewery-controller

DELETE /brewery/{id}
GET /brewery/qrcode/{id}
GET /brewery/map/{id}
GET /brewery/getBrewery/{id}
GET /brewery/allBrewery/
POST /brewery
PUT /brewery/{id}


beer-controller

DELETE /beer/{id}
GET /beer/zip
GET /beer/pdf/{id}
GET /beer/image/{id}/{imageType}
GET /beer/getBeer/{id}
GET /beer/details/{id}
GET /beer/allbeers/
POST /beer
PUT /beer/{id}


Schemas

Brewery
Link
Beer
BuildPDF
JSONObject
CollectionModelBeer

2.0 Self-Evaluation

API Strengths

One strength of this API created is its use of Hateoas to link certain pages together. When a user views all the beers, they have the option of following a link for a specific beer. The link will take the user to the page containing the information on the beer that was selected. Here another link will take the user back to the page to view all the beers. This functionality is also available for the brewery with the addition of a link to generate a QR code for a specific brewery. The incorporation of Hateoas provides easier navigation around the site for the user while also linking pages to information that is related to the current page.

This API follows best practices around naming with the use of Http verbs. The Http methods used were GET, PUT, POST and DELETE. Nouns Are preferable to verbs In Rest API when naming resources with verbs being used primarily for commands. This API follows this best practice for the resources. This API also avoids using any unnecessary forward slashes when forming a URL particularly a trailing slash, to avoid any confusion for the user. The API also incorporates the use of consumes and produces allows the requests and responses to be customized.

Bean validation is enabled in this API for updating or adding beers or breweries. Bean validation is important as it stops any unwanted data entering the database which could also lead to a security concern. It also tells the user what they are doing wrong instead of just displaying http status responses, allowing the user to fix there mistake and improving overall performance of the project.

API Weaknesses

API pagination is needed if you have a lot of data and endpoints. Pagination signifies that the query result has been given an order. This API does not include pagination which greatly reduces the readability as all data is returned on one page. If there is a huge amount of data, then the applications performance may be affected. Pagination also contains functionality for sorting the results. If an API request returned a huge amount of data that was not sorted this would be a problem.

Validation around http status responses is lacking in this API and is an important feature when developing Rest API's. This API offers some validation around id values that are not in the database which would return a 404 not found response. However, the lack of validation across the API shows a big weakness. Validation can inform users if there is a problem with the path they have entered and return with an appropriate response that the user can understand. The lack of validation also goes against best practices when designing and developing Rest API’s.

Another weakness of the API is the lack of variation between the beer controller and brewery controller. The requirements were to avoid making the functionality the same, however after attempts to implement new functionality failed, the brewery implemented the saw functionality of the beer.

3.0 Benchmarking and Enhancements

APIs bring a number of benefits to a project with one being improving connectivity. APIs, often known as internal APIs, can help businesses increase cooperation and internal communication. The basic capability of APIs is connectivity, it allows disparate systems, applications, and platforms to interact and share data, as well as execute a variety of operations. This increased connection and collaboration results in interoperable components that let businesses offer their intended functions without constraints.

Another benefit of API use in a project is its ability to enhance user experience. API users may take control of their own user experiences, allowing organizations to disclose their data and services via APIs, opening up a world of possibilities. These interfaces allow user to take control and identify its flaws. Developers may use APIs to design solutions that fulfil specific client requirements, which would be impossible to do otherwise.

I believe my API would benefit from having more security applied to it. In my opinion security is key regarding any API and this project is lacking in this department. An option to show or hide vulnerable details associated with a specific item in the database is one example of a benefit of having more security. Also, by always using HTTPS would enhance the API, any authentication credentials may be reduced to a randomly generated access token by utilizing SSL.

I think this API would benefit from the document API first approach. Before a single line of code is produced, the document API first model requires that all documents be completed and suitable for consumption by a person or a computer, this implies the document is organized rather than free flowing. This helps with the design of the project, which will lead to a better coded API and allows best practices to be adhered to more easily.