Rest API Documentation

# 7.1 Documentation

Graphical user interface, application

Description automatically generated

{size} for the Beer/FindImage/{id}/{size} function uses either large, or thumbs as a parameter.

/allImages does not work in Firefox, but does in Postman or Google chrome, this is possibly due to request type or how the browser handles .zip files.

# 7.2 Self Evaluation

The rest API I have developed has its benefits and downsides. There are many things that could have been improved in the API. There are a few existing issues with the API, for one the API currently has no functionality added for caching, so request cannot really be cached on the client or server-side which could cause more requests to be made than needs to be. Adding functionality for caching could be very beneficial to this Rest API. There is also Limited error handling in this API, while I tried to add as much Error 404 response codes if objects are not found there is certain case scenarios in this project which are not covered fully. For example, what if a beer PDF was generated but it was missing data on the style or category name, currently in the project there is little error handling to deal with such a scenario or to let the user know about such an error. A current big downside to the /allImages function is that it works based off a ZIP that is saved and then loaded and returned to the user. This functionality has not been tested with several requests at once and this could potentially cause issues with the timing as the data is updated. What if two requests are made at the same time and the file is being overwritten while it is fetched by an API? There is a high likelihood for error in this function and given more time it should be made so that the function does not save it but rather returns it from memory like the current implementation of the /findPdf function. A good and bad thing about the API is that it works mostly with JSON files, and this is a good thing because JSON is possibly one of the most popular and commonly used modern formats for sending data through a Rest API, but it could be a nice feature to add options for XML and such. A good thing about the API is I currently tried to keep it as stateless as I could have, and requests do not depend on other requests to the best of my knowledge. A good thing about the API is that it currently implements swagger so that documentation can be easily viewed. All functions attempt to either return response code 500 for OK or error code 404 if objects were not found. A con of the project is possibly more testing could be done to IRON out possible misinputs and errors. I tried to keep the function names as consistent as possible however some functions such as allImages, should probably be named something along the lines of findImages for consistency, however probably the better option would be renaming all the finds to just nouns and removing the find verb which would make the method calls less verbose and just as self-explanatory. However, I do believe the function names have been made clear and concise and they do a decent job of explaining what the method returns and they avoid ambiguity. They also stick to use of Camel-case across the board allowing for good consistency, however many rest API’s prefer to use a ‘-‘character to separate words and this could be a better naming scheme for the purpose of standardisation.

# 7.3 Benchmarking and enhancements

One benefit I have noticed myself in the use of an API when working with an API is that it can remove concern of the front-end and the data. For example, when working with react, I can generate a table using any data fetched from the API, so that allows you to create a front-end framework that can render any Data it receives if it is a similar format, for example I could request brewery information or beer information and render it using one view function in React. Leading to a nice separation of concern and modularity. As mentioned already the project could greatly benefit from additional error handling and from the ability to cache requests which would allow for less requests in theory and hence make the API more efficient. The POST and Delete functions could deal with additional error handling. As mentioned in the above section the implementation of /allImages is currently not ideal as it sends the file from the tmp folder which could be affected by several requests at once which means the request is possibly not idempotent. It could additionally provide more formats for the data, XML Could be used additionally as well as JSON. Finally the project has no authorisation or methods of limiting requests which is also not ideal as the server is vulnerable and provides no security features.