CST 3130 - Coursework 1:

Price Comparison Website

Student: Gabriel Pislar

Student ID: M00618169

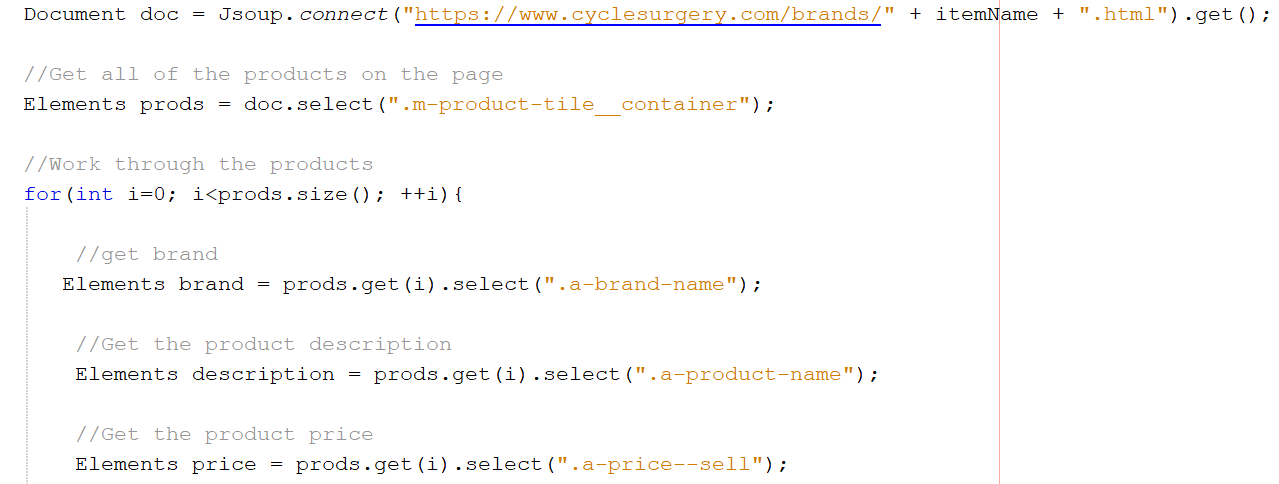
# Website description

This website is a price comparison website that allows users to search for bikes and to compare their price scrapped from different websites.

The web scraping from <https://www.ribblecycles.co.uk> is shown below. The web scrapping is done using JSOUP. In the first line the “itemName” value is added from the search to the end of the url to require the named products. Then that HTML document with the products is downloaded. I identified which div is showing the products and I assign it to elements prods, then a for loop is done for each product. On this for loop the price div and product description is taken and stored later in the database using hibernate.

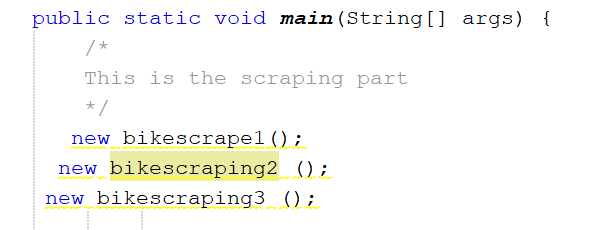


The second website from where I scraped bikes is [www.cyclesurgery.com](http://www.cyclesurgery.com) . From this website the price div, brand div and item description div are taken and assigned to a variable that I am using later to store it in the database.



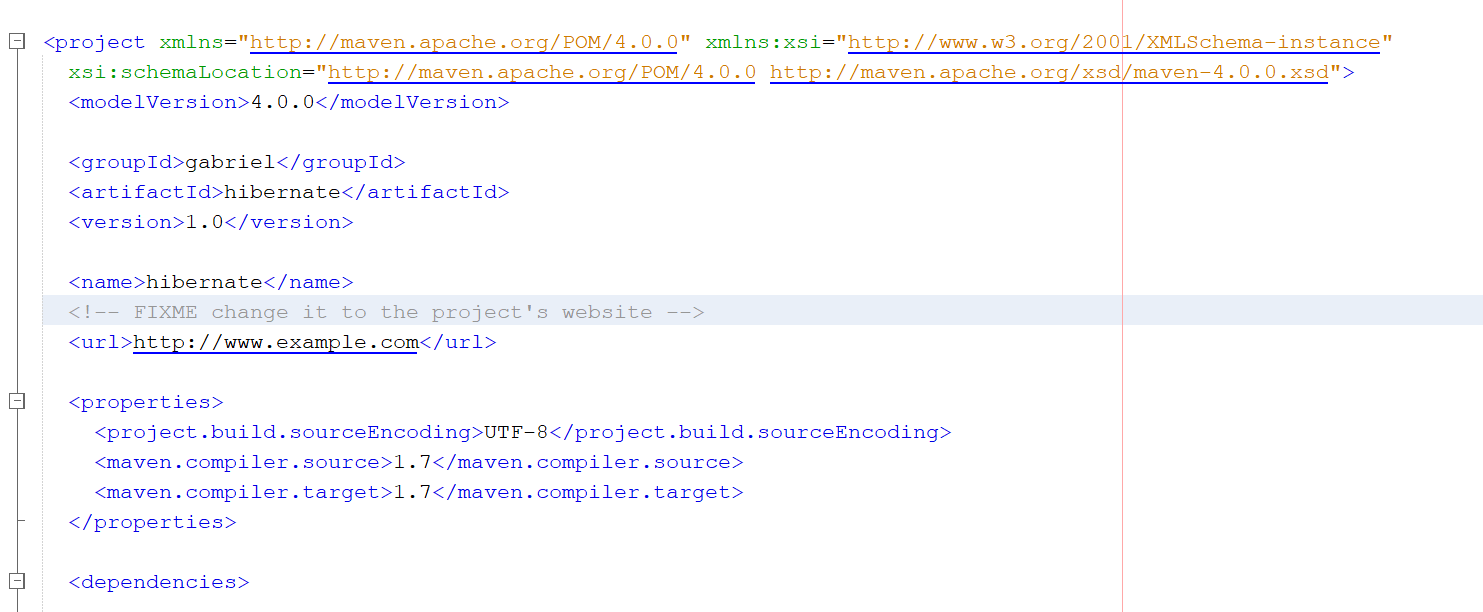
# Threads

I am using threads to pull data from different websites. Each thread called is taking data from one website.



# Maven

I have used Maven to build the Java project. Also I have used Maven to test my software and to build Jar files and included dependencies needed to run my code. Bellow is a part of my POM code.



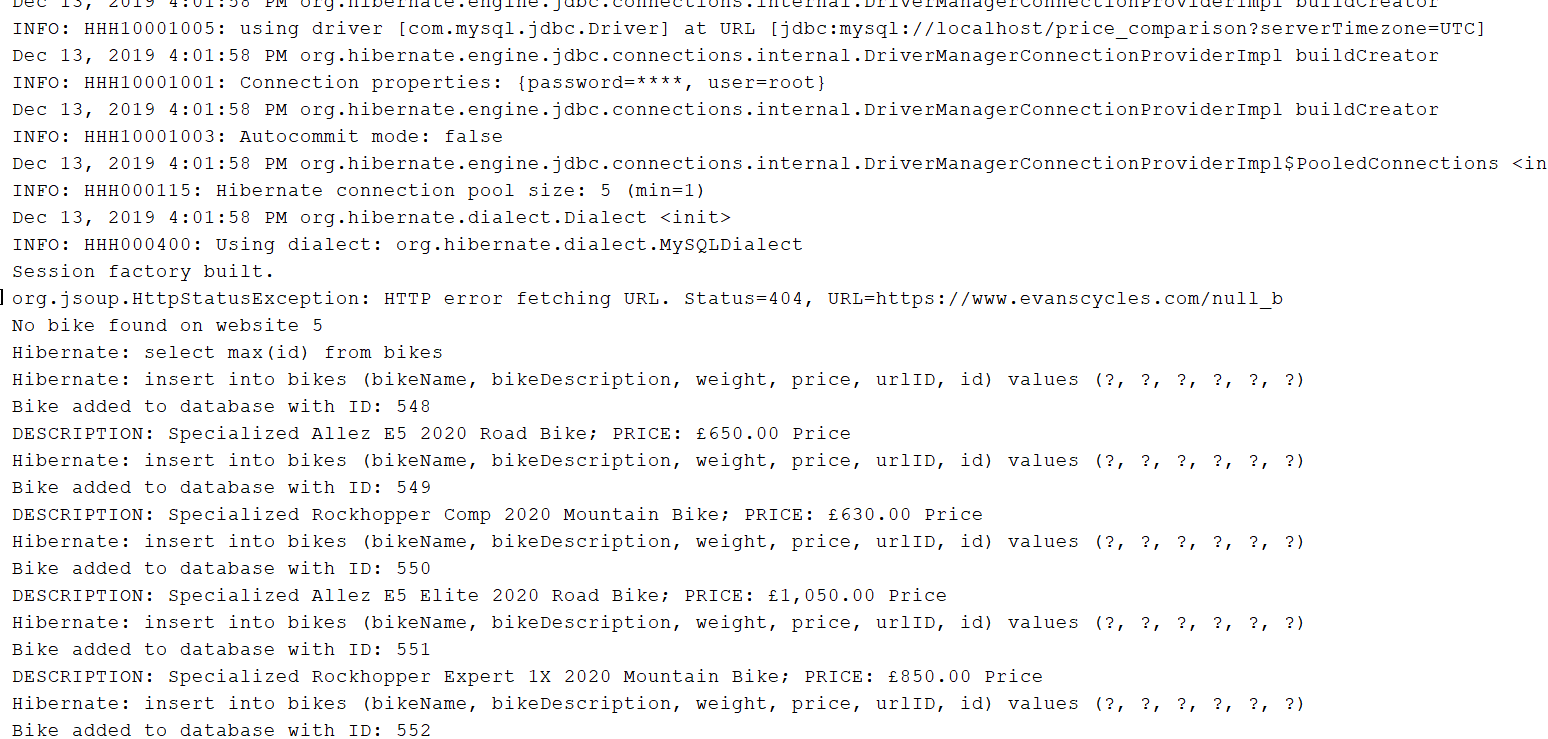


# Spring



# Hibernate

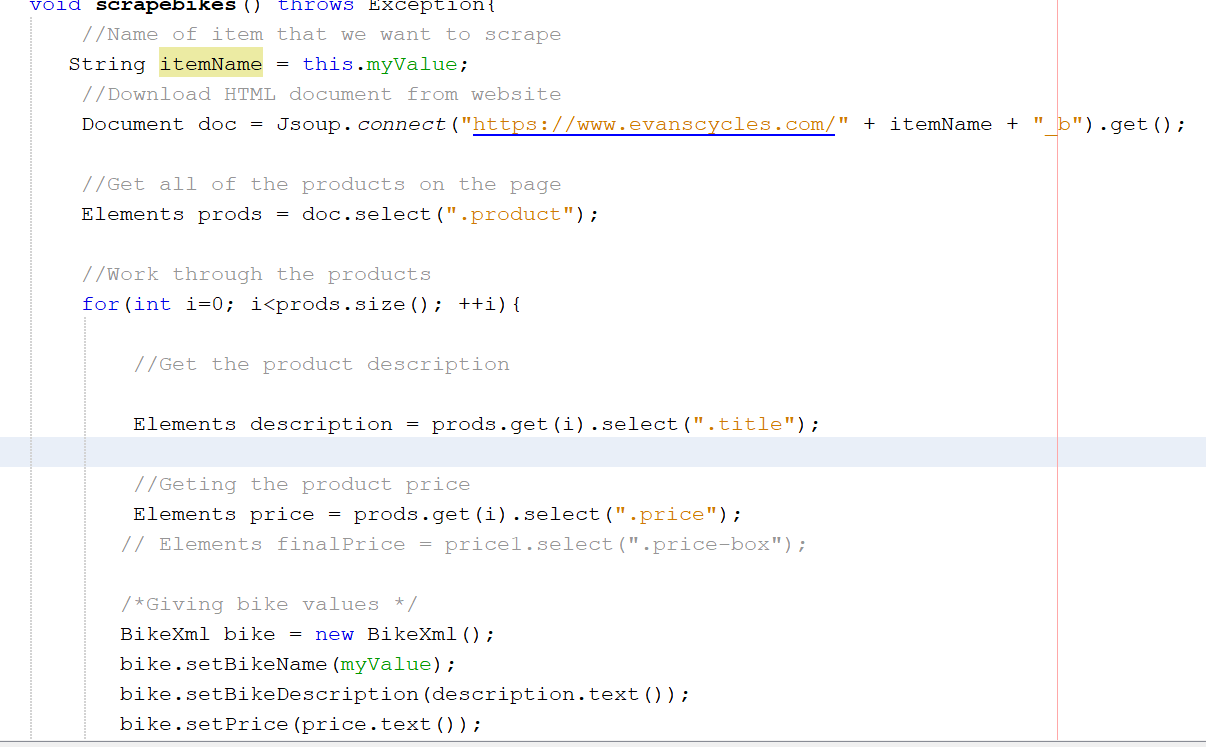
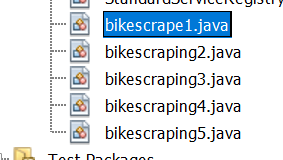
I have used hibernate to save scraped from all five websites to the database.





# Web scraping

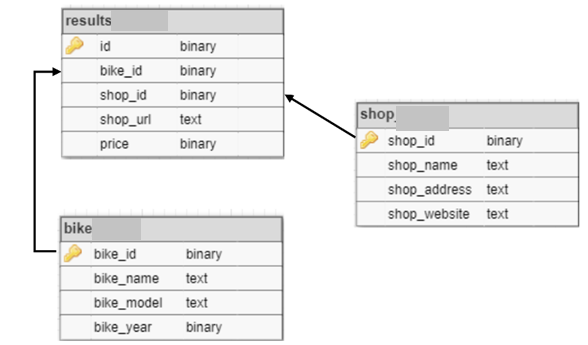
I have made five web scraping classes that I call on the main class. I have successfully scraped from five websites.



# Data

The data scraped is saved in the MySQL database. I have scraped over 500 rows.

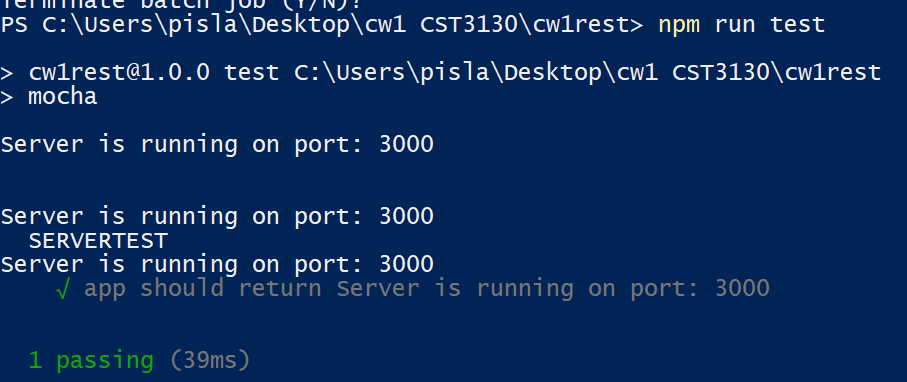
I have used same naming convention for all fields and tables. The data is stored efficiently in the database.



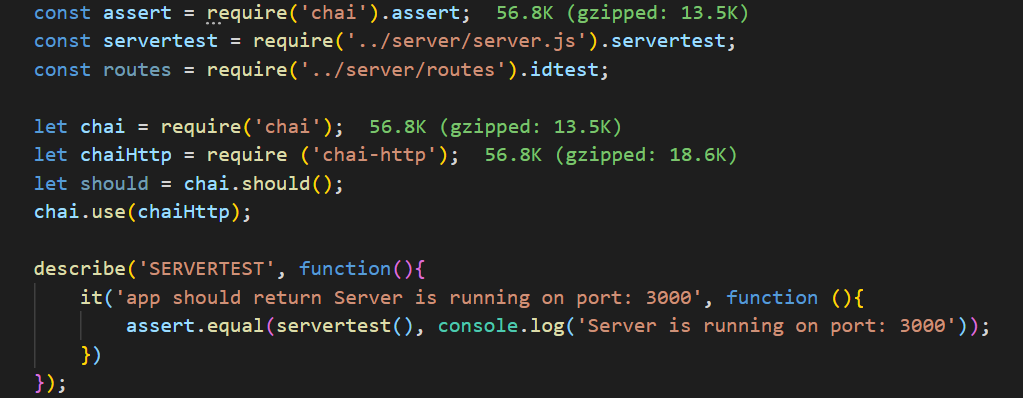


# Testing

Testing server using Mocha.



This one test:

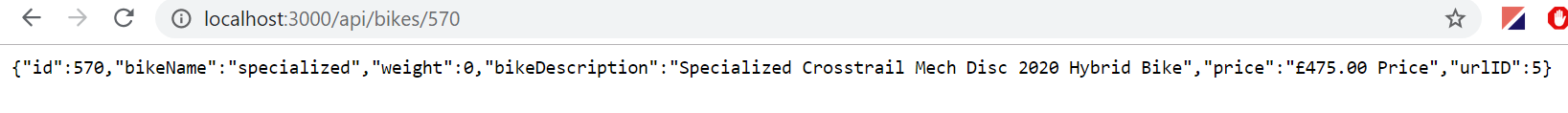


Unit testing for Java

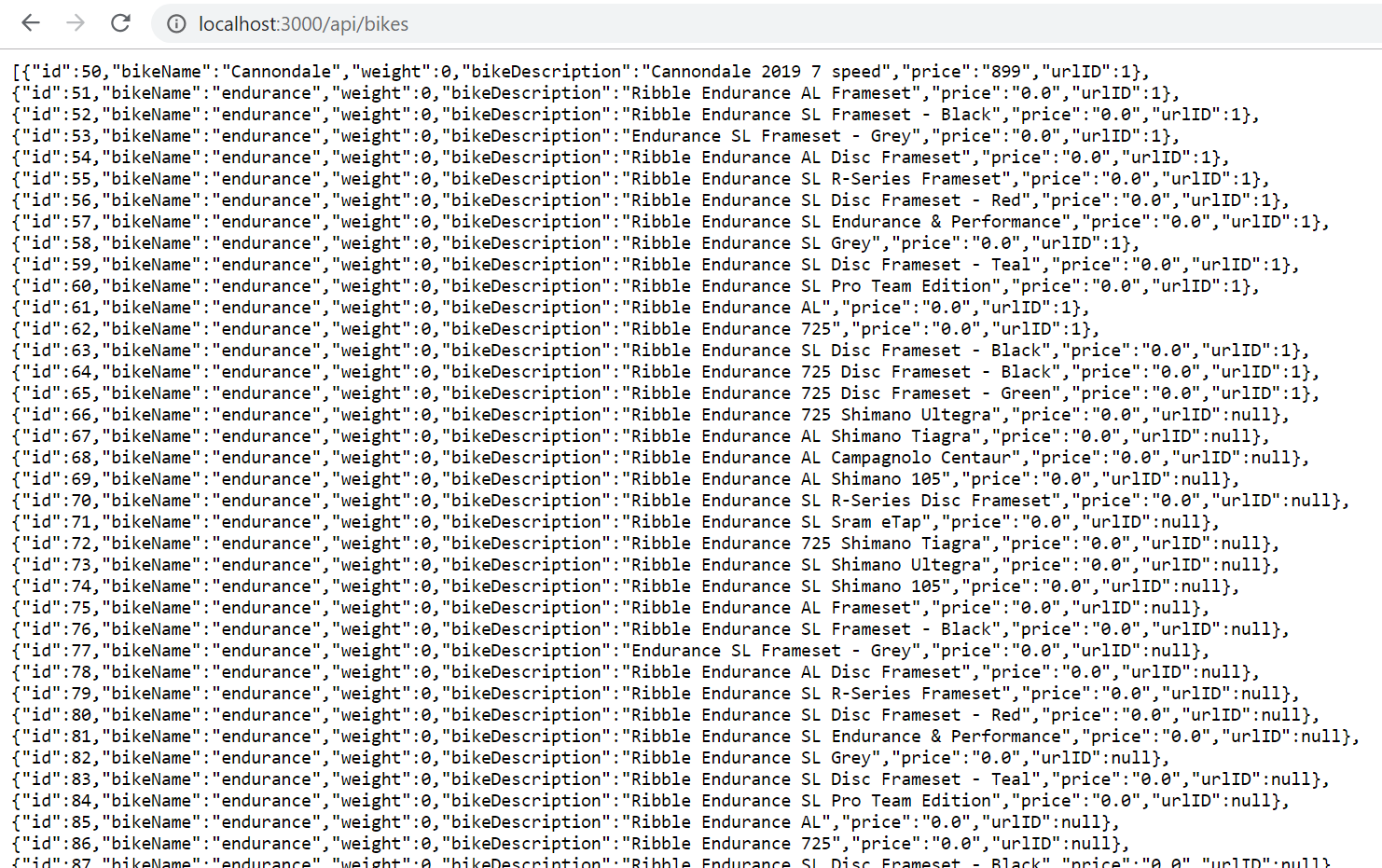


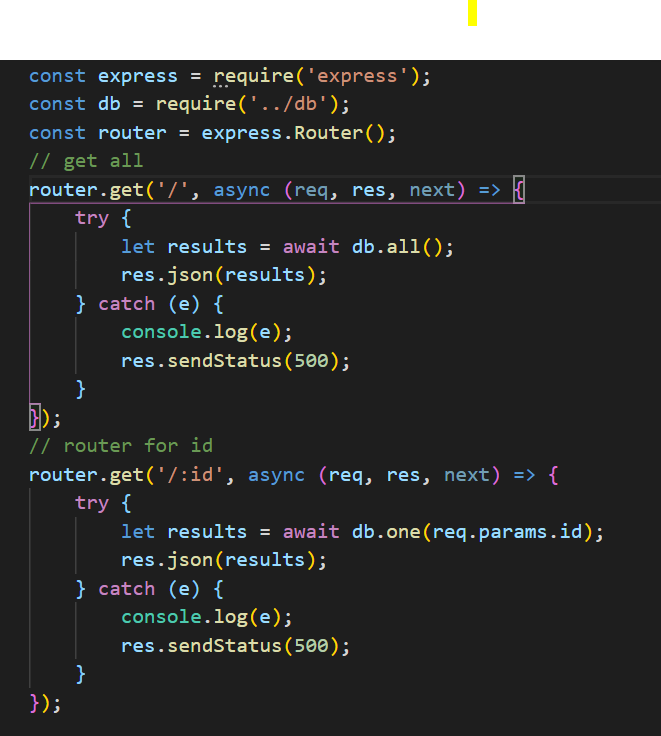
# REST API

The rest api uses path api/bikes/:id for getting an id.



To get all the bikes the path is api/bikes/





# Product display

Products are displayed in array format.

# Website quality

The website has an autofill search bar that retrieves automatically data from the database like a google search bar prediction.

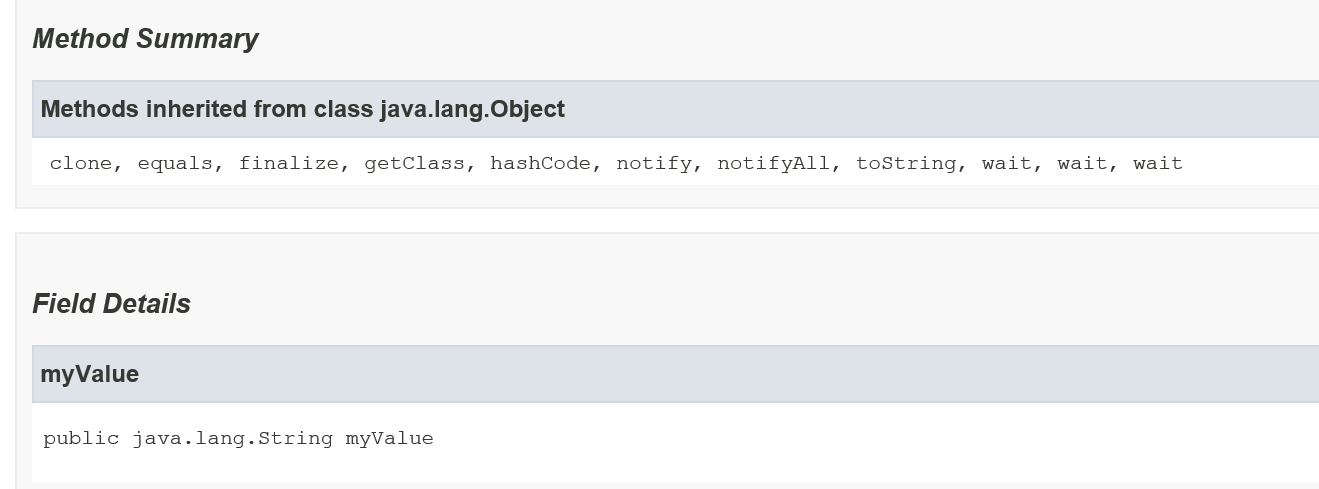


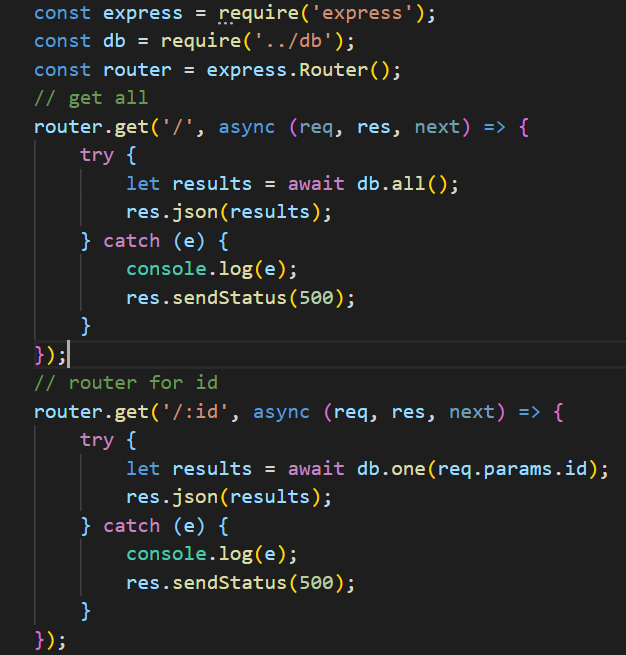
# Code quality

The code is well organised, and I have used comments for each function.

I used correct Java naming conventions.

I have created meaningful api documentation.





The code is very organised and well formatted, and I have used comments for each function, class, methods, variables.

