ShopAnalyst HackerEarth:

**Technologies used:**

Java 8

Spring boot

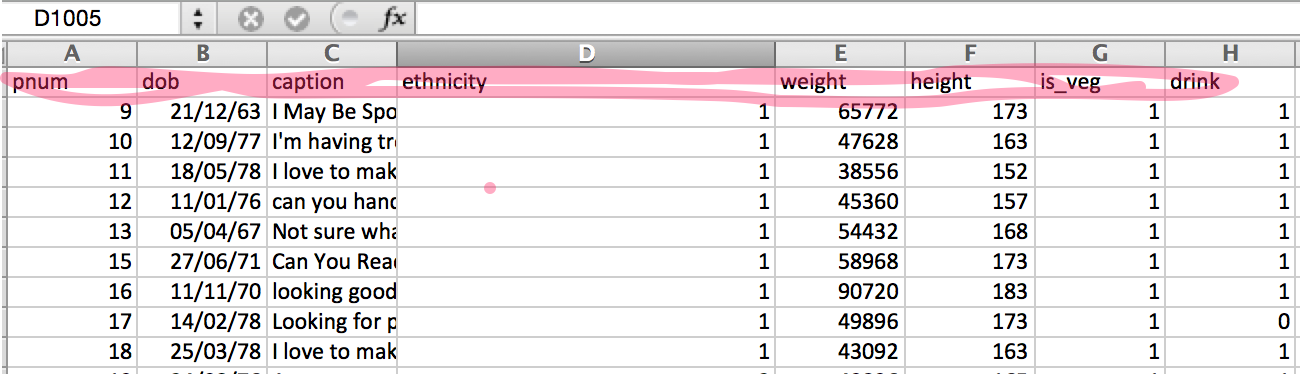
Json Library

MongoDb – Version 2

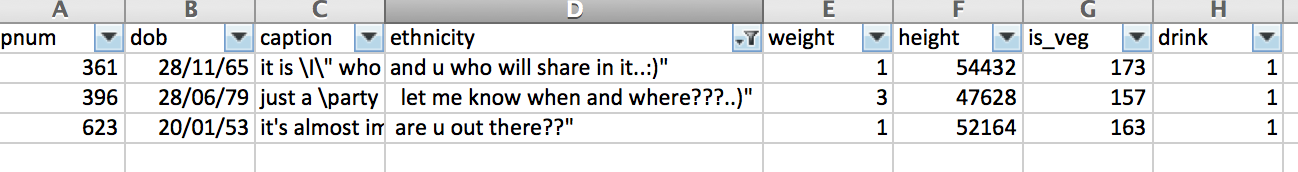
Codebeautify – To convert the given CSV file to Json format

**Modifications made to the existing data(CSV) file shared:**

1. I added another row in CSV file because the first row is used by codebeautyify to form key in JSON.

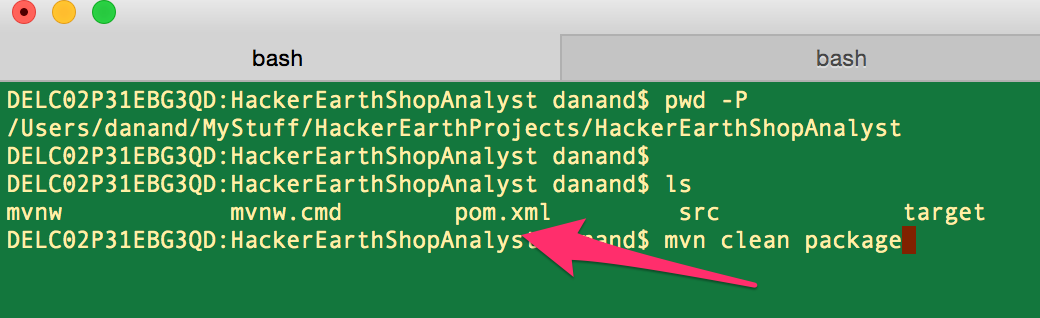
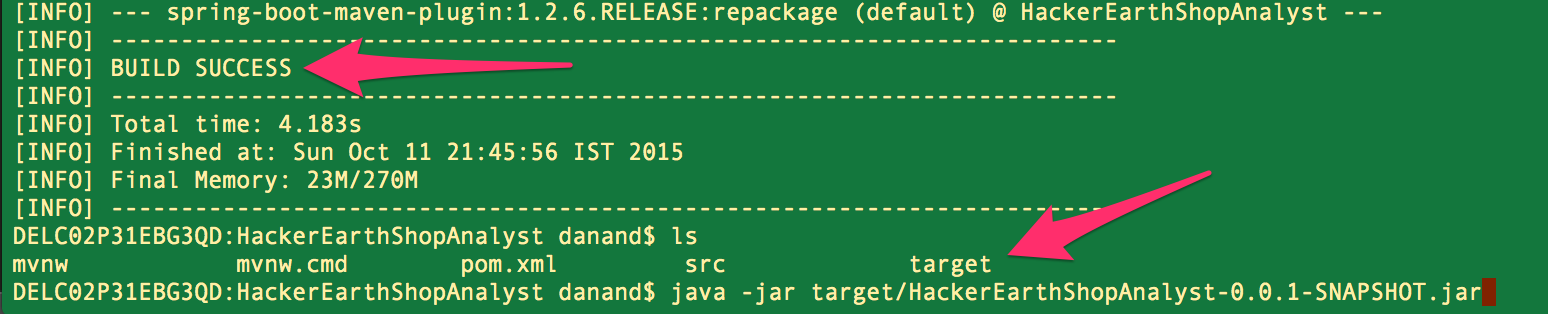


1. One of the CSV row had ethnicity as a string rather than integer value unlike other rows. So rather than changing that ethnicity to int I made sure in code that if ethnicity length is greater than 1 then it should be treated as -1.

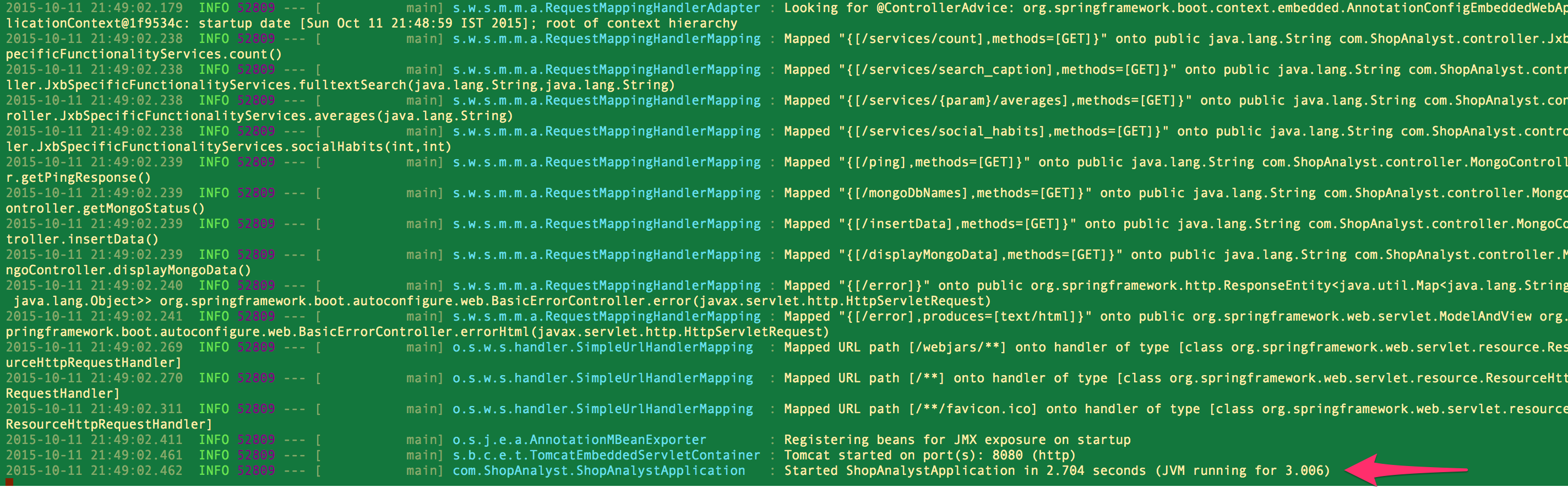


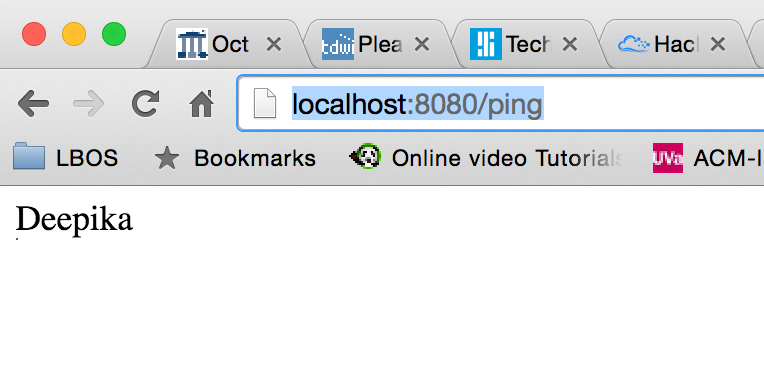
**How to run the code.**

Copy the code to folder

1. Run mvn clean ackage 
2. After succesful build. Run : java -jar target/HackerEarthShopAnalyst-0.0.1-SNAPSHOT.jar

Spring boot will instantiate its own tomcat. **Make sure you don’t have any tomcat running at port 8080.**



To confirm go to : [http://localhost:8080/ping](http://localhost:8080/ping#)

NOTE: I have attached .jar file in the target folder itself so you may also just run java -jar target/HackerEarthShopAnalyst-0.0.1-SNAPSHOT.jar if the dependencies are satisfied.

**APIs implemented:**

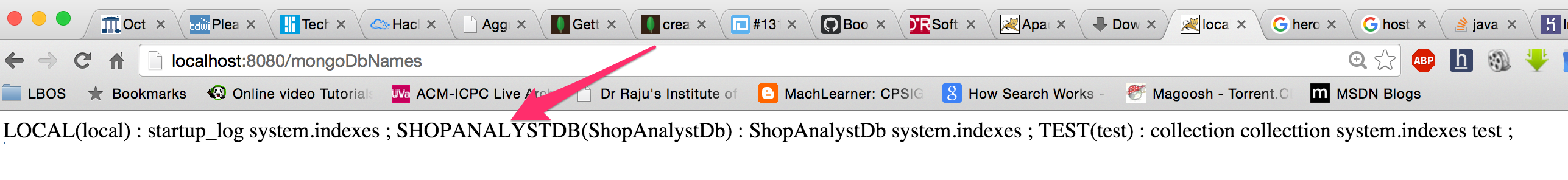
To load the data into MongoDB

1. Make sure mongoDb is installed on your system. Go to terminal and type “mongod” without quotes. This will start mongo db on your local machine.

First see the existing Database and collectionName

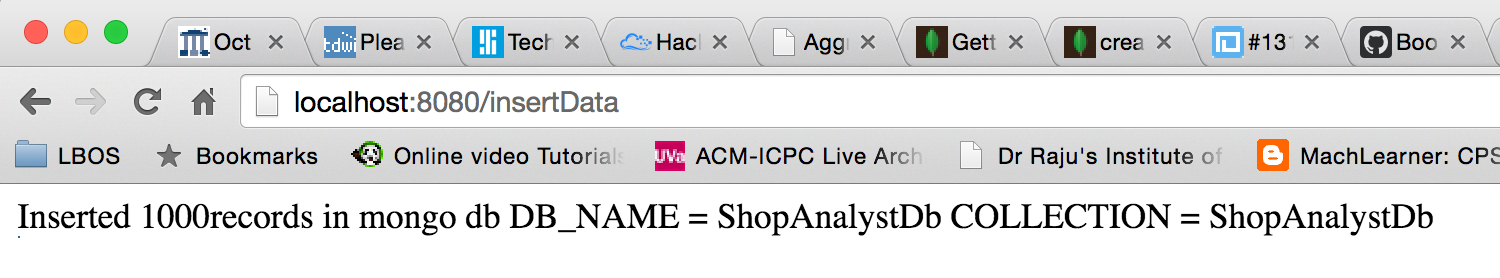
I have used **ShopAnalystDb** as database and “**ShopAnalystDb**” as collection

Use: http://localhost:8080/mongoDbNames



1. TO insert data

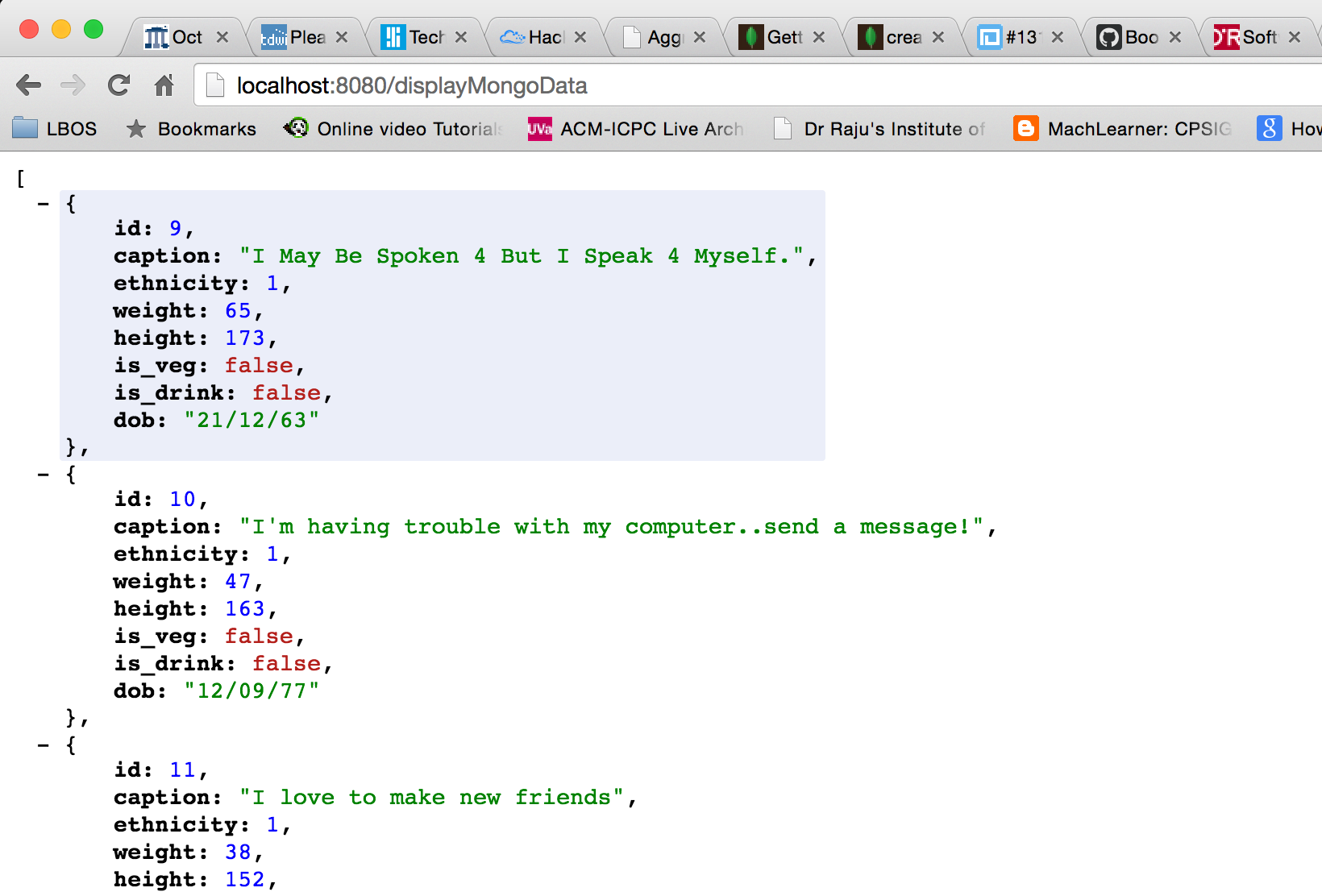
Use: <http://localhost:8080/insertData>

This will first delete the data and then insert 1000 rows in mongo. So use it as many times.

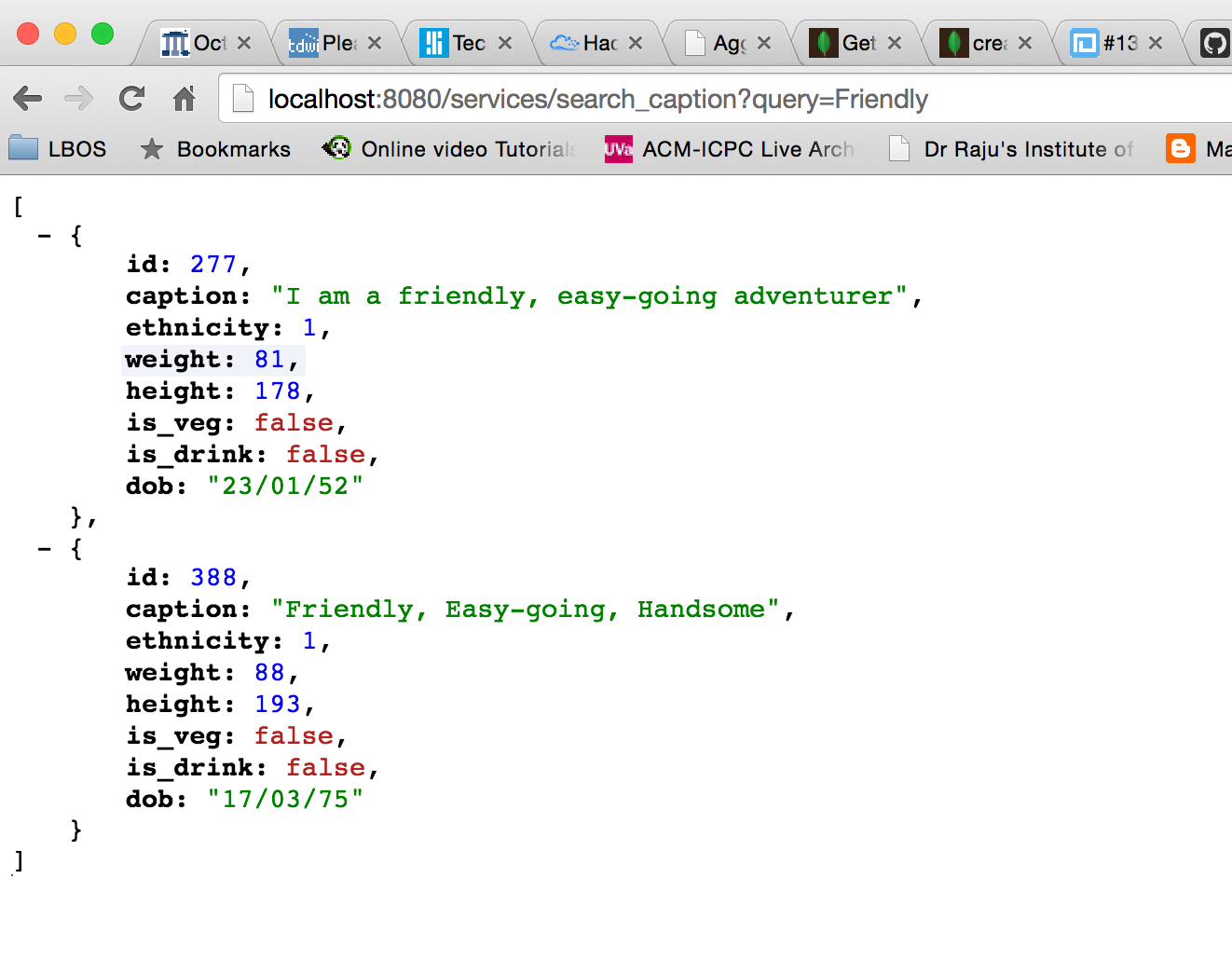
**The location of data1000.json file is in the source code resource folder**

1. View data : <http://localhost:8080/displayMongoData>

I have Json Viewer chrome extension installed on my system.



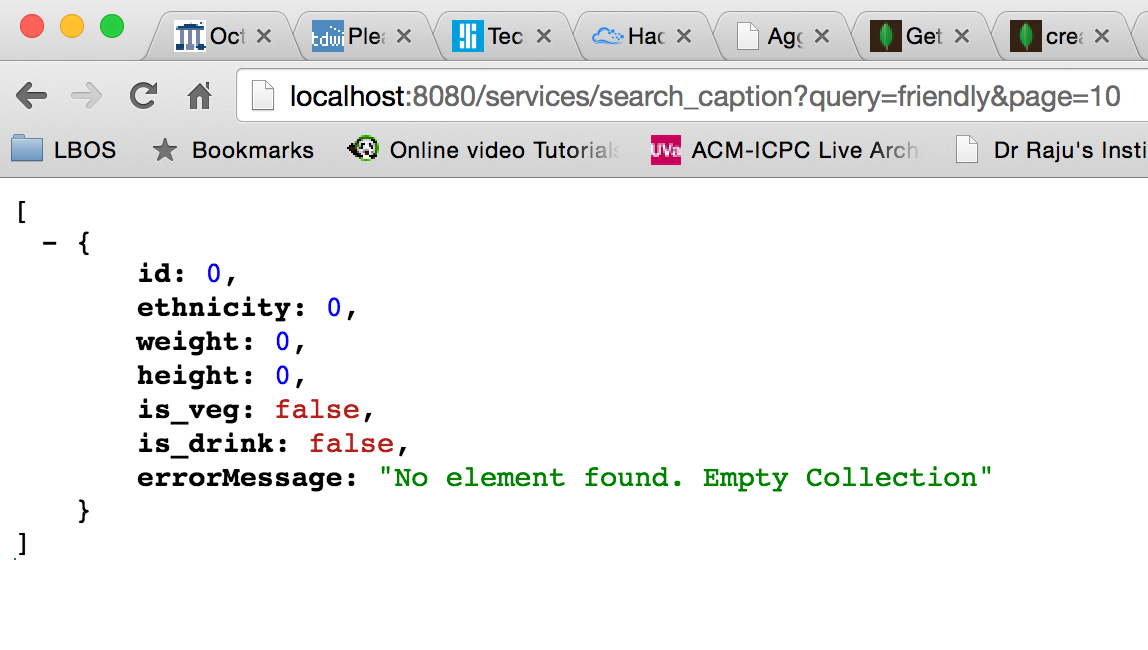
1. Search (case insensitive)

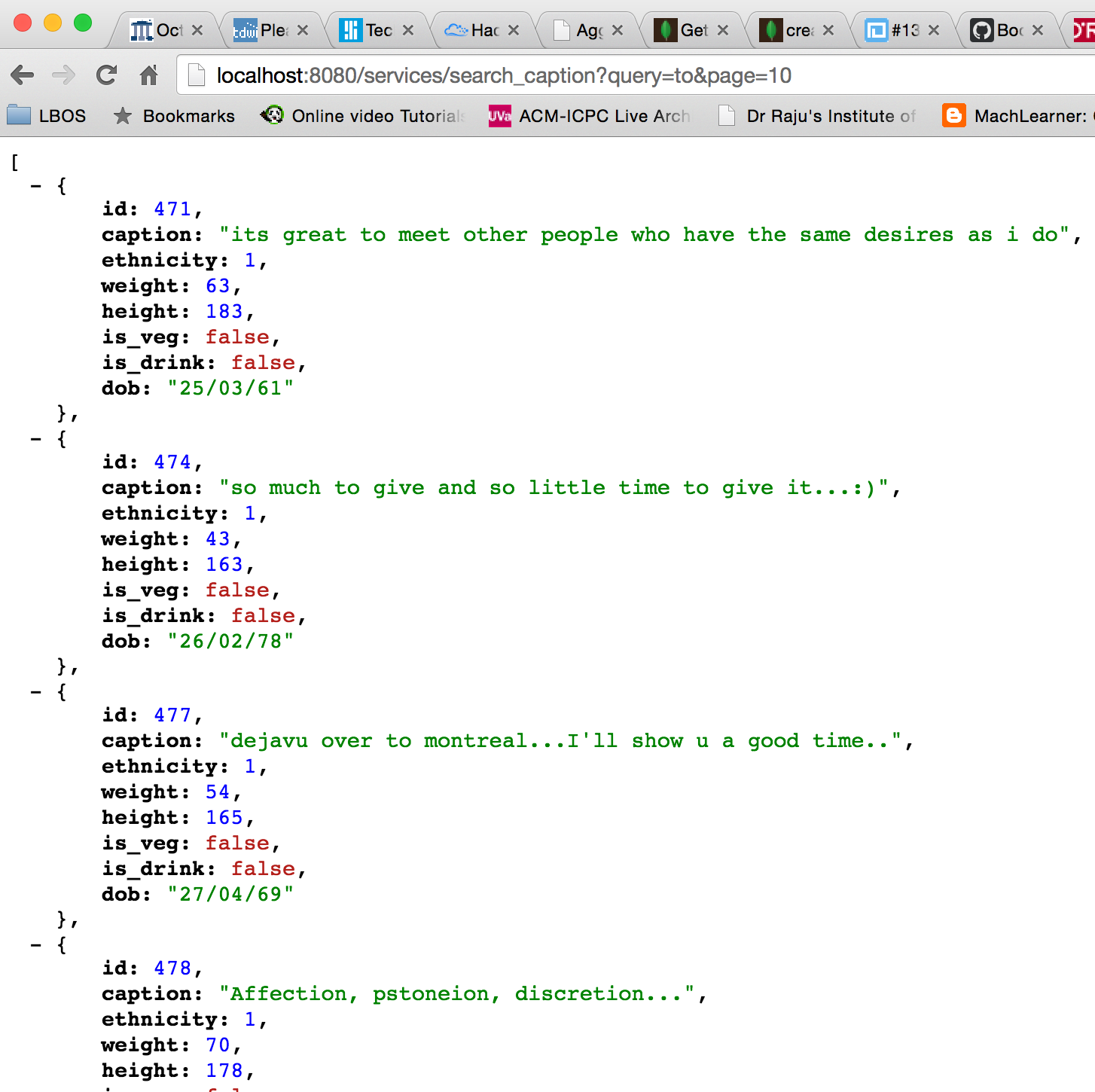
http://localhost:8080/services/search\_caption?query=Friendly 

1. Pagination(optional param “page”)

http://localhost:8080/services/search\_caption?query=to&page=10

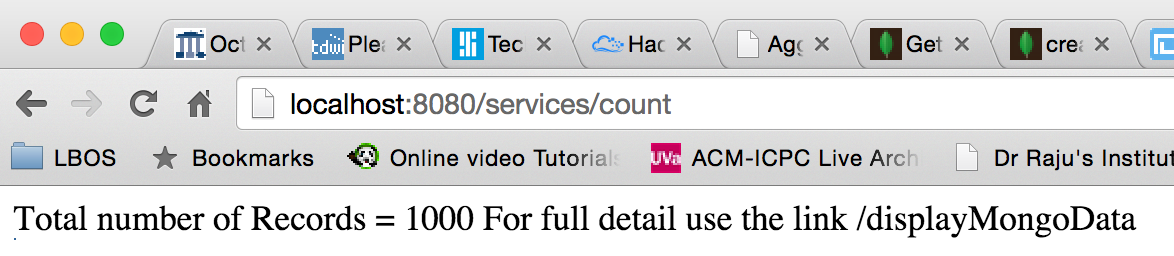
In case sufficient number of records are not present then response will look like:





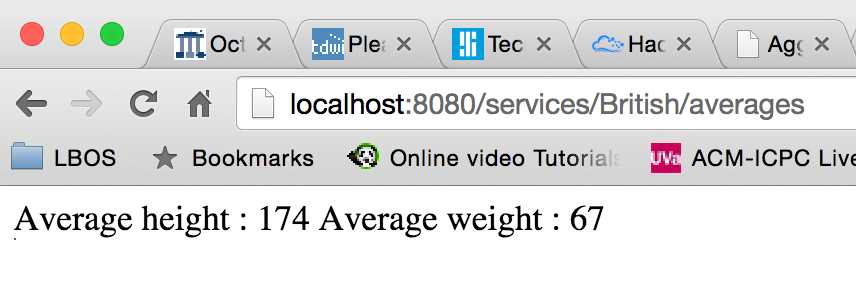
1. Count API

http://localhost:8080/services/count



1. Average according to ethnicity

Sample: <http://localhost:8080/services/British/averages> In general : http://localhost:8080/services/{Ethnicity}/averages



1. Social behaviour :

<http://localhost:8080/services/social_habits?vegetarian=0&drink=0>

