PRICE SEARCHER
DATA SCIENCE TASK

JAVA, SPRING, ALGORITHMS, DATA STRUCTURES

By

Obinna Onyimadu

obinna240@yahoo.co.uk
07805117769

2018

Contents

Chapter 1

Chapter 1

1.1 Approach Overview

The application was built as a SpringBoot App. Additionally, the java version used was 1.8. Logging and testing was carried out using log4j 2 and Junit. The port 8099 is set in the properties file.

1.2 Solutions

You can run the entire application as a rest service.

1.2.1 Dbpedia

To run the dbpedia problem, you can call: `localhost:8099/service/name/cameron` to get the place where David Cameron was born or `localhost:8099/service/name/blair` to identify Tony Blair's date of birth.

1.2.2 Revenue Calculator

Revenue Calculator is implemented in `com.semanticIntegration.model` and the interface can be found in `com.semanticIntegration.interfaces`. To run this as a REST call, do localhost:8099/service/revenueCalc/{marginPercentage}/{costOfGoods}. Note that the implementation also supports entering String values to estimate revenue. You can thus do `localhost:8099/service/revenueCalc/20/400`.

1.2.3 Pagination

Pagination classes is spread across the entire application in packages `utils` and `objects`. You can do a call to localhost:8099/service/pagination/{search}. The {search} parameter variable is optional and regardless of whatever string you enter, it will return the same results. We have used some dummy entries to get it working. You can see a sample of the returned json in the submission.

Chapter 2

How to Run

1. You can copy the entire project into a directory of your choosing and using maven do > `mvn clean compile exec:java`

2. Alternatively, you can take the jar file and