

|  |
| --- |
| Documentation@Turing Back-end Challenge  2018 |
|  |
| July 10  Individual  Authored by: Kowshik Dutta |

# Overview

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subtitle Text Here A set of APIs are built and deployed as per Turing’s backend challenge documentation guide. This document describes the set-up required to deploy and run the services.  In-Scope  Scope of the document is as per the below attachment:   |  |  | | --- | --- | | GET ALL DEPARTMENTS | GET A CUSTOMER BY ID | | GET A SINGLE DEPARTMENT | UPDATE CUSTOMER DETAILS | | GET ALL CATEGORIES | UPDATE CUSTOMER ADDRESS | | GET A SINGLE CATEGORY | UPDATE CUSTOMER CREDIT CARD | | GET PRODUCT CATEGORY | CREATE AN ORDER | | GET ALL CATEGORIES IN A DEPARTMENT | GET AN ORDER | | GET ALL ATTRIBUTES | GET CUSTOMERS ORDER | | GET SINGLE ATTRIBUTES | GET ORDER SHORT DETAILS | | GET ALL ATTRIBUTE VALUES IN AN ATTRIBUTE | GENERATE CART UNIQUE ID | | GET ALL ATTRIBUTES OF A PRODUCT | ADD PRODUCT TO SHOPPING CART | | GET ALL PRODUCTS | GET LIST OF PRODUCTS IN A SHOPPING CART | | SEARCH PRODUCTS | UPDATE CART ITEM QUANTITY | | GET A SINGLE PRODUCT | EMPTY SHOPPING CART | | GET ALL PRODUCTS IN A CATEGORY | REMOVE ITEM FROM SHOPPING CART | | GET ALL PRODUCTS IN A DEPARTMENT | GET ALL TAXES | | GET REVIEWS OF A PRODUCT | GET A SINGLE TAX | | POST A PRODUCT REVIEW | GET ALL SHIPPING REGIONS | | CREATE A NEW CUSTOMER | GET ALL SHIPPINGS IN A REGION | | LOGIN A CUSTOMER | POST PAYMENT TO STRIPE | |  | REQUEST HEADER TOKEN FORMAT | |  |  | |  |  |   Out of Scope  **I am unable to complete the below 2 services**   |  | | --- | | FACEBOOK LOGIN | | STRIPE WEBHOOK |    Environment Set-up MySQL set-up:  Create a data-base called “turing”  Create a user called “turing” with password “***abcd1234@”***  ***Give them full access to DB “turing”.***  CREATE DATABASE IF NOT EXISTS turing;  CREATE USER 'turing'@'localhost' IDENTIFIED BY 'abcd1234@';  GRANT ALL PRIVILEGES ON turing.\* TO 'turing'@'localhost';  GRANT SELECT ON mysql.proc TO 'turing'@'localhost';   * ***You may create any user and DB of your choice. Please make the necessary changes to the “context.xml” in the turing.war file***   Start your MySQL DB.  Docker set-up:  Please create a folder called “turing”. Place the “Dockerfile” and “turing.war” in the same folder. Next build and run the docker file as below:  sudo docker build -f Dockerfile -t turingimage .  sudo docker run --rm --network="host" -d turingimage  To check the docker container:  sudo docker ps  Now you can call the service. Please view “Testing the Service” for greater details |
| Design Basic design of the services are as under |

Any user request will first hit the “SecurityManager” filter in “turing” application running in tomcat.

SecurityManager then sends the request to the requested service

Each service has it’s own DataAccessObject ( Dao ). Dao classes talks to the MySQL server running on Host

## Logging:

Log4j has been used for logging. All logs are written to a file called “Application.log” inside /var/turing/log/Application.log

This setting can be changes by changing the “log4j.appender.file.File” parameter in the log4j.properties file under “Resources” folder in turing.war file

## Other Parameters:

Login timeout is set to 10 mins from login time. This is hardcoded in the “Token.java” in side “com.turing.security” package

Below parameters are also set in “web.xml” file in turing.war

<context-param>

<param-name>stripeSecretKey</param-name>

<param-value>sk\_test\_lomdOfxbm7QDgZWvR82UhV6D</param-value>

</context-param>

<context-param>

<param-name>pagesize</param-name>

<param-value>10</param-value>

</context-param>

stripeSecretKey is used to call Stripe. Whereas, pagesize is used to limit rows in getting products from application (paging).