**SHOP LOCATOR APPLICATION**

The Shop Locator Application exposes following two RESTful APIs.

1. **Create Shop Detail API :** This POST API available to Retail Manager allows to save Shop Details with Geocoded Location Info, to an in memory array/List, with minimal concurrency features provided by Java Concurrent Collections.

Though the Google Geocoding API is used, the application supports plugging in another third party Geocoding API in future, by introducing another GeoCodingDelegateImpl Class which implements the GenericGeoCodingDelegate Interface. This can then be injected into the façade that prepares the Model to be persisted.

1. **Get Closest Shop API:** This GET API available to the Customer allows him to pass his Location coordinates as query-string parameters, and retrieve the Shop Details of the shop closest to his location.

The application leverages various spring features like the following:

1. Aspect based logging for entry and exit points of controller and service methods.
2. Pluggable Conversion Service to seamlessly convert between Request DTO and Model Objects.
3. Pluggable Validator components to validate data both at Controller and Service Layers.

and many more…

The following pages demonstrate how to build and deploy the application and get the REST APIs up and running.

**Installation Guide**

Maven is used as the build runner for the application.

To perform build, run the following command from within the application folder

* **mvn clean install**

This will generate a .war file in the target directory of the application folder.

Deploy this war in any J2EE compliant Server. The application has been tested on Apache Tomcat 7.

Once this is done, the Application should be up and running and you can invoke the APIs from a Rest Client Browser Extension. DHC Rest Extension for Chrome was used to test the APIs, but any other extension like Postman, should work equally well.

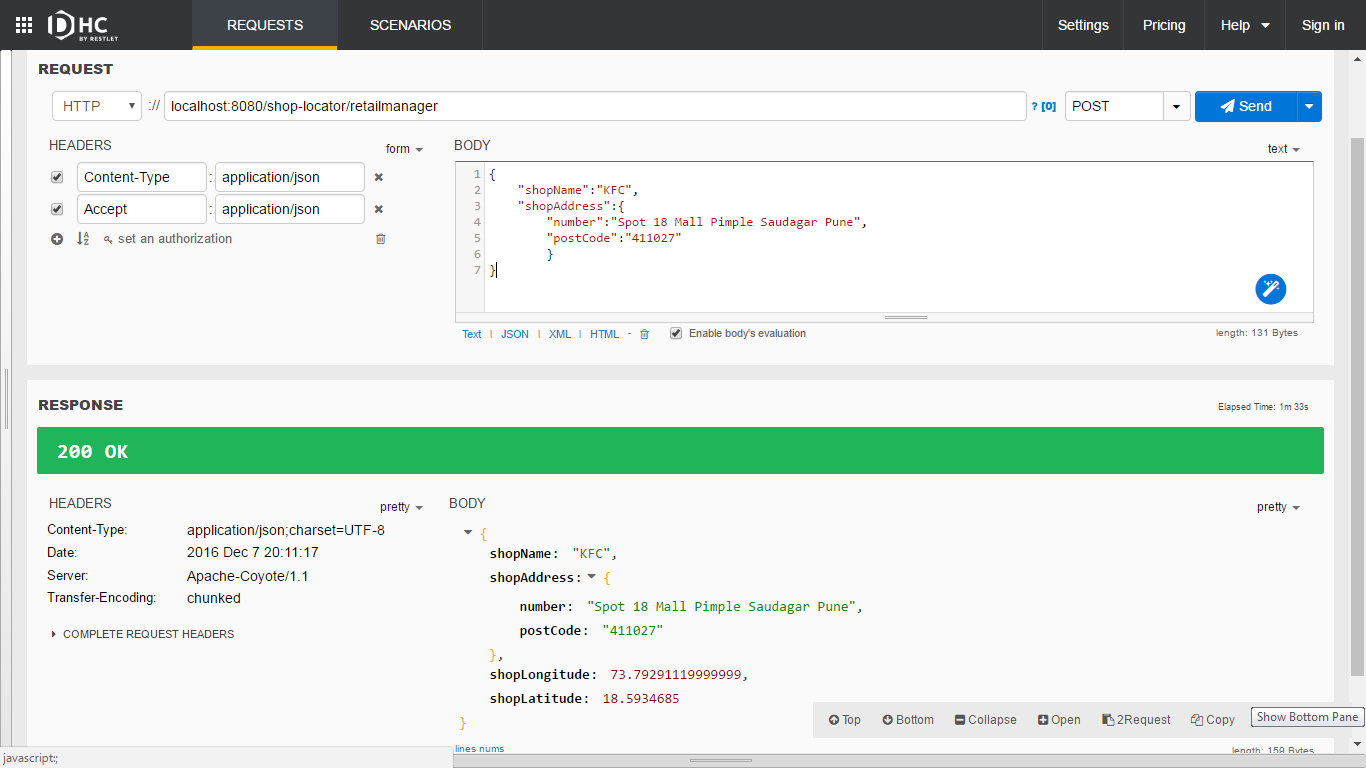
The API Documentation that follows will guide you how to invoke the APIs through the Rest Client Browser extension of your choice.

**API DOCUMENTATION**

**CreateShopDetail RetailManager API**:

Method: POST

Please refer the below screenshot demonstrating the usage of this API. It shows the Request and Response JSON formats, Request Headers, The Rest API URL and other minute details. Please note that currently only JSON inputs and outputs are supported for now. The application is fully extensible though, to support XML format (by adding the relevant Message Converters to the Spring Configuration).



**GetClosestShop Customer API**:

Method: GET

Please refer the below screenshot demonstrating the usage of this API. It shows the Request and Response JSON formats, Request Headers, The Rest API URL and other minute details. Please note that currently only JSON inputs and outputs are supported for now. The application is fully extensible though, to support XML format (by adding the relevant Message Converters to the Spring Configuration).

