**REST Api For to perform CRUD Operations**

Tools and Environment

|  |  |
| --- | --- |
| Java | 1.8.0.152 |
| Tomcat | 7 |
| Project | Maven |
| Jersey REST API | 2.17 |
| Log4j | 1.2.17 |
|  |  |

**Design and Architecture**

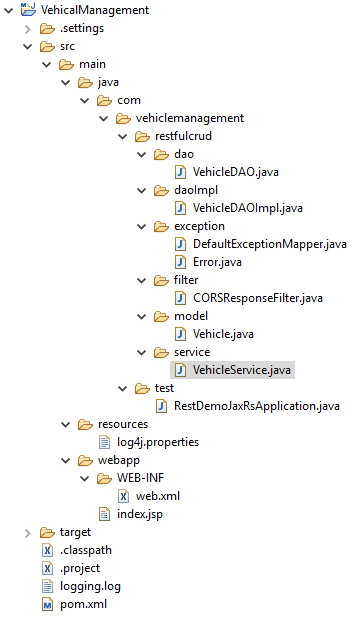
Following are the few points which are considered for developing the web services

1. Versioning : v1 is used in the url to specify the version of the api
2. Result filtering and searching is provided based on Id/Year & Make/ Year & Model / Make and Model
3. Updates & creation return a resource representation To prevent an API consumer from having to hit the API again for an updated representation
4. JSON and XML Response are supported
5. Supports CORS Request
6. Service layer is separated from the DAO Layer
7. Exposing DAO Interface to service layer , DAO Impl can be done differently
8. HashMap collection is used for instore memory storage.
9. Proper HTTP status codes are used to return to the API,Which helps the consumer to understand the response and act accordingly
10. Log4j is used for the logging
11. Exception Handling is done implementing ExceptionMapper. All Exceptions are sent back with proper status code and Message

RESTful service implement the following routes:

|  |  |  |
| --- | --- | --- |
| Service | Description | Supports |
| GET vehicles | Get all Vehicle from the system | APPLICATION\_JSON , APPLICATION\_XML |
| GET vehicles/{id} | Get Vehicle Object with id | APPLICATION\_JSON , APPLICATION\_XML |
| GET Vehicles/Year/{Year}/Model/{Model} | Get all Vehicle for a gives Year and Model | APPLICATION\_JSON , APPLICATION\_XML |
| GET Vehicles/Year/{Year}/Make/{Make} | Get all Vehicle for a gives Year and Make | APPLICATION\_JSON , APPLICATION\_XML |
| GET Vehicles/Make/{Make}/Model/{Model} | Get all Vehicle for a gives Make and Model | APPLICATION\_JSON , APPLICATION\_XML |
| POST vehicles | Create a new Vehicle | APPLICATION\_JSON , APPLICATION\_XML |
| PUT vehicles | Update a Existing Vehicle | APPLICATION\_JSON , APPLICATION\_XML |
| DELETE vehicles/{id} | Delete Vehicle with id |  |

Project Hierarchy



**Class and Description**

|  |  |
| --- | --- |
| **Class** | **Description** |
| Vehicle | Model bean class |
| VehicleDAO | Interface for the Data Access Operation |
| VehicleDAOImpl | Implementaion of DAO Interface |
| DefaultExceptionMapper | Mapping of all the exception and providing proper Error Messgae |
| Error | Model bean class for exception |
| CORSResposeFilter | Used to support CORS |
| VehicleService | Main Rest Service class to implement rest calls |

**Validations Applied**

1. Year should be between 1950 and 2050
2. Make should not be null or empty
3. Model should not be null or empty