Discovery

DHS – Offsite Java Assignment

11th October 2017

OVERVIEW

The given assignment can be achieved in many patterns. Since, the assignment is to assess the capacity of individual on web services, rest services and integration, I chose to use the implement the service oriented architecture. Hence, 3 individual applications were created to accomplish the task and can be deployed anywhere and easy to integrate each other. They are named as below:

- harika-dindu-domain, for persisting the graph into database and exposing as rest service. Used JPA with Hibernate implementation so that implementation can be changed without changing the code.
    - Note: I used jdk 8 intentionally to take advantage of default methods. Due to the advantage that they can be provided to an interface without affecting implementing classes as it includes an implementation. If each added method in an interface defined with implementation then no implementing class is affected. An implementing class can override the default implementation provided by the interface. **This is just for the variety of implementation and can be implemented as our traditional approach.**
- Harika-dindu-service, for implementing the shortest path algorithm and connecting to the rest service and exposing the service via SOAP webservice. and
- harika-dindu-web(for user interface).

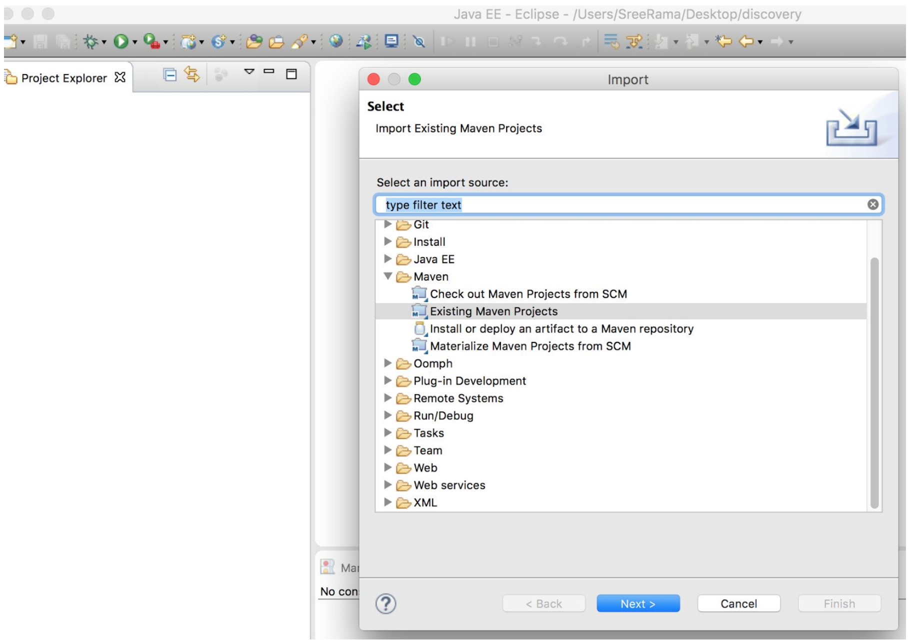
The class names and method names explains the purpose of their implementation. The interceptors are configured for logging the client and server logs of web-services, class files explains more details. Since, i am using tomcat server, i have redirected the log files to my server and can be changed according to the requirement. Since, this is an assignment, Only basic logging and exception handling and were implemented, AOP can be implemented for logging. Maven dependencies can also be improved to support modularity.

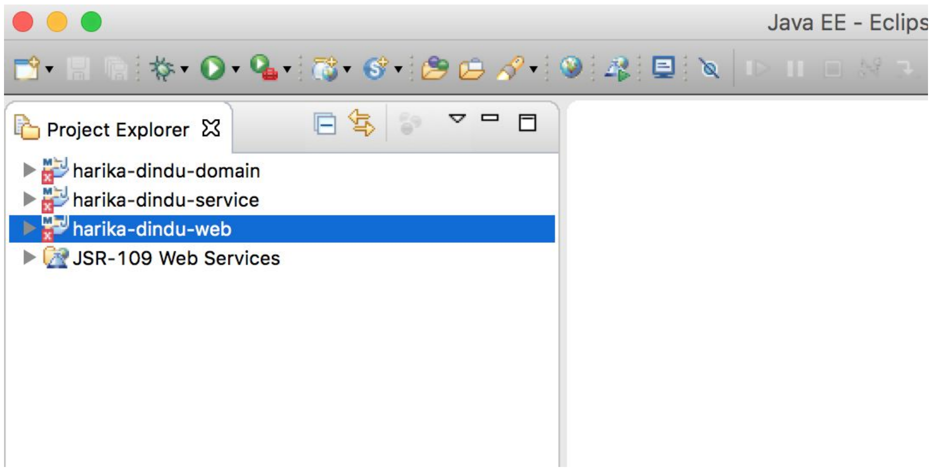
Note: Due to time constraint, i could not include mock testing. I have only created test cases only for integration testing. In simple terms, it can be improved in all aspects based on our requirements

SETUP

After checking out the code, the projects can be compiled either via command prompt or using IDE using maven. There is no specific server plugin configured due to time constraint, hence the war files can be either deployed from the server console or from IDE.

Importing the files into workspace.

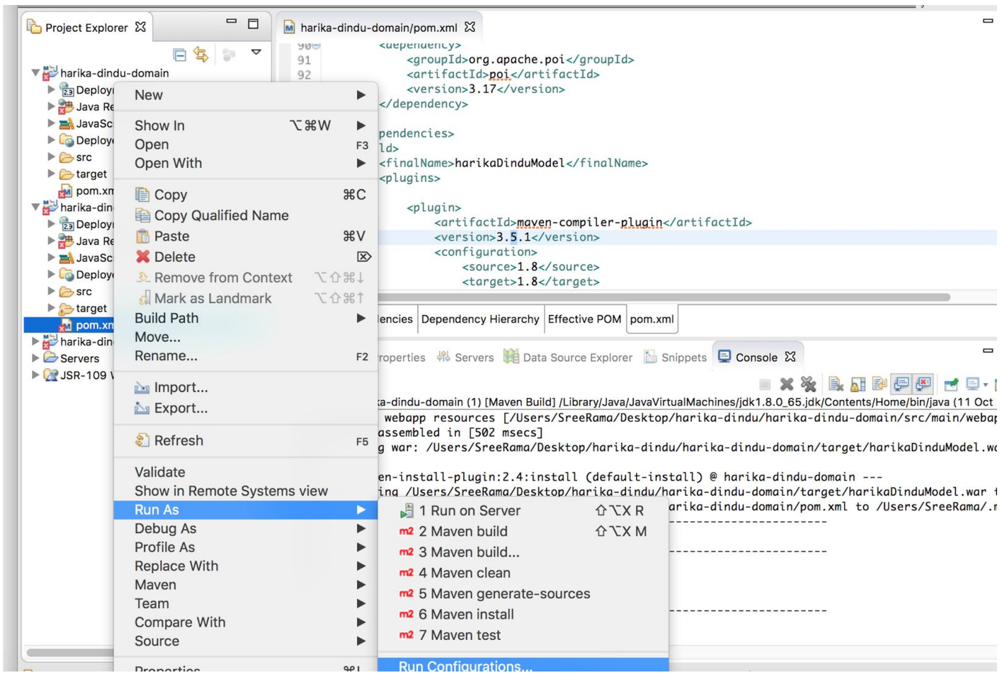




The applications should be deployed in the below specific order.

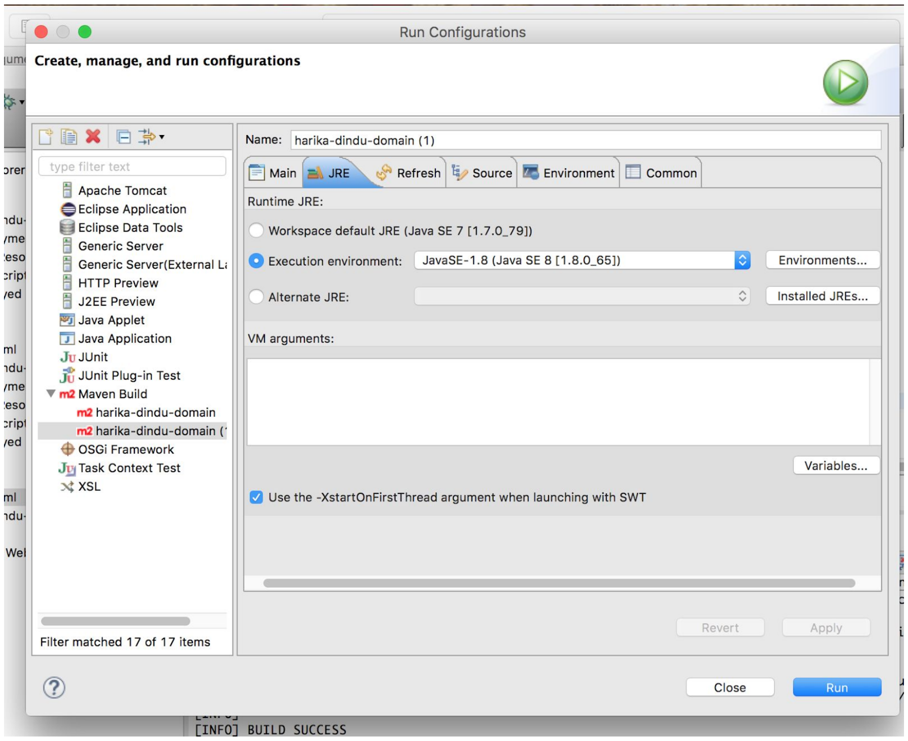
- ## harika-dindu-domain

Since the applications are based on JDK 1.8, just making sure that the server is using the exact runtime. If not change the execution environment to 1.8 as shown below.



```
        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi</artifactId>
            <version>3.17</version>
        </dependency>

pendencies>
ld>
<finalName>harikaDinduModel</finalName>
<plugins>

    <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.5.1</version>
        <configuration>
            <source>1.8</source>
            <target>1.8</target>
```

```
a-dindu-domain (1) [Maven Build] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (11 Oct
 webapp resources [/Users/SreeRama/Desktop/harika-dindu/harika-dindu-domain/src/main/webap
assembled in [502 msecs]
g war: /Users/SreeRama/Desktop/harika-dindu/harika-dindu-domain/target/harikaDinduModel.wa

en-install-plugin:2.4:install (default-install) @ harika-dindu-domain ---
ing /Users/SreeRama/Desktop/harika-dindu/harika-dindu-domain/target/harikaDinduModel.war t
                                        arika-dindu-domain/pom.xml to /Users/SreeRama/.r
                                        -----------------------
                                        -----------------------
                                        -----------------------
```



Name: harika-dindu-domain (1)

Runtime JRE:

Workspace default JRE (Java SE 7 [1.7.0_79])

[✓ Execution environment: JavaSE-1.8 (Java SE 8 [1.8.0_65])]

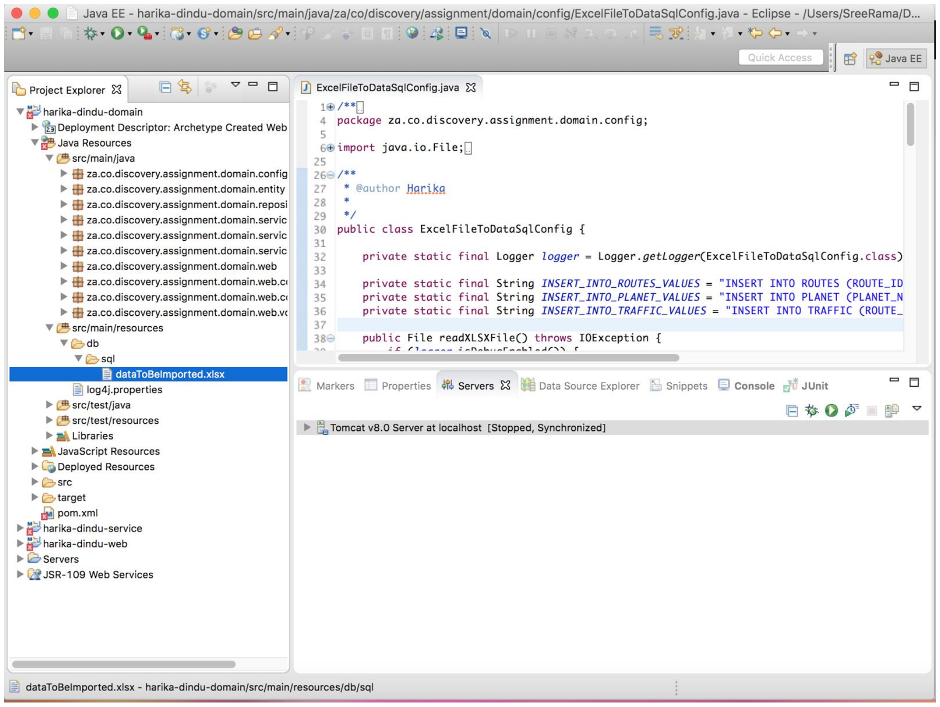
Alternate JRE:

VM arguments:

✓ Use the -XstartOnFirstThread argument when launching with SWT

```
[INFO] BUILD SUCCESS
```

The excel data file is already inside the resource folder of application. Therefore no need to do any extra data conversions.



Java EE - harika-dindu-domain/src/main/java/za/co/discovery/assignment/domain/config/ExcelFileToDataSqlConfig.java - Eclipse - /Users/SreeRama/D...

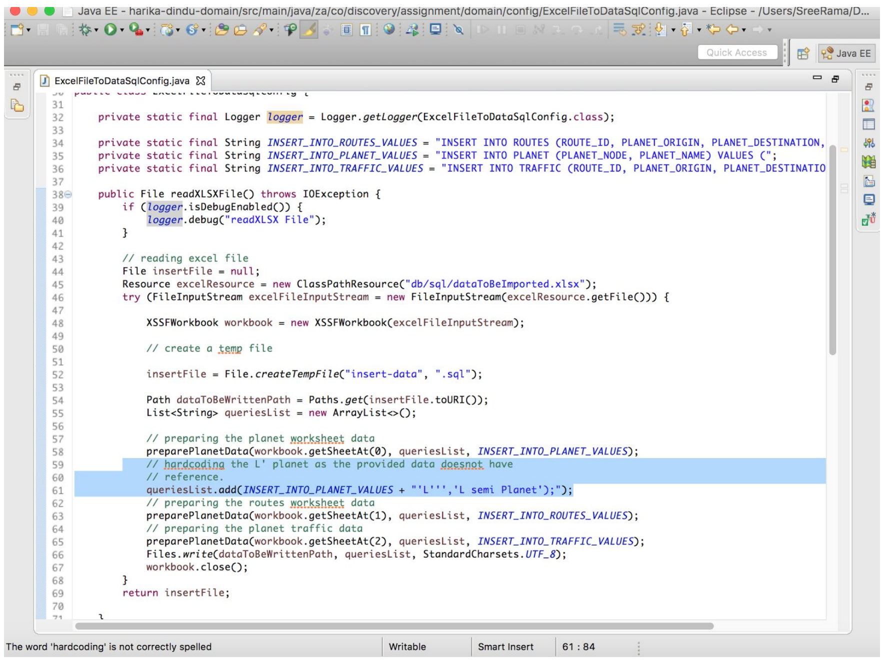
```
harika-dindu-domain
  Deployment Descriptor: Archetype Created Web
  Java Resources
    src/main/java
      za.co.discovery.assignment.domain.config
      za.co.discovery.assignment.domain.entity
      za.co.discovery.assignment.domain.reposi
      za.co.discovery.assignment.domain.servic
      za.co.discovery.assignment.domain.servic
      za.co.discovery.assignment.domain.servic
      za.co.discovery.assignment.domain.web
      za.co.discovery.assignment.domain.web.c
      za.co.discovery.assignment.domain.web.c
      za.co.discovery.assignment.domain.web.v
    src/main/resources
      db
        sql
          dataToBeImported.xlsx
      log4j.properties
    src/test/java
    src/test/resources
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  pom.xml
harika-dindu-service
harika-dindu-web
Servers
JSR-109 Web Services
```

```java
/**
package za.co.discovery.assignment.domain.config;

import java.io.File;

/**
 * @author Harika
 *
 */
public class ExcelFileToDataSqlConfig {

    private static final Logger logger = Logger.getLogger(ExcelFileToDataSqlConfig.class)

    private static final String INSERT_INTO_ROUTES_VALUES = "INSERT INTO ROUTES (ROUTE_ID
    private static final String INSERT_INTO_PLANET_VALUES = "INSERT INTO PLANET (PLANET_N
    private static final String INSERT_INTO_TRAFFIC_VALUES = "INSERT INTO TRAFFIC (ROUTE_

    public File readXLSXFile() throws IOException {
```

```
Tomcat v8.0 Server at localhost  [Stopped, Synchronized]
```

dataToBeImported.xlsx - harika-dindu-domain/src/main/resources/db/sql

Note: I have added the "L'" planet information to Planets table, as the Route and Planet entities have relation and routes data was failing to import as the Planet data does not have L' planet in it's data. The entity files explains the relations in detail.

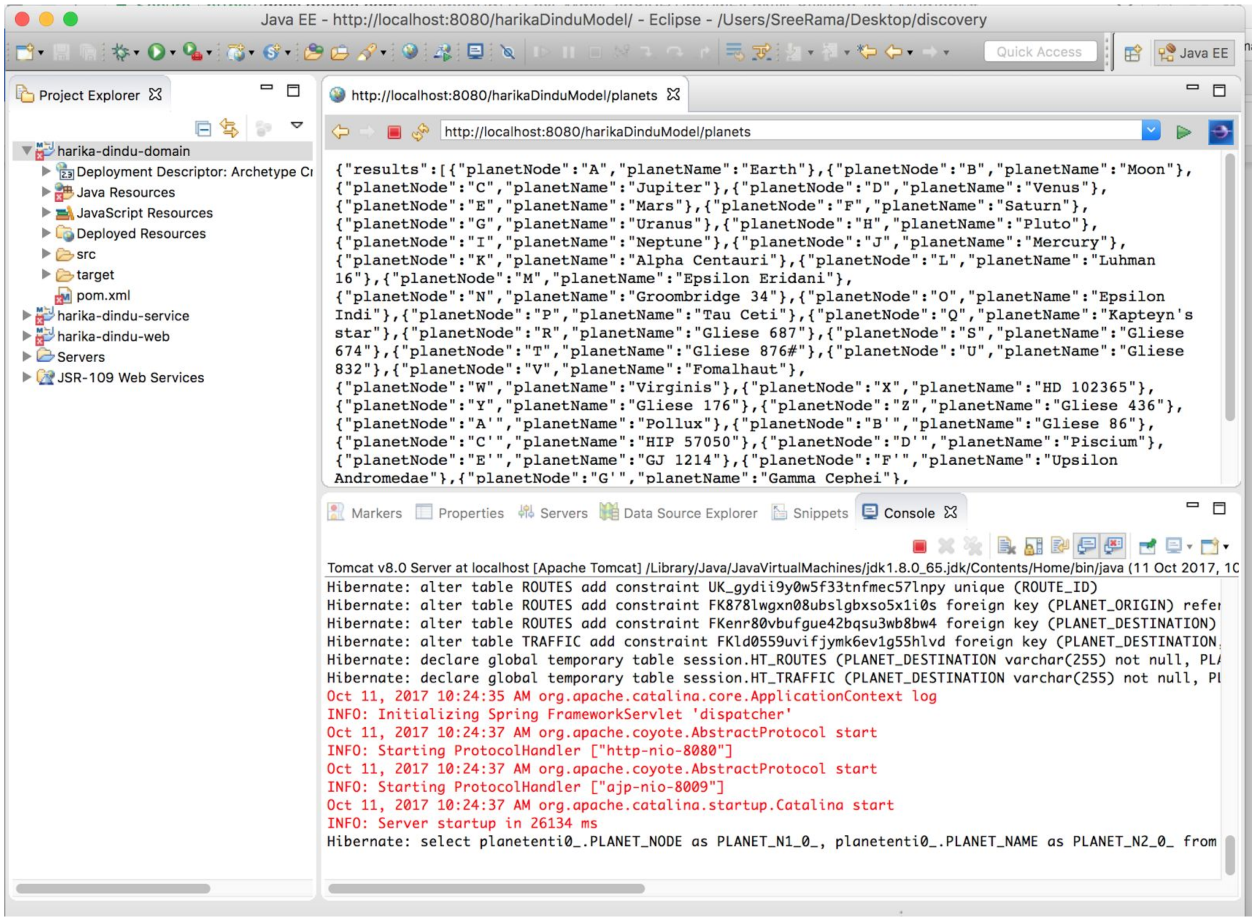


Java EE - harika-dindu-domain/src/main/java/za/co/discovery/assignment/domain/config/ExcelFileToDataSqlConfig.java - Eclipse - /Users/SreeRama/D...

```java

    private static final Logger logger = Logger.getLogger(ExcelFileToDataSqlConfig.class);

    private static final String INSERT_INTO_ROUTES_VALUES = "INSERT INTO ROUTES (ROUTE_ID, PLANET_ORIGIN, PLANET_DESTINATION,
    private static final String INSERT_INTO_PLANET_VALUES = "INSERT INTO PLANET (PLANET_NODE, PLANET_NAME) VALUES (";
    private static final String INSERT_INTO_TRAFFIC_VALUES = "INSERT INTO TRAFFIC (ROUTE_ID, PLANET_ORIGIN, PLANET_DESTINATIO

    public File readXLSXFile() throws IOException {
        if (logger.isDebugEnabled()) {
            logger.debug("readXLSX File");
        }

        // reading excel file
        File insertFile = null;
        Resource excelResource = new ClassPathResource("db/sql/dataToBeImported.xlsx");
        try (FileInputStream excelFileInputStream = new FileInputStream(excelResource.getFile())) {

            XSSFWorkbook workbook = new XSSFWorkbook(excelFileInputStream);

            // create a temp file

            insertFile = File.createTempFile("insert-data", ".sql");

            Path dataToBeWrittenPath = Paths.get(insertFile.toURI());
            List<String> queriesList = new ArrayList<>();

            // preparing the planet worksheet data
            preparePlanetData(workbook.getSheetAt(0), queriesList, INSERT_INTO_PLANET_VALUES);
            // hardcoding the L' planet as the provided data doesnot have
            // reference.
            queriesList.add(INSERT_INTO_PLANET_VALUES + "'L''','L semi Planet');");
            // preparing the routes worksheet data
            preparePlanetData(workbook.getSheetAt(1), queriesList, INSERT_INTO_ROUTES_VALUES);
            // preparing the planet traffic data
            preparePlanetData(workbook.getSheetAt(2), queriesList, INSERT_INTO_TRAFFIC_VALUES);
            Files.write(dataToBeWrittenPath, queriesList, StandardCharsets.UTF_8);
            workbook.close();
        }
        return insertFile;

```

The word 'hardcoding' is not correctly spelled | Writable | Smart Insert | 61 : 84

Domain application can be accessed as below. This is my localhost and can be accessed according to server.



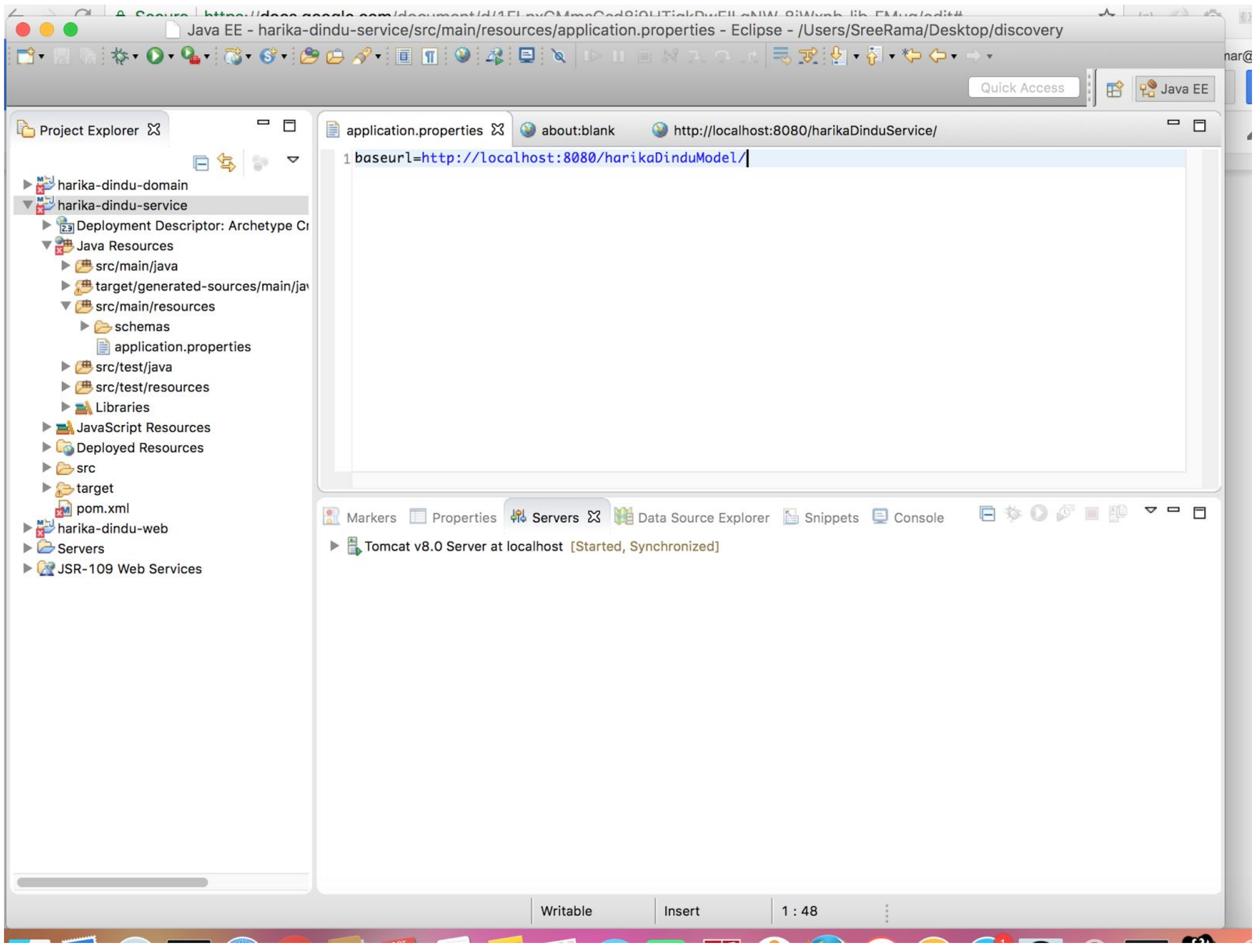
http://localhost:8080/harikaDinduModel/planets

```
{"results":[{"planetNode":"A","planetName":"Earth"},{"planetNode":"B","planetName":"Moon"},
{"planetNode":"C","planetName":"Jupiter"},{"planetNode":"D","planetName":"Venus"},
{"planetNode":"E","planetName":"Mars"},{"planetNode":"F","planetName":"Saturn"},
{"planetNode":"G","planetName":"Uranus"},{"planetNode":"H","planetName":"Pluto"},
{"planetNode":"I","planetName":"Neptune"},{"planetNode":"J","planetName":"Mercury"},
{"planetNode":"K","planetName":"Alpha Centauri"},{"planetNode":"L","planetName":"Luhman
16"},{"planetNode":"M","planetName":"Epsilon Eridani"},
{"planetNode":"N","planetName":"Groombridge 34"},{"planetNode":"O","planetName":"Epsilon
Indi"},{"planetNode":"P","planetName":"Tau Ceti"},{"planetNode":"Q","planetName":"Kapteyn's
star"},{"planetNode":"R","planetName":"Gliese 687"},{"planetNode":"S","planetName":"Gliese
674"},{"planetNode":"T","planetName":"Gliese 876#"},{"planetNode":"U","planetName":"Gliese
832"},{"planetNode":"V","planetName":"Fomalhaut"},
{"planetNode":"W","planetName":"Virginis"},{"planetNode":"X","planetName":"HD 102365"},
{"planetNode":"Y","planetName":"Gliese 176"},{"planetNode":"Z","planetName":"Gliese 436"},
{"planetNode":"A'","planetName":"Pollux"},{"planetNode":"B'","planetName":"Gliese 86"},
{"planetNode":"C'","planetName":"HIP 57050"},{"planetNode":"D'","planetName":"Piscium"},
{"planetNode":"E'","planetName":"GJ 1214"},{"planetNode":"F'","planetName":"Upsilon
Andromedae"},{"planetNode":"G'","planetName":"Gamma Cephei"},
```

```
Tomcat v8.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (11 Oct 2017, 1
Hibernate: alter table ROUTES add constraint UK_gydii9y0w5f33tnfmec57lnpy unique (ROUTE_ID)
Hibernate: alter table ROUTES add constraint FK878lwgxn08ubslgbxso5x1i0s foreign key (PLANET_ORIGIN) refe
Hibernate: alter table ROUTES add constraint FKenr80vbufgue42bqsu3wb8bw4 foreign key (PLANET_DESTINATION)
Hibernate: alter table TRAFFIC add constraint FKld0559uvifjymk6ev1g55hlvd foreign key (PLANET_DESTINATION
Hibernate: declare global temporary table session.HT_ROUTES (PLANET_DESTINATION varchar(255) not null, PL
Hibernate: declare global temporary table session.HT_TRAFFIC (PLANET_DESTINATION varchar(255) not null, P
Oct 11, 2017 10:24:35 AM org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring FrameworkServlet 'dispatcher'
Oct 11, 2017 10:24:37 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Oct 11, 2017 10:24:37 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Oct 11, 2017 10:24:37 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 26134 ms
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_ from
```

• harika-dindu-service

If your hostname, IP and context root are not same as below. After deploying the harika-dindu-domain application, change the baseurl property in application.properties file according to the url of domain application, which is inside service resources classpath. Then build the war file of service application and deploy.

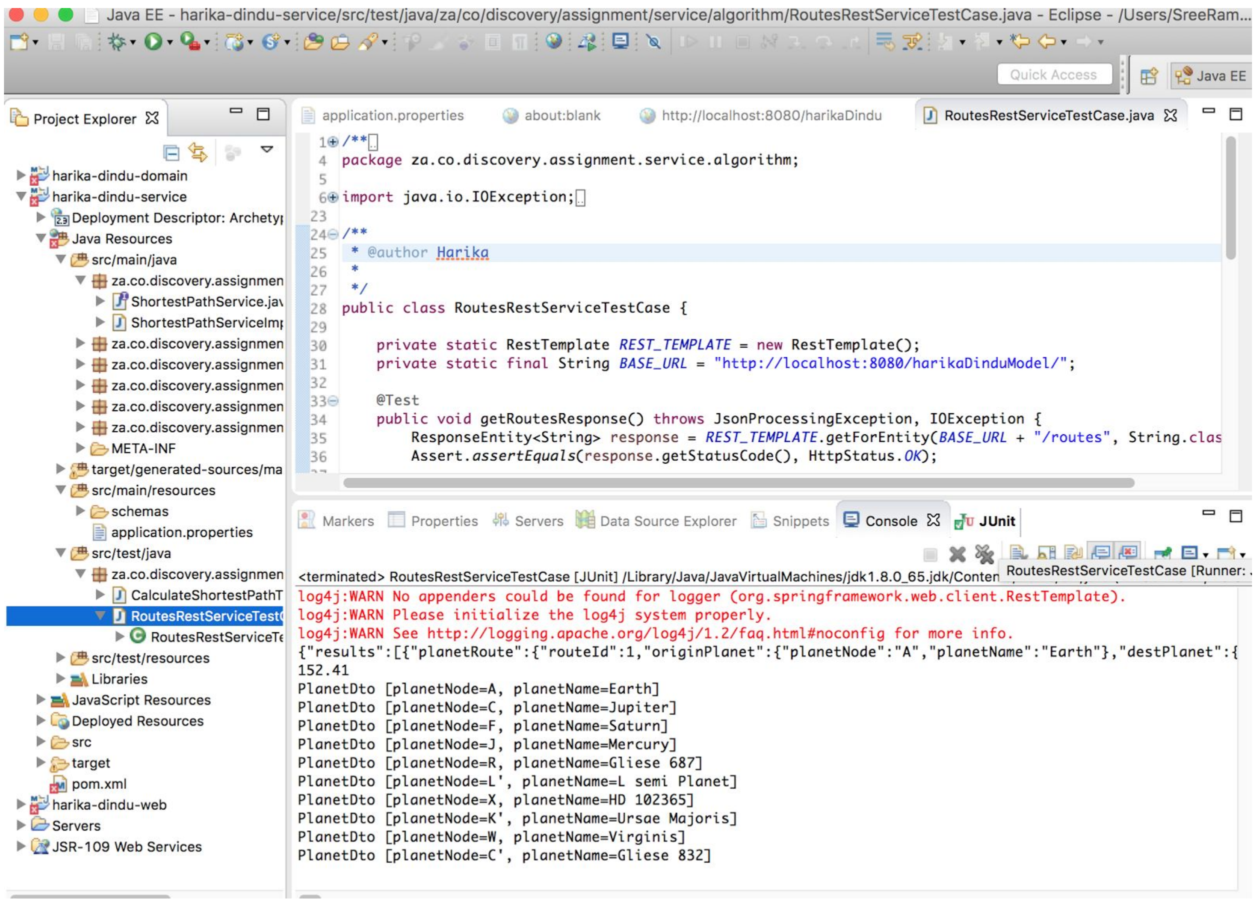


```
baseurl=http://localhost:8080/harikaDinduModel/
```

The service will be published to below url.

http://localhost:8080/harikaDinduService/shortestpathws/shortestroute.wsdl

After the harika-dindu-service application is deployed, the below test case can be executed to verify the integration with domain application.

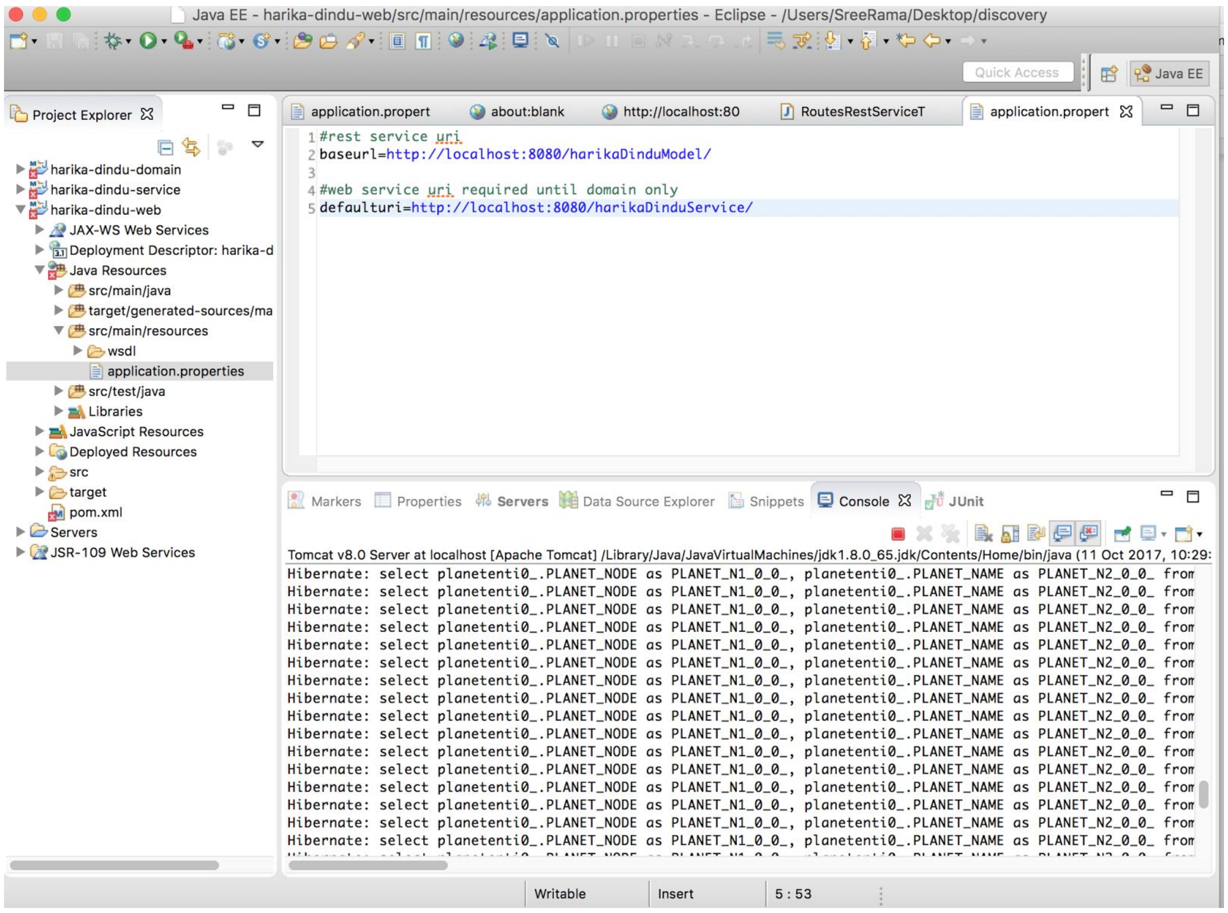


```
/**
package za.co.discovery.assignment.service.algorithm;

import java.io.IOException;

/**
 * @author Harika
 *
 */
public class RoutesRestServiceTestCase {

    private static RestTemplate REST_TEMPLATE = new RestTemplate();
    private static final String BASE_URL = "http://localhost:8080/harikaDinduModel/";

    @Test
    public void getRoutesResponse() throws JsonProcessingException, IOException {
        ResponseEntity<String> response = REST_TEMPLATE.getForEntity(BASE_URL + "/routes", String.clas
        Assert.assertEquals(response.getStatusCode(), HttpStatus.OK);
```

```
<terminated> RoutesRestServiceTestCase [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Conten
log4j:WARN No appenders could be found for logger (org.springframework.web.client.RestTemplate).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
{"results":[{"planetRoute":{"routeId":1,"originPlanet":{"planetNode":"A","planetName":"Earth"},"destPlanet":{
152.41
PlanetDto [planetNode=A, planetName=Earth]
PlanetDto [planetNode=C, planetName=Jupiter]
PlanetDto [planetNode=F, planetName=Saturn]
PlanetDto [planetNode=J, planetName=Mercury]
PlanetDto [planetNode=R, planetName=Gliese 687]
PlanetDto [planetNode=L', planetName=L semi Planet]
PlanetDto [planetNode=X, planetName=HD 102365]
PlanetDto [planetNode=K', planetName=Ursae Majoris]
PlanetDto [planetNode=W, planetName=Virginis]
PlanetDto [planetNode=C', planetName=Gliese 832]
```

- **harika-dindu-web**

If your hostname, IP and context root are not same as below. After deploying the domain and service applications, change the baseurl and defaulturi properties in application.properties file according to the url of domain and service applications, which is inside web resources classpath. Then build the war file of service application and deploy.

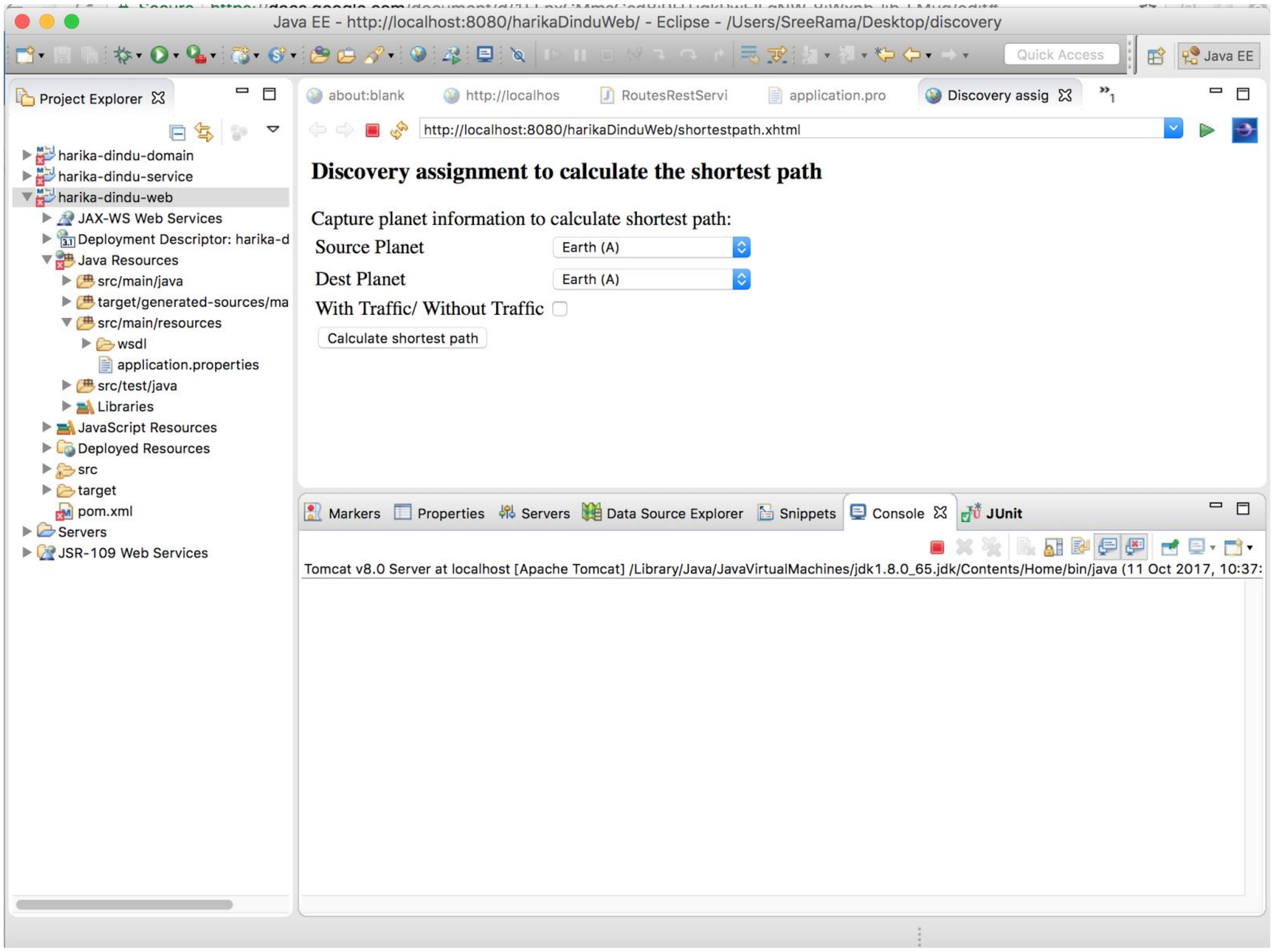


```
#rest service uri
baseurl=http://localhost:8080/harikaDinduModel/

#web service uri required until domain only
defaulturi=http://localhost:8080/harikaDinduService/
```

```
Tomcat v8.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (11 Oct 2017, 10:29:
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
Hibernate: select planetenti0_.PLANET_NODE as PLANET_N1_0_0_, planetenti0_.PLANET_NAME as PLANET_N2_0_0_ from
```

Run the web application as below, it will redirect automatically to the shortestpath controller.

http://localhost:8080/harikaDinduWeb


Java EE - http://localhost:8080/harikaDinduWeb/ - Eclipse - /Users/SreeRama/Desktop/discovery

http://localhost:8080/harikaDinduWeb/shortestpath.xhtml

Discovery assignment to calculate the shortest path

Capture planet information to calculate shortest path:

Source Planet: Earth (A)

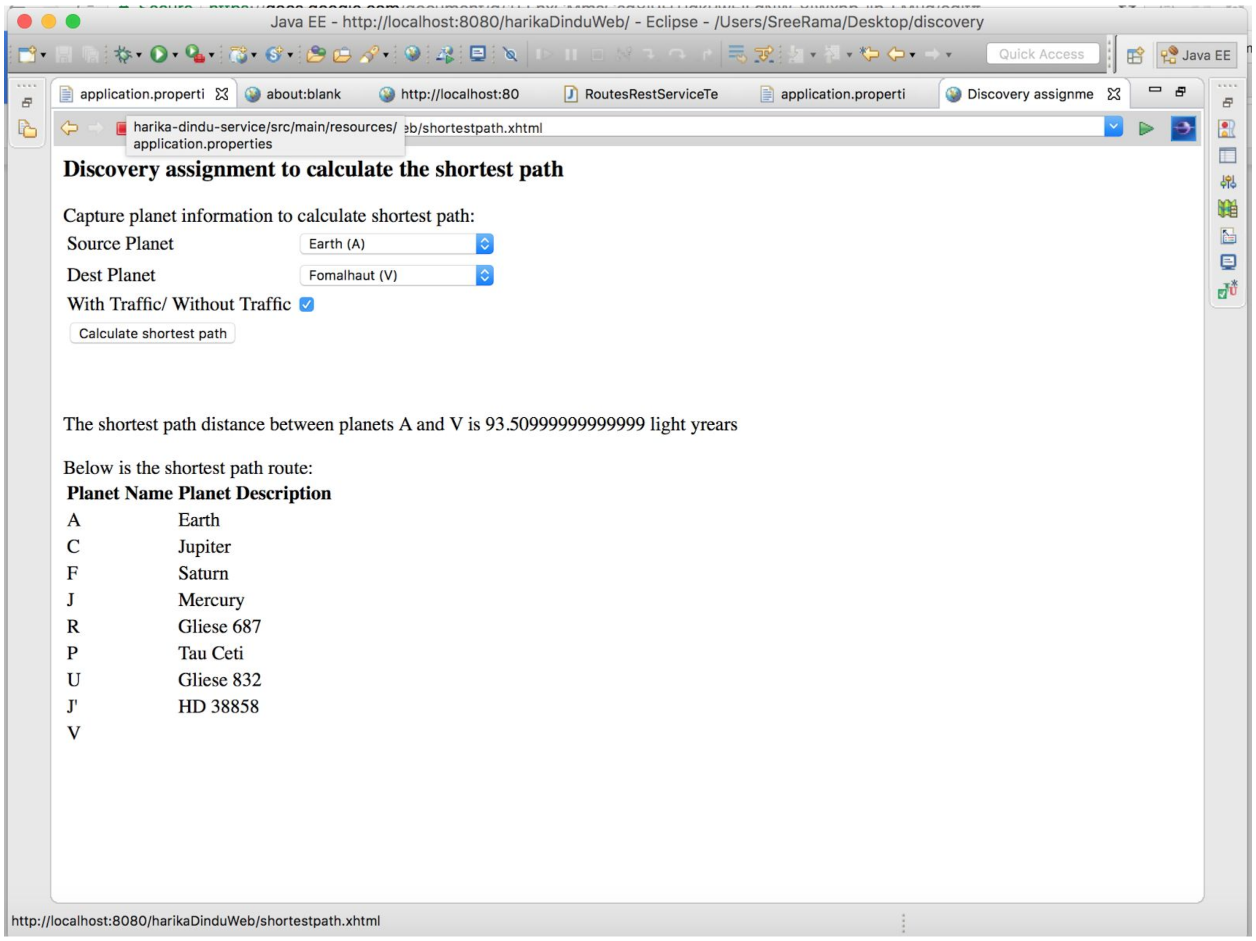
Dest Planet: Earth (A)

With Traffic/ Without Traffic: ☐

Calculate shortest path

```
Tomcat v8.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (11 Oct 2017, 10:37:
```

Capture the details and the response data will be displayed.


Java EE - http://localhost:8080/harikaDinduWeb/ - Eclipse - /Users/SreeRama/Desktop/discovery
Quick Access
Java EE
application.properti
about:blank
http://localhost:80
RoutesRestServiceTe
application.properti
Discovery assignme
harika-dindu-service/src/main/resources/application.properties
eb/shortestpath.xhtml

Discovery assignment to calculate the shortest path

Capture planet information to calculate shortest path:

Source Planet Earth (A)

Dest Planet Fomalhaut (V)

With Traffic/ Without Traffic ✓

Calculate shortest path

The shortest path distance between planets A and V is 93.50999999999999 light yrears

Below is the shortest path route:

Planet Name	Planet Description
A	Earth
C	Jupiter
F	Saturn
J	Mercury
R	Gliese 687
P	Tau Ceti
U	Gliese 832
J'	HD 38858
V	

http://localhost:8080/harikaDinduWeb/shortestpath.xhtml