LAB 13 – Arithmetic Service

1. **Launch Eclipse**
    a. Accept the default workspace
2. **Create a Project**
    a. *File > New > Dynamic Web Project*
    b. **Project Name**: `Calculator`
    c. Click **Finish**.
3. **Convert project into a Maven project**
    a. Right-click on the project. *Configure > Convert to Maven Project*
    b. Accept defaults and click **Finish**.
4. **Set Maven dependencies to include required libraries**
    a. Expand project in the Project Explorer view
    b. Double-click **`pom.xml`** (this is the Maven config file)
    c. Click on the “Dependencies” tab
    d. Add each dependency by clicking the “Add” button on the left.
    e. Add the following dependencies (same as previous lab):

Group Id	Artifact Id	Version
asm	asm	3.3.1
org.json	json	20140107
com.sun.jersey	jersey-bundle	1.19
com.sun.jersey	jersey-core	1.19
com.sun.jersey	Jersey-server	1.19

    f. Save `pom.xml` (*File > Save* or *Ctrl + S*).
5. **Create a service implementation class**
    a. Right-click on the project. *New > Class*
    b. **Package:** `com.labs.rest`
    c. **Name:** `ArithmeticService`
    d. Click **Finish**.
6. **Complete the source code for the service**

```
package com.rest.labs;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
```

```
@Path("/math")
public class ArithmeticService {
   @GET
   @Produces("text/html")
   public String noParams() {
      return "Service supports two resource paths:
      <ul><li><i>/math/add/?/?</i></li> <li><i>/math/subtract/?/?</i></li></ul>";
   }

   @Path("add/{p1}/{p2}")
   @GET
   @Produces("application/xml")
   public String addNums(@PathParam("p1") Double p1, @PathParam("p2") Double p2){
         Double sum = p1 + p2;

         String result = "Output: \n\nCalculator Addition Results: \n\n" + sum;
         return "<mathservice>" + "<add>" + "<firstInput>" + p1 + "</firstInput>"
               + "<secondInput>" + p2 + "</secondInput>" + "</add>"
               + "<output>" + result + "</output>" + "</mathservice>";
   }

   //TODO: Set path to “subtract” and include two parameters

   //TODO: Set method to GET request

   //TODO: Set output format to “application/xml”

   //TODO: Define two parameters: p1 and p2, both of type Double
   public String subtractNums( ){
         //TODO: Define variable to store the difference between params

         //TODO: Build a result object

         //TODO: Return a result string including result object

   }

}
```

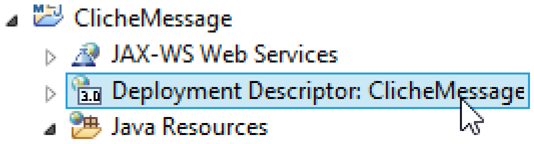
a. Update the `subtractNums()` definition by following the instructions in the `TODO` comments.

b. Save `ArithmeticService.java` (*File > Save* or *Ctrl + S*).

7. Configure the service container

a. Right-click the project. *Java EE Tools > Generate Deployment Descriptor Stub*

b. Open the project’s Deployment Descriptor.



ClicheMessage
JAX-WS Web Services
Deployment Descriptor: ClicheMessage
Java Resources

c. Add a `Servlet` and `ServletMapping` definitions below the `welcome-file-list` element.

```
<servlet>
  <servlet-name>Restful Web App</servlet-name>
  <servlet-class>
    com.sun.jersey.spi.container.servlet.ServletContainer
   </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>Restful Web App</servlet-name>
  <url-pattern>/calc/*</url-pattern>
</servlet-mapping>
```

8. Deploy project to server

   a. Click the *Servers* tab directly below the editor.

   b. Right-click on the Tomcat server. Select *"Add and Remove…"*

   c. Click the project name on the left, click the "Add" button, and click **Finish**.

9. Test the service

   a. Click the green play button on the right side of the *Servers* tab.

   b. Test the service using your browser.

      i. Launch browser of your choice.

      ii. Navigate to: `http://localhost:8080/Calculator/calc/math/add/12/5`

      iii. Result should be `17`

      iv. Replace `add` with `subtract` and substitute other numbers in the URI