

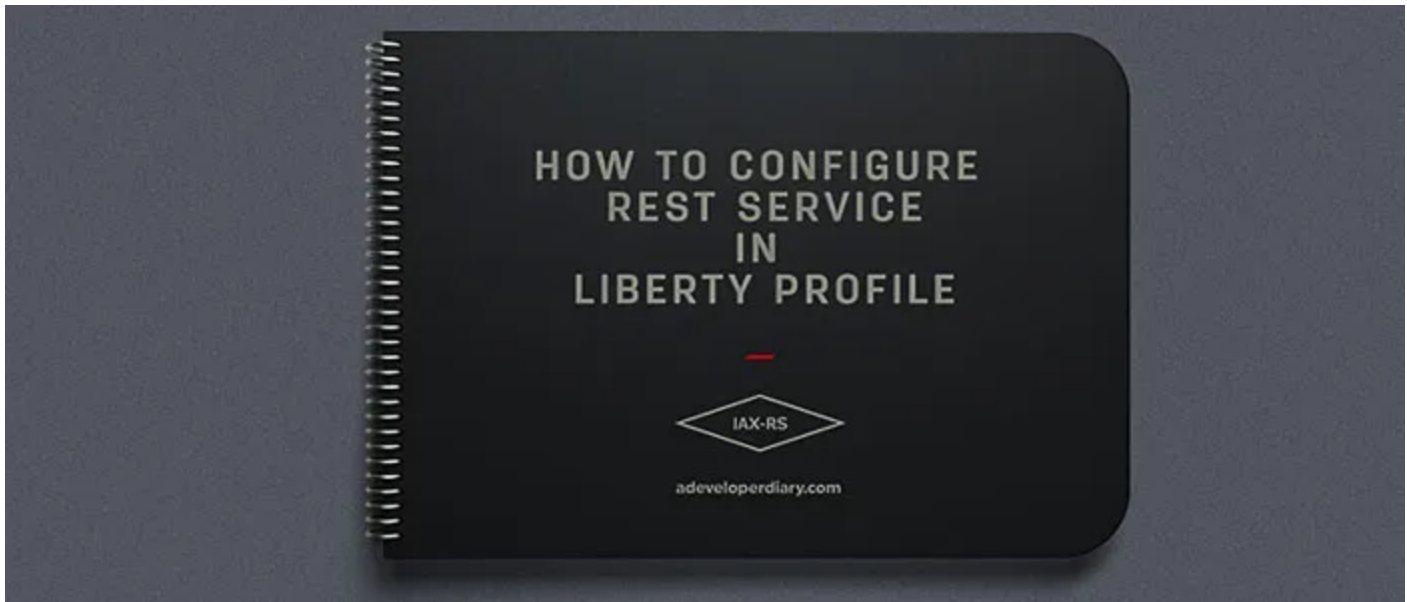
A Developer Diary

{about:"code learn and share"}

[Home](#)[Data Science](#)[Java](#)[JavaScript](#)[jBPM](#)[Tools](#)[Tips](#)[About](#)

September 1, 2015 By [Abhisek Jana](#) — [Leave a Comment \(Edit\)](#)

How to configure REST Service (JAX-RS) in IBM Liberty Profile



Liberty has built in support for JAX-RS, so you can easily implement this without any additional dependency. It's very easy and you will the service setup in no time.

JAX-RS has gained the popularity over JAX-WS for many reasons, such as light weight, ability to invoke from Java Script, easy to setup etc.

If you want to find out how to setup Liberty you can follow my previous post on [How to Setup Liberty](#). Here are the below steps on how to configure REST Service (JAX-RS) in IBM Liberty Profile.

Liberty Server Setup :

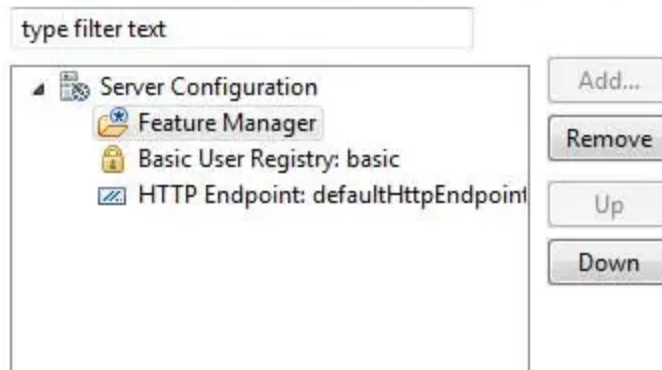
Expand the Server and open the **Server Configuration**.

[the_ad id="763"]



Select Feature Manager in the design view.

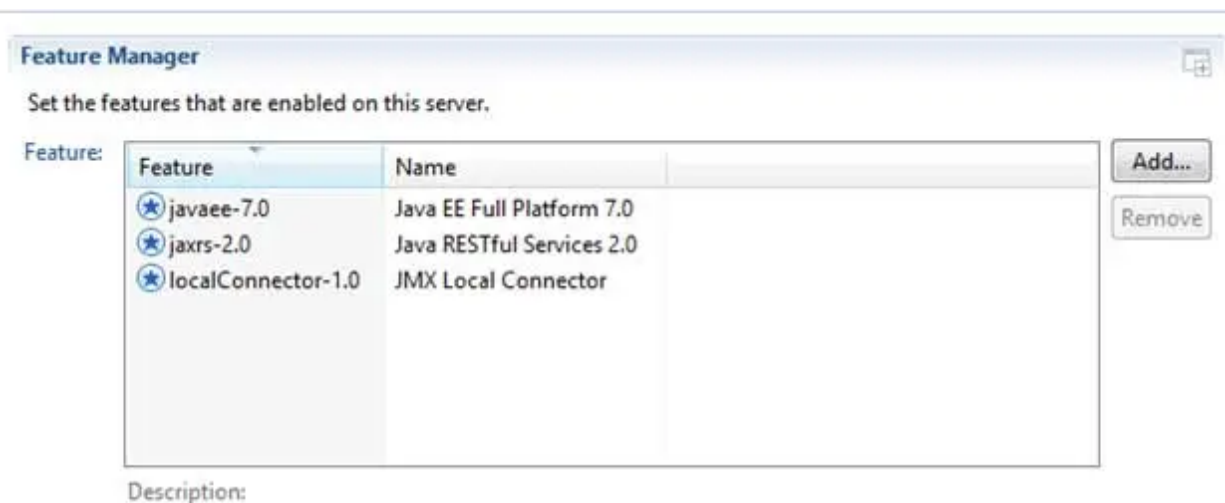
Define the main contents of the configuration in this section.



If you already have `javaee-7.0` then you should be good. Otherwise click on Add.

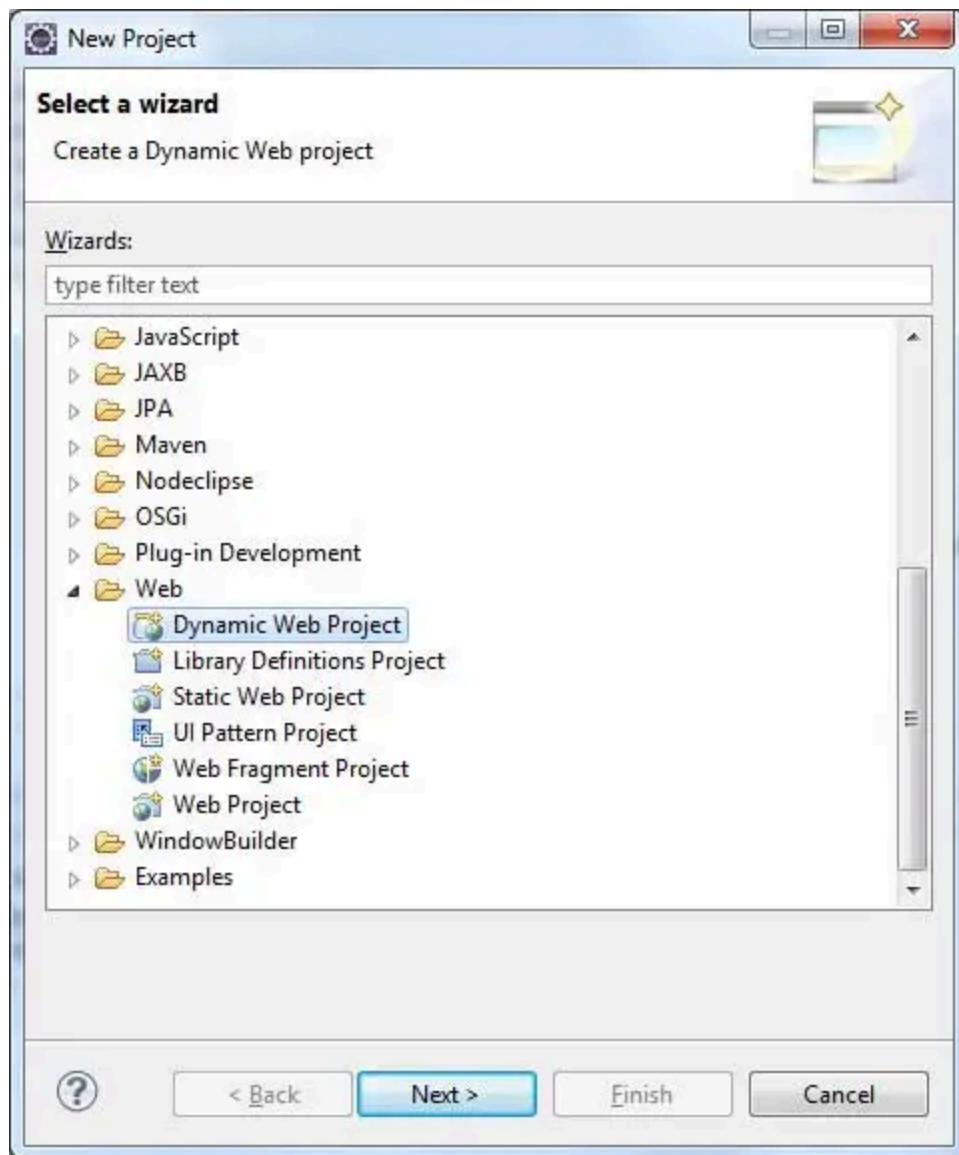


Select `jaxrs-2.0` and click on OK. Save the `server.xml` by `Ctrl+S`.



Create a REST Project:

Create a New **Dynamic Web Project**.



Add the Project Name and Click on Finish.

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

A good starting point for working with WebSphere Application Server Liberty Profile runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

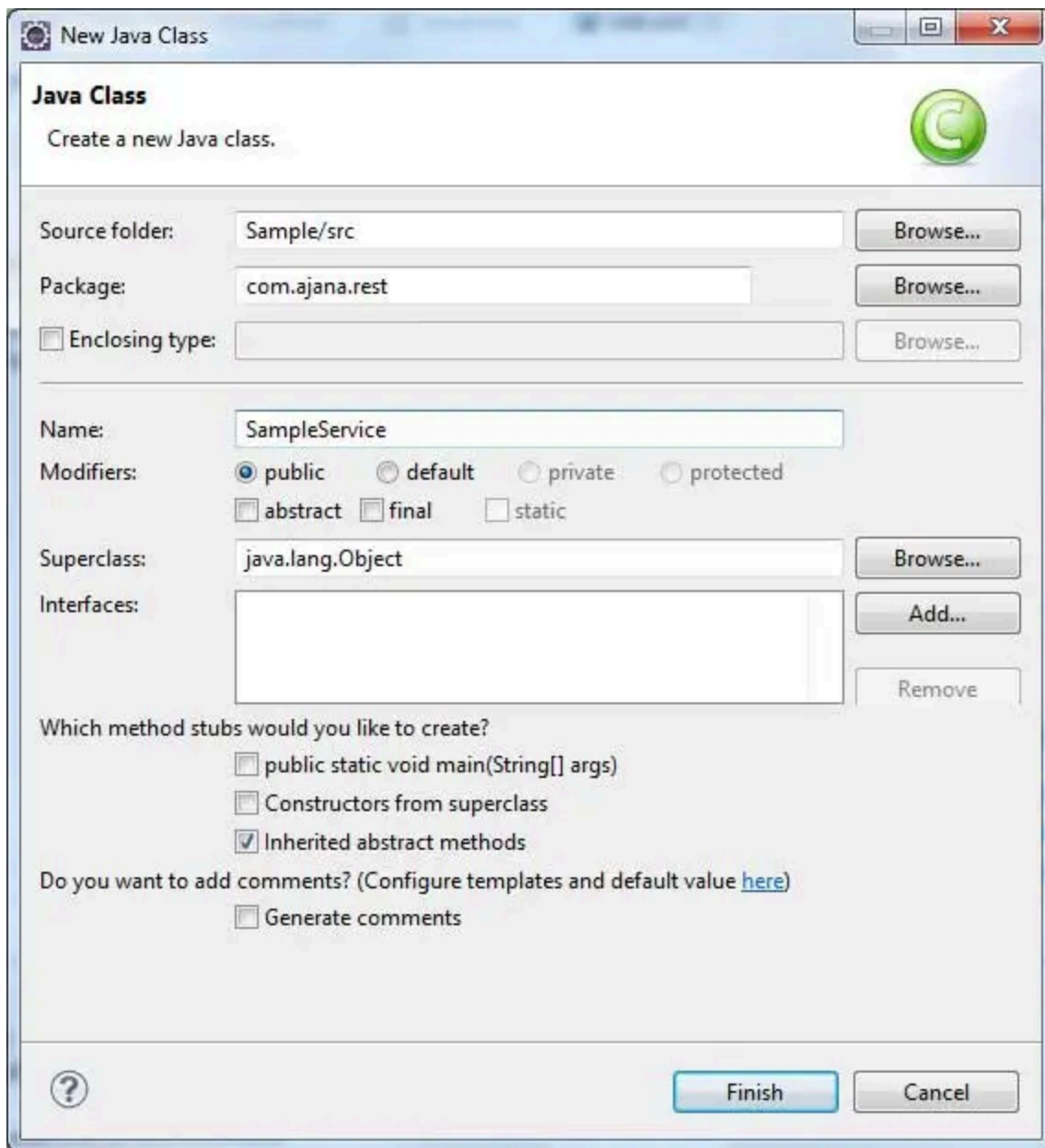
EAR project name:

Working sets

☐ Add project to working sets

Working sets:

Create a new class named `SampleService`.



Code:

Here is a sample java service class. I have extended `javax.ws.rs.core.Application` in the service however you could create a separate class for this and have the `@ApplicationPath` defined there. I have put everything in same class just for the demo. Once the class is ready, add the project to the server and start the server.

[the_ad id="767"]

```
package com.ajana;
```

```
import java.util.Arrays;
import java.util.List;

import javax.ws.rs.ApplicationPath;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;
import javax.ws.rs.core.Application;

@ApplicationPath("rest")
@Path("sample")
public class SampleService extends Application {

    @GET
    @Path("getArray")
    @Produces("application/json")
    public List<String>
getSampleData(@QueryParam("value") String value) {
        return Arrays.asList(new String[] {
"one", "two", "three", "four", value });
    }

    @GET
    @Path("getString")
    @Produces("application/json")
    public String getString() {
        return "Test";
    }
}
```

There are two services defined here, `getString` & `getSampleData`.

In order to invoke the service please use the following urls:

```
http://localhost:9080/Sample/rest/sample/getString
```

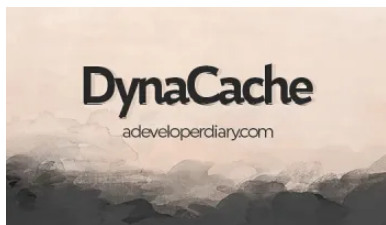
Above should return the test as the response.

```
http://localhost:9080/Sample/rest/sample/getArray?value=five
```

The above URL should return the following JSON as response.

```
["one","two","three","four","five"]
```

Related



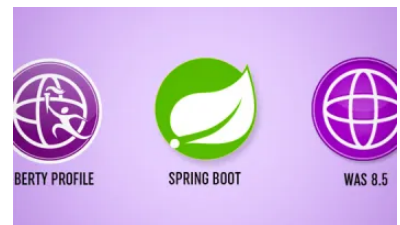
**Steps for implementing
DynaCache in Liberty and
Websphere**

In "Cache"



**How to setup IBM Liberty
Profile**

In "Server"



**How to deploy Spring Boot
application in IBM Liberty
and WAS 8.5**

In "Spring Boot"

Filed Under: **REST** Tagged With: **Java, JAX-RS, liberty, Liberty Profile, REST, Webservice**

Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

Leave a Reply

Logged in as Abhisek Jana. [Edit your profile](#). [Log out?](#) Required fields are marked *

Comment *

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed](#).

Copyright © 2024 A Developer Diary