

## COMMUNITY

### Welcome

[Dashboard](#)  
[Download Now](#)  
[Start Here](#)  
[Participate](#)  
[Contribute](#)

### Liferay Projects

[Alloy UI](#)  
[Liferay Faces](#)  
[Liferay IDE](#)  
[Liferay Portal](#)  
[Liferay Social Office](#)

### Community Projects

### Forums

### Blogs

### Wiki

### User Groups

### Special Activities

[Security Team](#)  
[Translation Team](#)  
[100 Papercuts](#)  
[BugSquad](#)  
[Community Verifier](#)  
[Radio Liferay](#)  
[2012 Contributor Awards](#)

### Upcoming Events

### Liferay University

## Wiki

Search

[Main](#) | [Proposals](#)

[FrontPage](#) | [Recent Changes](#) | [All Pages](#) | [Orphan Pages](#) | [Draft Pages](#)

# Creating Liferay 6 plugin web service

[« Back to Web Services](#)

[Details](#)
[Print](#)

Community: [developing-on-liferay](#)

Tags: [development](#) [building portlets using liferay plugins sdk](#) [liferay ide](#) [build-wsdd](#) [build-service](#) [building liferay services](#)

Today I tried to create a service in a plugin portlet of liferay. Since this is not very straight forward, it took me the whole day to get my simple service running. As a matter of fact, I decided to publish this on the web.

I used Liferay6.0.5 source and plugin sdk, along with eclipse 3.6, with WTP and Liferay IDE 1.0.1. The base platform was Java1.6.0\_21 and MySQL5.1.46, both running on openSuse11.3. The deployment server was tomcat6.0.26.

### Step 1: Setup liferay source project and tomcat

1. Unzip the liferay source in your eclipse workspace. In eclipse, create a new project with the name liferay-portal-src-6.0.5. Eclipse will automatically detect the project type and configure it accordingly.
2. Open the app.server.properties and edit the app.server.parent.dir to point to the folder containing the tomcat installation. Note that this is the parent folder of CATALINA\_HOME. For me it was:

```
app.server.parent.dir=/home/swoosh/installations
```

3. Un-comment: `app.server.type=tomcat`
4. Scroll down and change the `app.server.tomcat.dir`, so that it points to the `CATALINA_HOME`. For me it was:  

```
app.server.tomcat.dir=${app.server.parent.dir}/liferay-  
tomcat-6.0.26
```
5. Save and close the `app.server.properties`
6. Create a DB schema and user using any tool (I used MySQL Administrator)
7. Open the `portal.properties` and comment out the HSQL (Hypersonic) DB settings.
8. Un-comment the MySQL DB settings and change the host, schema name, username and password. For me the settings were:  

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver  
jdbc.default.url=jdbc:mysql://localhost/lportal?  
useUnicode=true&characterEncoding=UTF-  
8&useFastDateParsing=false  
jdbc.default.username=liferay  
jdbc.default.password=test
```
9. Save and close the `portal.properties` file.
10. Now open the ant view in eclipse and drag `liferay-portal-src-6.0.5/portal-web/build.xml` and `liferay-portal-src-6.0.5/build.xml` file to the ant view. Wait for some time as ant loads the set of build files.
11. Run the `build-themes` task under `portal-web` (from `liferay-portal-src-6.0.5/portal-web/build.xml`). Make sure that you get Build Successful message everytime you fire an ant task. (There is a bug in liferay's builder due to which it doesn't create the themes when you compile the source. If you directly deploy the source, you wont be able to access the home page).
12. Now run the `compile` task under `portal` (from `liferay-portal-src-6.0.5/build.xml`).
13. Then run the `deploy` task under the same file. Watch the log to verify that ant is copying the files to the right folder. If not, then double check the settings specified in the `app.server.properties` file.
14. Now you have tomcat bundled up with the liferay portal. To start it you need to set up the runtime in eclipse. Go to `Window->Preferences->Server->Runtime Environments`. Click Add. Select the server type as

Liferay v6.0 (Tomcat 6). Click next. Give a name to your runtime - Liferay v6 (Tomcat 6). Browse and select the tomcat folder that you specified in `app.server.properties` `app.server.tomcat.dir` property. Select runtime as java 1.6 and click Finish.

15. If you don't have a server project in your workspace, then create a new one. Right click on the server project and Create a new server. Select the server type as Liferay v6.0 (Tomcat 6) and runtime as Liferay v6 (Tomcat 6). Server's host name should be localhost. Click next and click Finish.
16. Now the server should appear under your server view (Add the view from Window->Show View->Servers)
17. Double click on the server and change the Timeouts to 300 seconds. This will give enough time to the server to create and populate the DB schema for the first time.
18. Right click on the server and select start.
19. The liferay server should start. For the first time, it will create all the tables and populate them with the seed data.
20. Once you get the server started message, open the web browser and open the `http://localhost:8080`. You should get the liferay's default home page. Enter `test@liferay.com` as email address and `test` as password.
21. If you are not able to see the liferay's page even though the source was compiled and deployed successfully, then please check your `app.server.properties` file.

## Step 2: Setup plugin SDK.

1. Download and unzip the plugins sdk.
2. Open the `liferay-plugins-sdk-6.0.5/build.properties` file and change the server settings:

```
app.server.type=tomcat
app.server.dir=/home/swoosh/installations/liferay-
tomcat-6.0.26
app.server.deploy.dir=${app.server.dir}/webapps
app.server.lib.global.dir=${app.server.dir}/lib/ext
app.server.portal.dir=${app.server.dir}/webapps/ROOT
```
3. Go to Window->Preferences->Liferay->Installed SDKs.
4. Click on Add. In the pop-up window browse to the unzipped sdk folder and click Ok.

### Step 3: Create the service project

1. Now create a new Liferay plug-in project. Enter the details as in the screenshot below:

#### Liferay Plug-in Project

Create a new plug-in project for Liferay Portal.



Project name:

Display name:

Liferay SDK:  
 [Configure SDKs](#)

Liferay runtime:

Plug-in Type

<input checked="" type="radio"/>		Portlet	A plug-in for implementing Liferay's portlet modes and actions
<input type="radio"/>		Hook	A hot-deployable plug-in for extending or customizing Liferay's functionality.
<input type="radio"/>		Ext	A monolithic plug-in allowing for complete Liferay customization.
<input type="radio"/>		Layout	A plug-in for customizing Liferay's page layouts.
<input type="radio"/>		Theme	A plug-in for customizing Liferay's look and feel.

2. Click Next and select Dynamic Web module version 2.5 and Default configuration for Liferay v6.0 (Tomcat 6).
3. Click Finish and Click Yes on the Switch Perspective dialog box.
4. A new TestPlugin-Portlet Project will appear in the navigator.
5. Now right click on the plugin project and select New->Liferay Service.
6. Enter the detail as in the screenshot below:

## Create Liferay Service

Create a new service builder file in a project.



Plug-in project:

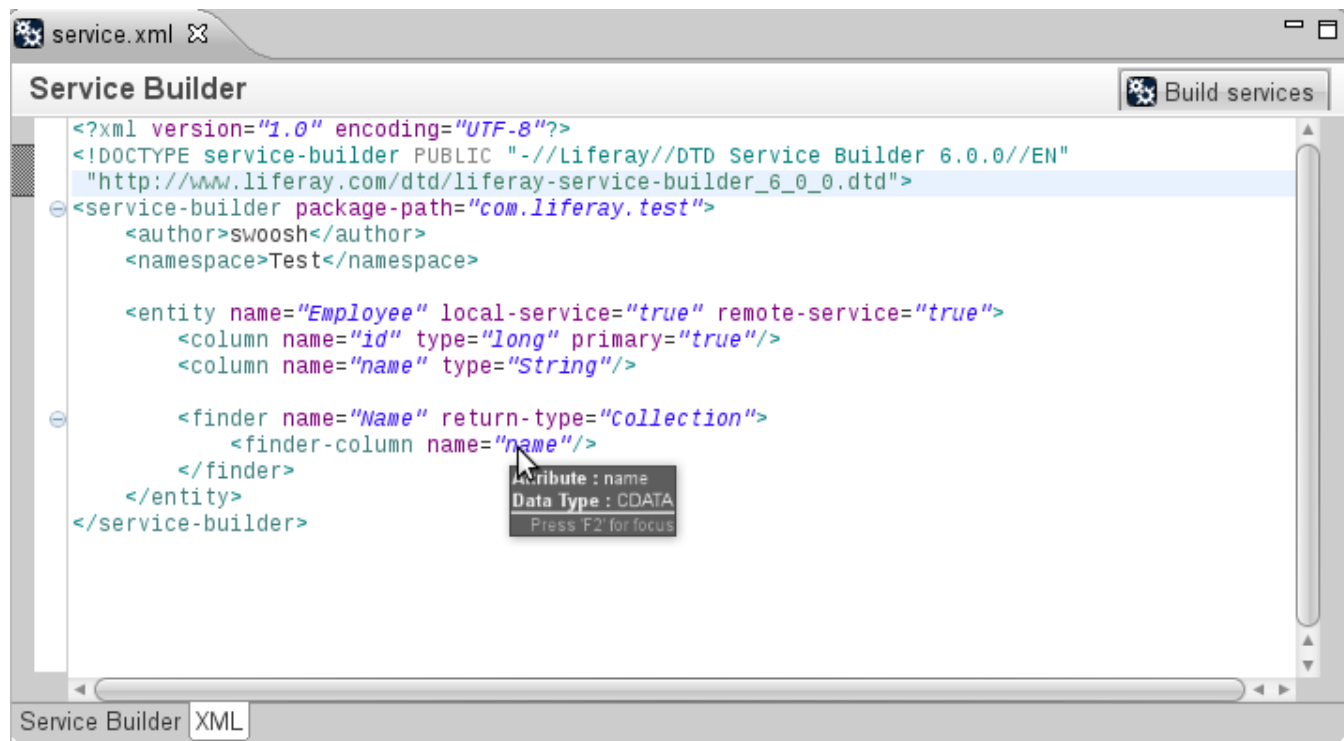
Service file:

Package path:

Namespace:

Author:

7. You can enter a different package and Namespace. In that case you will have to follow the next steps accordingly.
8. Click Finish. The service.xml file will be created and will be opened in the service editor.
9. The service.xml is similar to the service definition files in the ext environment of older liferay versions. Here we can define as many entities along with their functions.
10. We will create an Employee entity with id (long) and name (String) attributes. The service definition looks like:



11. Save the service.xml file.
12. Drag the TestPlugin-portlet/build.xml to the Ant view.
13. In the ant view, expand the TestPlugin-portlet and run the build-service task.
14. If the build fails, please check if you have followed the point number 2 in Step 2. On successful build, the ant task creates the files related to Employee entity.
15. Refresh your project (F5) to see the files.
16. Right click on the project go to Properties->Java Build Path->Source tab. Add the TestPlugin-portlet/docroot/WEB-INF/service and TestPlugin-portlet/docroot/WEB-INF/src folders, and click Ok.
17. Implement the service by editing  
com.liferay.test.service.impl.EmployeeLocalServiceImpl. Add the implementations for CRUD and finder methods. They look like the following listing:
 

```

public Employee create(Employee model) throws
SystemException{
    long id =

```

```

CounterLocalServiceUtil.increment(Employee.class.getName());
    model.setId(id);
    Employee newModel =
EmployeeUtil.create(model.getId());

    newModel.setName(model.getName());
    EmployeeUtil.update(newModel, true);
    return newModel;
}

public Employee update(Employee model) throws
NoSuchEmployeeException, SystemException{
    Employee newModel =
EmployeeUtil.findByPrimaryKey(model.getId());

    newModel.setName(model.getName());

    EmployeeUtil.update(newModel, true);

    return newModel;
}

public List<Employee> findByName(String name) throws
SystemException{
    return EmployeeUtil.findByName(name);
}

public Employee remove(long id) throws
NoSuchEmployeeException, SystemException{
    return EmployeeUtil.remove(id);
}

```

18. Run the TestPlugin-portlet/build-service task and make sure that build is successful.

19. Similarly edit the

```

com.liferay.test.service.impl.EmployeeServiceImpl Add
the CRUD and finder methods as listed below:
public Employee create(Employee model) throws
SystemException{
    return EmployeeLocalServiceUtil.create(model);
}

```

```

public Employee update(Employee model) throws
NoSuchEmployeeException, SystemException{
    return EmployeeLocalServiceUtil.update(model);
}

public List<Employee> findByName(String name) throws
SystemException{
    return EmployeeLocalServiceUtil.findByName(name);
}

public Employee remove(long id) throws
NoSuchEmployeeException, SystemException{
    return EmployeeLocalServiceUtil.remove(id);
}

```

20. Run the TestPlugin-portlet/build-service task again and make sure that build is successful.
21. There is a bug in liferay's service builder due to which it doesn't generate the ServiceSoap class correctly. It is not able convert the parameters to their corresponding soap type. See <http://issues.liferay.com/browse/LPS-12629> for details. To get around this issue open the EmployeeServiceSoap and change the generated create and update methods as:

```

publicstatic com.liferay.test.model.EmployeeSoap create(
    com.liferay.test.model.EmployeeSoap model) throws
RemoteException {
    try {
        com.liferay.test.model.Employee returnValue =
EmployeeServiceUtil.create(com.liferay.test.model.impl.EmployeeModelImpl.toModel(model));

        return
com.liferay.test.model.EmployeeSoap.toSoapModel(returnValue);
    }
    catch (Exception e) {
        _log.error(e, e);

        throw new RemoteException(e.getMessage());
    }
}

```



```

publicstatic com.liferay.test.model.EmployeeSoap update(
    com.liferay.test.model.EmployeeSoap model) throws
    RemoteException {
    try {
        com.liferay.test.model.Employee returnValue =
        EmployeeServiceUtil.update(com.liferay.test.model.impl.EmployeeModelImpl.toModel(model));

        return
        com.liferay.test.model.EmployeeSoap.toSoapModel(returnValue);
    }
    catch (Exception e) {
        _log.error(e, e);

        thrownew RemoteException(e.getMessage());
    }
}

```

22. Now run the `TestPlugin-Portlet/build-wsdd` task to generate the webservices.
23. Once this is complete, start the server if not already started.
24. Then run `TestPlugin-Portlet/compile` and `TestPlugin-Portlet/deploy`.
25. Check the logs for "1 portlet for TestPlugin-portlet is available for use" message.
26. If that appears, open the `http://localhost:8080` in your browser.
27. Login as `test@liferay.com` and password `test`.
28. Go to Add -> More->Samples. Add the TestPlugin to you page. Make sure that the default page for the plugin appears. This confirms that the plugin was deployed properly.
29. Now open `http://127.0.0.1:8080/TestPlugin-portlet/axis/Plugin_Test_EmployeeService?wsdl` in your browser and confirm that the wsdl opens up.
30. This completes the plugin service creation.

#### Step 4: Create Web service client

1. Now we need to test our webservice. For that we need to have another tomcat runtime as our Liferay tomcat 6 runtime eclipse plugin doesn't support building webservices. Also we need to make sure that the tomcat runtime and liferay runtime run on different ports. I used the ports 9005, 9080 and 9009 for my tomcat runtime.
2. In eclipse go to New->Webservice Client and enter the details as in the

screenshot below:

## Web Services

Select a service definition and move the slider to set the level of client generation.

Service definition:

Client type:

**Test client**

Configuration:

- [Server runtime: Tomcat v6.0 Server](#)
- [Web service runtime: Apache Axis](#)
- [Client project: WebServiceProject](#)

☐ Monitor the Web service

3. Click Next on this and next screen. Then click on "Start server" to start the tomcat runtime. Then click Finish.
4. Select all the methods and click Finish
5. Eclipse will then deploy the client and open up a screen to test your webservice.
6. Click on the methods to test them.
7. We will test create method first. We will provide the primaryKey=0, name=Liferay and id=0 and click invoke.
8. In the results section we can see that the employee entity has been created with the id=1.
9. Now invoke findByName and provide name=Liferay. The result will show the reference of the entity object returned by the webservice.

That completes the tutorial. The complete plugin and the webservice client can

be downloaded from the attachments of this post. Hope that was helpful. Do write your comments.

## Children Pages

- [Creating and Consuming Web Services in Liferay](#)

 [2 Attachments](#) | 38929 Views

Average (10 Votes)



### ▼ Comments

#### Showing 45 Comments



Juan Fernández

9/14/10 1:28 AM

Nice article. Thanks!



Kapil Dev Verma

9/14/10 2:24 AM

Thanks for the feedback 😊



XinYang Zhang

9/14/10 2:29 AM

Some picture of this article can't display,I can't see them.Could you fix it



Kapil Dev Verma

9/14/10 4:25 AM

I have updated the article, hope you can see the images now. Please do a ctrl+F5.



Marius Marius

9/14/10 5:50 AM

Please, I need your help 😊

Everything compiles, deploys etc. but I am stuck at this point:

`http://127.0.0.1:8080/TestPlugin-portlet/axis/Plugin_Test_EmployeeService?wsdl` I get axis error because "Could not generate WSDL"

of course and there: `http://127.0.0.1:8080/tunnel-web/axis` I don't see my plugin.



Andrius Kurtinaitis

9/14/10 6:06 AM

In this tutorial you advice to edit portal.properties file. This is officially not recommended. This file does not even exist in a zipped bundle. Wouldn't it be better to change this filename to portal-ext.properties?



**Kapil Dev Verma**

9/14/10 9:48 AM

---

Thanks for the suggestion 😊



**Kapil Dev Verma**

9/14/10 9:50 AM

---

If everything goes fine, then you should get docroot/WEB-INF/server-config.wsdd file generated. Verify if it is getting generated. Also, it would be helpful if you attach your plugin project.



**sasi Kanth**

9/16/10 11:12 PM

---

Hi Kapil,

Really very nice description to generate service and web services in a portlet.

Thank you.



**Kapil Dev Verma**

9/17/10 1:37 AM

---

Thanks Sasi for the feedback!



**Liferay Developer**

9/17/10 8:41 AM

---

Hi Kapil

Thanks for the excellent article.

I am having the same problem as Marius. There is an error when I try to generate the wsdl and I checked for the server-config.wsdd and was not able to find it.



**Kapil Dev Verma**

9/19/10 11:56 AM

---

Hi.. Please provide me the zip of your plugin project. Thanks!



**Liferay Developer**

9/23/10 12:23 PM

---

Kapil,

what is the best way to send you the zip?

Thanks..



**Kapil Dev Verma**

9/26/10 10:15 PM

You can mail me that at kapildverma@gmail.com



**Harish Kumar**

9/27/10 9:34 PM

Very Nice Article. Thanks!!



**Andreea Tabacariu**

10/27/10 2:53 AM

I need some help. Everything compiles, the wsdd is created, the deploy works with no error, the portlet page loads but when I try to see the wsdl:

[http://127.0.0.1:8080/TestPlugin-portlet/axis/Plugin\\_Test\\_EmployeeService?wsdl](http://127.0.0.1:8080/TestPlugin-portlet/axis/Plugin_Test_EmployeeService?wsdl) I

am redirected to the portal home page and my service does not appear in

<http://127.0.0.1:8080/tunnel-web/axis>



**Kapil Dev Verma**

11/10/10 11:59 PM

Can you try running the samples attached with the post? If you still face the problem, please feel free to mail me your plugin.



**DO Hung Thuan**

11/14/10 11:55 PM

anybody know this error, help me.

[http://www.liferay.com/community/forums/-/message\\_boards/message/6414062](http://www.liferay.com/community/forums/-/message_boards/message/6414062)



**Seshendra Nalla**

11/24/10 8:29 AM

Is there a way to expose REST based services instead of SOAP using Liferay Service builder?



**Clayton Jarczewski**

11/25/10 11:55 AM

Hi there, at the end of your tutorial when I try to test by adding a Webservice Client I get the following error:

"The Apache Axis Web service runtime in Tomcat v6.0 Server does not support the client project TestPlugin-portlet."

I set it up exactly as you have it laid out in the picture as well so hopefully someone knows how to solve this problem because I find nothing when googling.



**Clayton Jarczewski**

11/26/10 10:17 AM

Ok so I noticed what I was doing wrong, stupid mistake trying to setup the WebService Client to point to the TestPlugin instead of the WebServiceProject attached.

I'll just list some problems and solutions I had here for anyone else that runs into the same stuff I did. You need to import that WebServiceProject attached and do the WebServices Client on that.

After installing another version of Tomcat on your system you have to go into the tomcat/conf/server.xml to modify the ports like Kapil mentioned.

I also had to copy all the jars inside the WEB-INF/lib folder of WebServiceProject into the alternate Tomcat/lib folder.



**Sreeraj AV**

12/26/10 11:37 PM

---

Result

exception: java.rmi.RemoteException:

com.liferay.portal.kernel.dao.orm.ORMEException:

org.hibernate.exception.SQLGrammarException: could not load an entity:

[com.liferay.test.model.impl.EmployeeImpl#2]

how can i solve this issue?



**Kapil Dev Verma**

12/28/10 9:22 PM

---

@Raju - The issue seems to somewhere in database. Are you sure that the tables are created as required? See portlet-hbm.xml and portlet-orm.xml in /TestPlugin-portlet/docroot/WEB-INF/classes/META-INF/. Use sql files in /TestPlugin-portlet/docroot/WEB-INF/sql/ to create the tables.



**Dien NH**

1/20/11 6:38 PM

---

Very nice article. Thanks!!!



**Luca Andreatta**

1/25/11 3:09 AM

---

When I was executing the build-service task ant was waiting in the java-compile ant task. I setted fork to "no" and all goes ok. Is there any controindication?



**Venkat S**

2/7/11 2:27 PM

---

Hi Kapil,

I have a requirement, where instead of accessing db i want a webservice which does action on cache, so i tried to follow the same steps except one change, (ie)

```
<entity name="Employee" local-service="true" remote-service="true"></entity>
```

I was able to deploy, but i get the below classcastexception, can u please help,

"AXIS error

Sorry, something seems to have gone wrong... here are the details:

Fault - ; nested exception is:

org.apache.axis.ConfigurationException: java.lang.ClassCastException:

org.apache.axis.handlers.http.URLMapper cannot be cast to

org.apache.axis.Handler

java.lang.ClassCastException: org.apache.axis.handlers.http.URLMapper cannot be cast to org.apache.axis.Handler

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.makeNewInstance(WSDDDeployableItem.java:302)

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.getNewInstance(WSDDDeployableItem.java:274)

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.getInstance(WSDDDeployableItem.java:260)

at

org.apache.axis.deployment.wsdd.WSDDDeployment.getHandler(WSDDDeployment.java:394)

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.getNewInstance(WSDDDeployableItem.java:276)

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.getInstance(WSDDDeployableItem.java:260)

at

org.apache.axis.deployment.wsdd.WSDDChain.makeNewInstance(WSDDChain.java:125)

at

org.apache.axis.deployment.wsdd.WSDDDeployableItem.getNewInstance(WSDDDeployableItem.java:274)...."



**Kapil Dev Verma**

2/9/11 9:54 PM

Hi Venkat,

From the stacktrace, we can see that the makeNewInstance method in

WSDDDeployableItem.java is expecting org.apache.axis.Handler, however, org.apache.axis.handlers.http.URLMapper is being passed.

It seems to be some configuration problem or it can even be version miss match. This has nothing to do with accessing DB or calling action on cache. The error is purely on axis side.

Please send me your project zip at kapildverma@gmail.com so that i can debug it.

Regards,  
Kapil



**Andrew Tillinghast**

2/28/11 11:52 AM

---

First thank you for a great guide, works great for 6.0.5

Now I have a Axis security question:

Is there any way to make /TestPlugin-Portlet/axis available to external IPs and still leave /tunnel-web/axis local IP only?

I know I can set axis.servlet.hosts.allowed= and leave the value blank to open the axis service to the world, but that opens the liferay wsdd as well, I just want to expose my custom SOAP services.

It looks to me that I could change the <init-param> values in my web.xml but would be over written every time I deploy.

```
<init-param>
<param-name>portal_property_prefix</param-name>
<param-value>axis.servlet.</param-value>
</init-param>
```

-Andrew



**Kapil Dev Verma**

3/4/11 10:24 AM

---

Hi Andrew,

Thanks for your comments. I am not an expert with axis. However, you can have a look at apache webserver's mod\_proxy to achieve this....

[http://httpd.apache.org/docs/trunk/mod/mod\\_proxy.html](http://httpd.apache.org/docs/trunk/mod/mod_proxy.html)

Regards,



Kapil



**Nguyen Trung Kien**

3/11/11 7:23 AM

---

Nice article.Thank Kapil!



**Nguyen Trung Kien**

3/12/11 5:55 PM

---

Hi Kapil,

I want add new method in existing Liferay webservice as add new addGroupEXT to GroupServiceSOAP. What should I do?

I'm using Liferay CE 6.05.

Thanks.



**Kapil Dev Verma**

3/21/11 12:20 AM

---

Hi Nguyen,

You might have to create a new webservice for your extended class. The problem is that the webservice definition for GroupServiceSoap is embedded in liferay code and it can't be modified (without changing liferay code). So you will have to create a new webservice, with your custom addGroupEXT() method, which internally uses the liferay APIs to accomplish the tasks.

You can create your new class something like GroupEXTServiceImpl, import com.liferay.portal.service.GroupServiceUtil and write a method addGroupEXT() in it. From that method, you can call the GroupServiceUtil.addGroup().

Then you can expose your new class as webservice.

You can also try this by changing the plugin project attached with this post. It will be a lot easier if you are using Liferay's Eclipse plugin.

--

Kapil



**Tom Mahy**

8/22/11 5:31 AM

---

Nice article. Thank you



**Martin Schwietzke**

9/9/11 3:46 PM

---

Concerning Error: "org.apache.axis.handlers.http.URLMapper cannot be cast to org.apache.axis.Handler"

There are some classloading issues. axis.jar is deployed in ROOT.war and in your Plugin WAR.

Solution: Remove the axis.jar in your Plugin WAR (WEB-INF/lib). It works fine for me.



**Martin Schwietzke**

9/12/11 5:38 AM

---

Sorry, it was a mistake. Removing axis.jar is not jet the solution. One ERROR has been gone, but NoClassDefError occurs on other location.

Does anybody have a solution for the Problem?



**Tamás Barabás**

11/28/11 3:53 AM

---

Venkat S,

dont forget to run a new tomcat server, so you have the liferay's tomcat boundled, and a clear tomcat for the client



**Raghu teja**

12/14/11 9:58 PM

---

Hi Kapli,

Nice Article

But i have an issue with this ,when i am trying to bulid client (as per you article URL)as below its working fine ,methods are invoking and getting results

`http://127.0.0.1:8080/TestPlugin-portlet/axis/Plugin_Test_EmployeeService?wsdl`

But when i am trying to bulit webservice client as below URL ,

`http://127.0.0.1:8080/TestPlugin-portlet/secure/axis/Plugin_Test_EmployeeService?wsdl`

methods are invoked but when iam trying to get results shows as  
RE:404 unauthroized

Is Anything wrong in the URL???



**Shantanu Puranik**

2/1/12 1:13 AM

---

I'm getting exception: java.rmi.RemoteException: BeanLocator has not been set



**Tom Mahy**  
2/17/12 6:03 AM

---

This worked perfect on 6.0.6 but on 6.1 i cant find the service under /tunnel-web/axis  
Has anything changed ?



**Sy Do**  
3/5/12 8:10 PM

---

You can find service under /api/axis on liferay 6.1



**Patrick Momin**  
3/15/12 7:50 AM

---

in step 3 -> 29 i cannot seem to get the WSDL to show i get following error:

HTTP Status 404 - /TestPlugin-portlet/axis/Plugin\_Test\_EmployeeService

type Status report

message /TestPlugin-portlet/axis/Plugin\_Test\_EmployeeService

description The requested resource (/TestPlugin-portlet/axis/Plugin\_Test\_EmployeeService) is not available.  
Apache Tomcat/7.0.23



**Patrick Momin**  
3/16/12 2:37 AM

---

I found the Solution by comparing the entries in the corresponding web.xml files:

in liferay 6.1 the url to call is "/TestPlugin-portlet/api/axis" instead of "/TestPlugin-portlet/axis"

I hope this saved some people's time.



**Giuseppe Pantaleo**  
6/19/12 2:17 AM

---

Hi, I'm experiencing a problem since few days.  
My development environment is the following: I'm using Liferay 6.1.0 CE GA1 source and plugin sdk, along with Eclipse Indigo Service Release 2, with WTP and Liferay IDE 1.5.2. The base platform is Java 1.6.0\_32 and the deployment server is Tomcat 7.0.23 (the one bundled with Liferay).  
While executing the step 3 -22 (running TestPlugin-Portlet/build-wsdd) I'm getting the following error:  
build-wsdd:  
[java] Loading jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-7.0.23/webapps/ROOT/WEB-INF/lib/portal-impl.jar!/system.properties

```
[java] 19-giu-2012 9.05.48 com.liferay.portal.kernel.log.Jdk14LogImpl info
[java] INFO: Global lib directory /D:/liferay-portal-6.1.0-ce-ga1/tomcat-7.0.23/lib/ext/
[java] 19-giu-2012 9.05.48 com.liferay.portal.kernel.log.Jdk14LogImpl info
[java] INFO: Portal lib directory /D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/
[java] 09:05:48,470 INFO [EasyConf:122] Refreshed the configuration of all
components
[java] 09:05:48,770 INFO [ConfigurationLoader:56] Properties for jar:file:/D:/liferay-
portal-6.1.0-ce-ga1/tomcat-7.0.23/webapps/ROOT/WEB-INF/lib/portal-impl.jar!/portal
loaded from [jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/portal-
impl.jar!/com.liferay.portal/tools/dependencies/portal-tools.properties, file:/D:/liferay-
portal-6.1.0-ce-ga1/tomcat-7.0.23/webapps/ROOT/WEB-INF/classes/portal-
ext.properties, jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/portal-impl.jar!/portal.properties]
[java] Loading jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/portal-impl.jar!/portal.properties
[java] Loading file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-7.0.23/webapps/ROOT/WEB-
INF/classes/portal-ext.properties
[java] Loading jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/portal-
impl.jar!/com.liferay.portal/tools/dependencies/portal-tools.properties
[java] SLF4J: Class path contains multiple SLF4J bindings.
[java] SLF4J: Found binding in [jar:file:/D:/liferay-portal-6.1.0-ce-ga1/tomcat-
7.0.23/webapps/ROOT/WEB-INF/lib/util-
java.jar!/org/slf4j/impl/StaticLoggerBinder.class]
[java] SLF4J: Found binding in [jar:file:/D:/liferay-plugins-sdk-6.1.0-ce-
ga1/portlets/TestPlugin-portlet/docroot/WEB-INF/lib/util-
java.jar!/org/slf4j/impl/StaticLoggerBinder.class]
[java] SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
explanation.
[java] Java2WSDL com.liferay.test.service.http.EmployeeServiceSoap
[java] 09:05:52,085 WARN [Types:1704] The class com.liferay.test.model.Employee
does not contain a default constructor, which is a requirement for a bean class. The
class cannot be converted into an xml schema type. An xml schema anyType will be
used to define this class in the wsdl file.
[java] WSDL2Java D:/liferay-plugins-sdk-6.1.0-ce-ga1/portlets/TestPlugin-
portlet20120619090551540/service.wsdl
[java] Running Wsdl2javaAntTask with parameters:
[java] verbose:false
[java] debug:false
[java] quiet:false
[java] server-side:true
[java] skeletonDeploy:false
[java] helperGen:false
```

```
[java] factory:null
[java] nsIncludes:[]
[java] nsExcludes:[]
[java] factoryProps:[]
[java] testCase:false
[java] noImports:false
[java] NStoPkg:{}
[java] output😄:\liferay-plugins-sdk-6.1.0-ce-ga1\portlets\TestPlugin-
portlet\20120619090551540
[java] protocolHandlerPkgs:
[java] deployScope:
[java] URL😄:\liferay-plugins-sdk-6.1.0-ce-ga1\portlets\TestPlugin-
portlet\20120619090551540\service.wsdl
[java] all:false
[java] typeMappingVersion:1.2
[java] timeout:45000
[java] failOnNetworkErrors:false
[java] printStackTraceOnFailure:true
[java] namespaceMappingFile:null
[java] username:null
[java] :passwordnull
[java] :noWrappedfalse
[java] :allowInvalidURLfalse
[java] :implementationClassName null
[java] :classpathnull
[java] http.proxyHost=null
[java] http.proxyPort=null
[java] http.proxyUser=null
[java] http.proxyPassword=null
[java] socks.proxyHost=null
[java] socks.proxyPort=null
[java] java.io.IOException: Type {http://model.test.liferay.com}Employee is referenced
but not defined.
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.checkForUndefined(SymbolTable.java:
665)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.add(SymbolTable.java:545)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.populate(SymbolTable.java:518)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.populate(SymbolTable.java:495)
[java] at org.apache.axis.wsdl.gen.Parser$WSDLRunnable.run(Parser.java:361)
[java] at java.lang.Thread.run(Thread.java:619)
```

```
[java] Exception in thread "main" WSDL processing error for D:\liferay-plugins-sdk-6.1.0-ce-ga1\portlets\TestPlugin-portlet\20120619090551540\service.wsdl :
[java] Type {http://model.test.liferay.com}Employee is referenced but not defined.
[java] at
org.apache.axis.tools.ant.wsdl.Wsdl2javaAntTask.execute(Wsdl2javaAntTask.java:304)
[java] at
com.liferay.util.ant.Java2WsddTask.generateWsdd(Java2WsddTask.java:108)
[java] at
com.liferay.portal.tools.WSDDBuilder._createServiceWSDD(WSDDBuilder.java:130)
[java] at com.liferay.portal.tools.WSDDBuilder.build(WSDDBuilder.java:93)
[java] at com.liferay.portal.tools.WSDDBuilder.main(WSDDBuilder.java:49)
[java] Caused by: java.io.IOException: Type {http://model.test.liferay.com}Employee is referenced but not defined.
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.checkForUndefined(SymbolTable.java:665)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.add(SymbolTable.java:545)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.populate(SymbolTable.java:518)
[java] at
org.apache.axis.wsdl.symbolTable.SymbolTable.populate(SymbolTable.java:495)
[java] at org.apache.axis.wsdl.gen.Parser$WSDLRunnable.run(Parser.java:361)
[java] at java.lang.Thread.run(Thread.java:619)
[java] Java Result: 1
```

Have you any suggestions to solve the problem?



**Petr Vašek**  
7/12/12 2:53 AM

---

I have same problem. Can you tell me solution?

Thak you



**Petr Vašek**  
7/14/12 2:36 AM


---

For all , problem is in implement of XXXLocalServicesImp class. IN version liferay 6.1 dont necesarry correct methods in XXXServiceSoap.

->

[http://www.liferay.com/community/forums/-/message\\_boards/message/14855175](http://www.liferay.com/community/forums/-/message_boards/message/14855175)



Powered by Liferay Portal 

[PRIVACY POLICY](#)

[SITEMAP](#)

[CONTACT US](#)

[ABOUT US](#)

[CAREERS](#)

© 2012 LIFERAY INC. ALL RIGHTS RESERVED