

TUTORIAL TEST WEB FRAMEWORK ZEROCOUPLAGE

I-DESCRIPTION OF THE FRAMEWORK

The Framework Zerocouplage can develop different type Application with a single layer of business. Hence the name Zerocouplage signifying zero coupling between the presentation layer and the business layer. When developing application it will be necessary to import the following jar files:

Zerocouplage .common-1.0.0 .jar: It contains tools treatments reflections for the entire Framework.

Zerocouplage.api.1.0.0.jar : the jar containing the interfaces Framework Zerocouplage.

ZerocouplageImpl-1.0.0.jar: classes that implement the interfaces of the Framework.

Zerocouplage.validators-1.0.0.jar : classes for validating fields view .

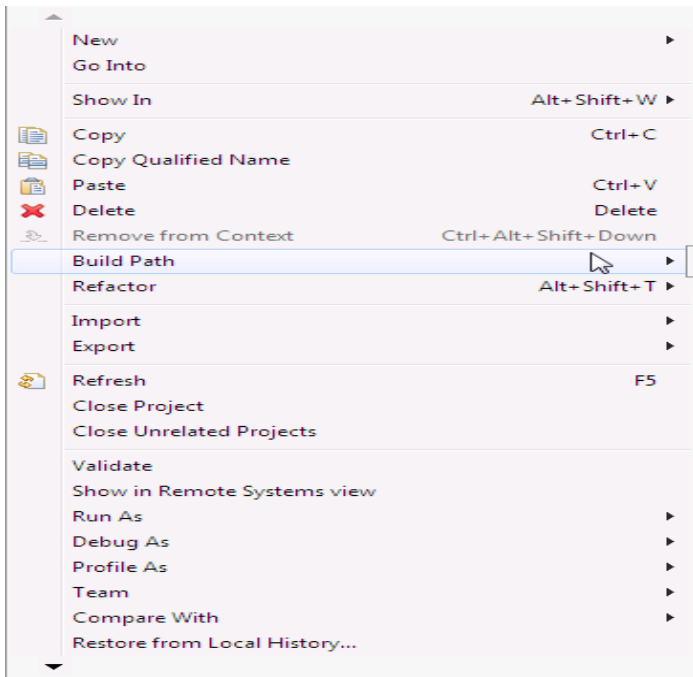
Zerocouplage-web-1.0.0.jar : Class and tools for development web application.

In this tutorial we will develop an application that allows Display Customer information enters from a form, based on the Framework zerocouplage to build this application.

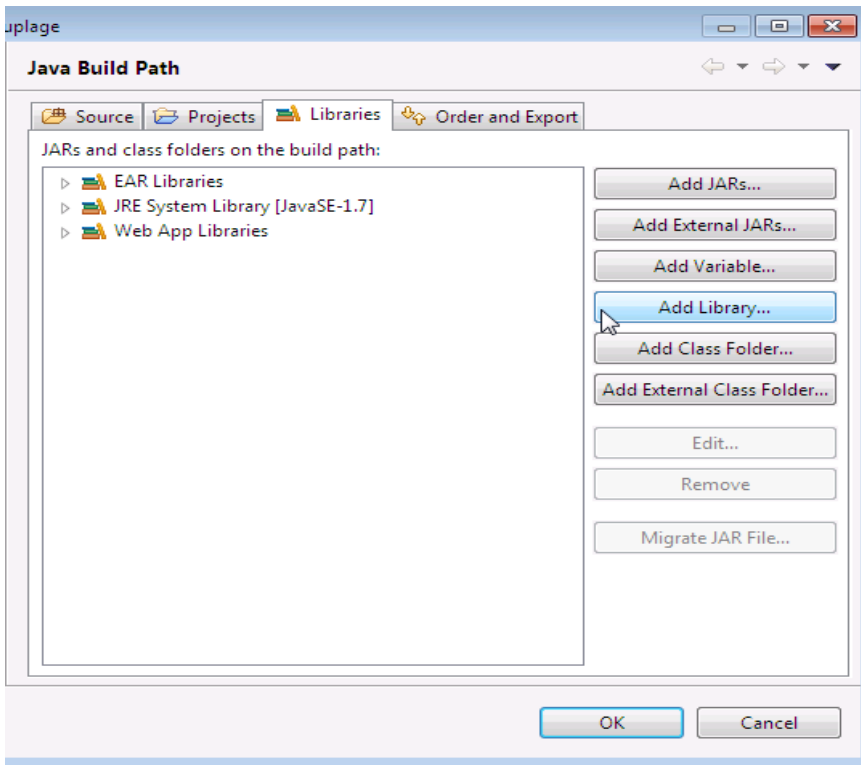
Download jar from this link: .

Unzip the file and then add jar in your project :

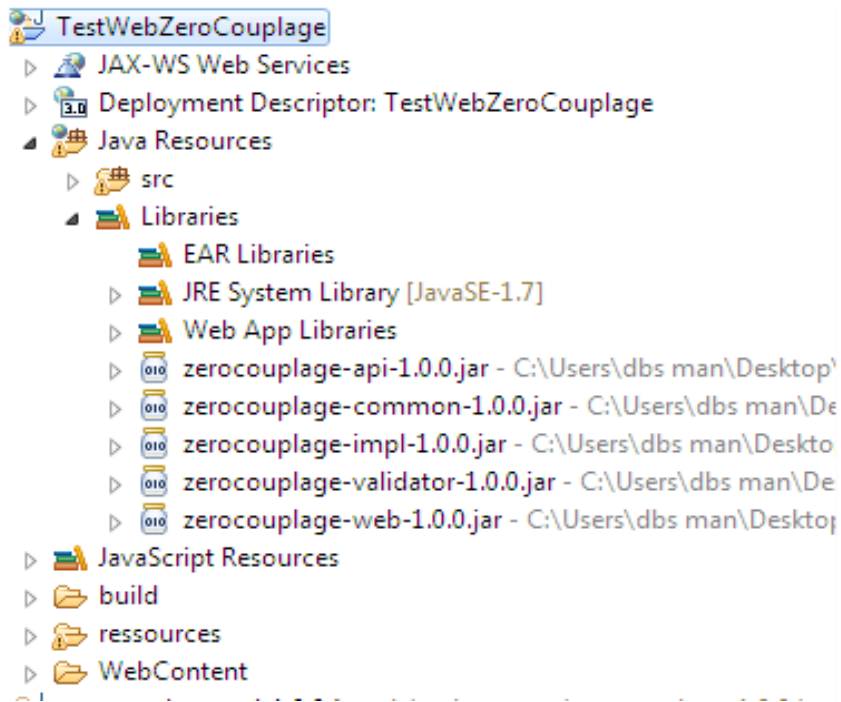
Click right on the project and BuildPath -> configure path -> add External JARs .



Add jar File by clicking add External JARS :

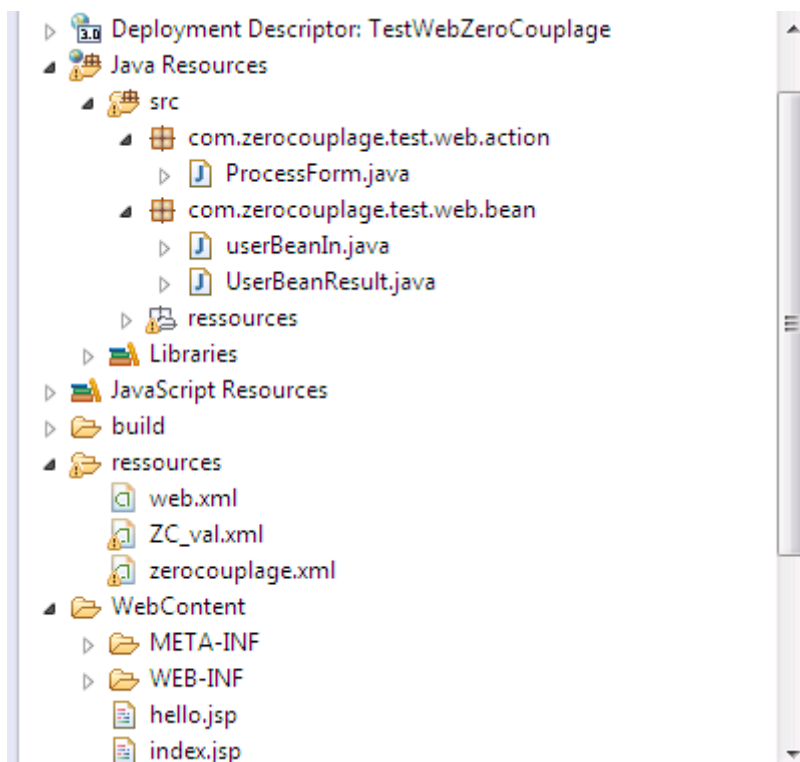


The jars Files will be add at our project like:



The project structure :

- The package `com.zerocouplage.test.web.action` contain the business of the project
- The package `com.zerocouplage.test.web.bean` contain beans of project
- The folder `resource` contain all files configuration like `ZC_val.xml` for validation ,`zerocouplage.xml` and `web.xml` .



The home page of the application formhome.jsp named it allows a customer to enter their first and last name then display information by clicking on Register .

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"> <html>
<head><meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Formulaire de connexion</title>
</head>
<body>
<%
String sFirstName = request.getParameter("firstName");
String sLastName = request.getParameter("lastName");
if(sFirstName == null)sFirstName = "";
if(sLastName == null)sLastName = "";
%>
<font color="red">${GLOBAL_MESSAGE}</font>
<form action="processing.0c" method="POST" >
<input type="hidden" name="_ZC_SOURCE_VIEW_NAME" value="viewI" />
<table>
<tr>
<td>First name:</td>
<td><input type="text" name="firstName" value="<%=sFirstName %>" / <font color="red">${error_nom}
</font> </td>
</tr>
<tr>
<td>Last name</td>
<td><input type="text" name="lastName" value="<%=sLastName %>" /> <font
color="red">${error_prenom}</font> </td>
</tr>
<tr>
.<td colspan="3" align="center"><input type="submit" value="enregistrer" /></td>
</tr>
</table>
</form>
</body>
</html>
```

```
<input type="hidden" name="_ZC_SOURCE_VIEW_NAME" value="view I" />
```

ZC_SOURCE_VIEW_NAME attribute must always be declared by allowing the Framework recognizes all requests from this page. Similarly, it must indicate the name the business responsible for processing the form in our project will be named "processing.oc."

The File validator form allows to respect the constraint of the view fields Firstname and Lastname .It locate to the resource folder .:

File validator form

```
<?xml version="1.0" encoding="UTF-8"?>
<bean-validators>
  <global-message>certains champs sont obligatoires</global-message>
  <bean-validator id="" class="">
    <field name="nom" constraint="[a-zA-Z]+">
      <message>Firstname is invalid</message>
    </field>
    <field name="prenom" constraint="[a-zA-Z]+">
      <message>Lastname is invalid</message>
    </field>
  </bean-validator>
</bean-validators>
```

We will now see page hello.jsp that display customer information :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Page Hello</title>
</head>
<body>
<H1> Hello : ${out.firstname} ${out.lastname} </H1>
</body>
</html>
```

Hello page. Jsp will aim Display informations retrieved from the result returned by the Bean business. The Bean out will enable us View the parameters of our desired form. The view must be declared in the package of views the zerocouplage.xml .

Fichier de configuration Zerocouplage.xml

```
<views-package>
<view name="view" target="hello.jsp" />
</views-package>
```

The business of the project is to perform processing when validating our form. It will consist to execute method Process belonging class ProcessForm

Declared in the configuration file:

Fichier de configuration Zerocouplage.xml

```
<business-package>
  <business name="processing" class="com.zerocouplage.test.web.action.ProcessForm" method="process" >
    <bean-ref ref-bean="UserBeanIn" />
    <validator-ref ref-val="validator" />
    <view-results>
      <view-results name="success" bean-result="out" ref-view="view1" />
    </view-results>
  </business>
</business-package>
```

Class ProcessForm .java :

```
package com.zerocouplage.test.web.action;

import com.zerocouplage.test.web.bean.UserBeanResult;
import com.zerocouplage.test.web.bean.userBeanIn;

public class ProcessForm {

    private UserBeanResult out;

    public UserBeanResult getOut() {

        return out;

    }

    public String process(userBeanIn in) {

        out = new UserBeanResult();

        out.setFirstname(in.getNom());

        out.setLastname(in.getPrenom());

        out.setFullname(in.getNom() + in.getPrenom() );

        return "success";

    }

}
```

The process method receives as argument the Bean recovery information from the form. Performs processing and fills the result which will be the Bean Object to display different information processing outputs.

Bean recovery information from the form captures:

```
package com.zerocouplage.test.web.bean;

public class userBeanIn {

    private String nom;

    private String prenom;

    public String getNom() {return nom; }

    public void setNom(String nom) {this.nom = nom;}

    public void setPrenom(String prenom) {this.prenom = prenom;}

    public String getPrenom() {return prenom ;}}
```

Result Bean container outputs information of business:

```
package com.zerocouplage.test.web.bean;

public class UserBeanResult {

    private String firstname;

    private String lastname;

    public String getFirstname() {return firstname; }

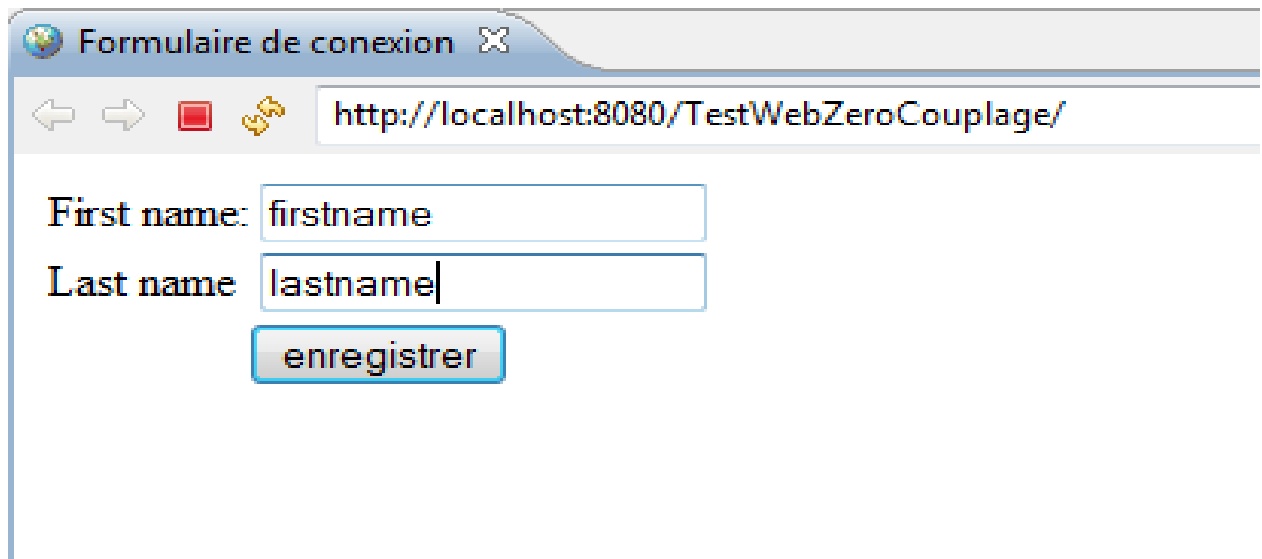
    public void setFirstname(String firstname) {this.firstname = firstname;}

    public String getLastname() {return lastname;}

    public void setLastname(String lastname) {this.lastname = lastname; }

}
```


FormHome.jsp



The screenshot shows a web browser window with the title 'Formulaire de connexion'. The address bar displays 'http://localhost:8080/TestWebZeroCouplage/'. The form contains two text input fields: 'First name' with the value 'firstname' and 'Last name' with the value 'lastname'. Below these fields is a button labeled 'enregistrer'.

Hello.jsp



The screenshot shows a web browser window with the title 'Page Hello'. The address bar displays 'http://localhost:8080/TestWebZeroCouplage/processing.0c'. The main content of the page is the text 'Hello : firstname lastname'.

The full configuration of the projet File Zerocouplage.xml :

```

<?xml version="1.0"?>
<!-- xml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="zcouplage.xsd"          <zcouplage DevModes="prod"
context="web" >
<!-- mapping package -->
<mapping-view-bean>
<mapping isSame="false"    ref-bean="UserBeanIn"    ref-view="viewI" >
<key  beanProperty="nom"    viewProperty="firstName"/ >
<key  beanProperty="prenom" viewProperty="lastName" />
</mapping>
</mapping-view-bean>
<!-- validators package -->
<validators-package>
<validator name="valForm" typeVal="file"    valueTarget="ressources/ZC_val.xml" />
</validators-package>
<!-- beans-package -->
<beans-package>
<bean name="UserBeanIn" class="com.zerocouplage.test.web.bean.userBeanIn" />
</beans-package>
<!-- Business package -->
<business-package>
<business name="processing" class="com.zerocouplage.test.web.action.ProcessForm"
method="process" >          <bean-ref ref-bean="UserBeanIn" />
<validator-ref ref-val="valForm" />
<view-results>
<view-result name="success" bean-result="out"    ref-view="viewR" /></view-results>
</business>
</business-package>
<!-- Views package -->
<views-package>
<view name="viewI" method="" methodError="" target="hello.jsp" />
    <view name="viewI" method="createGui" methodError="processError"
target="com.zerocouplage.test.desktop.view.UserView" />
</views-package>
<!-- Add packages here -->
</zcouplage>
</xml>

```

II-CONCLUSION:

The framework zerocouplage is a tool favorable for the development of web application, desktop or other .This will require simple configuration files zerocouplage.xml et ZC_val.xml. This tutorial aims at initialization zerocouplage framework. For the development of more complex applications please visit: www.zerocouplage.com