The process of back-end custom development

- Back-end custom development mainly customizes new resources module, with the following two ways:

Case 1. Develop new resources based on iPortal

- The example code used in this example is located in:% SuperMap iPortal_HOME% \ samples \ code \ CustomPortal \ Custom_Portal.
- The IportalHelloWorldResource.Java file (located under the **src** > **com.supermap.iportal.web.custom.rest.resource.impl** package) is the root entry for customizing the new resource, the IportalServletHandler.java file (located at **src** > **com.supermap.server.host. webapp.handlers** package below) is the route filter, as follows:

```
@Override
public void handle(HttpServletRequest req, HttpServletResponse
 res) throws ServletException,  IOException{
String path = getPathInfo(req);
if(path.startsWith("/helloworld")){
WebAppRequestDispatcher dispatcher = new WebAppRequestDispatcher("/helloworld",
 this.iportalServlet);
dispatcher.forward(req, res);
}
else{
return;
}
this.setHandleFinished(req);
}
```

- Here, the root entry path for customizing new resources is set to "/ helloworld". The root entry path can be set to the desired path name according to their own needs. When sending http: // localhost: 8091 / iportal / helloworld, it will enter into the IportalHelloWorldResource.Java portal resource, you can set a new child resource in resource manager as follows:

```
@Path("/")
public class IportalHelloWorldResource extends IportalJaxrsResourceBase{
//International Resource Management
private static ResourceManager resource = new ResourceManager("");
public Object iportalIndex(@Context HttpServletRequest
 request){
List<ChildResourceInfo> childInfos = new ArrayList<ChildResourceInfo>();
String rootUrl = this.getResourceURL(request);
childInfos.add(createChildResourceInfo(rootUrl, "registers"));
return request;
}
private ChildResourceInfo createChildResourceInfo(String
 rootUrl,  String resName){
ChildResourceInfo info = new ChildResourceInfo();
info.name = resName;
info.path = rootUrl + resName;
return info;
}
@GET
@Path("/register")
@Template(name="iportalRegister.ftl")
public Map<String, Object> singleHelloWorld(){
return new HashMap<String, Object>();
}
@Path("/custom")
public IportalCustomResource getIportalCustomResource(){
return new IportalCustomResource();
}
}
```

- Where "/ custom" points to the new sub-resource IportalCustomResource, you can add new resources in the new sub-resources, such as when sending http: // localhost: 8091 / iportal / helloworld / custom request will be assigned to the resources In IportalCustomResource, the default GET request is specified to the resource in the IportalCustomResource resource with the following code:

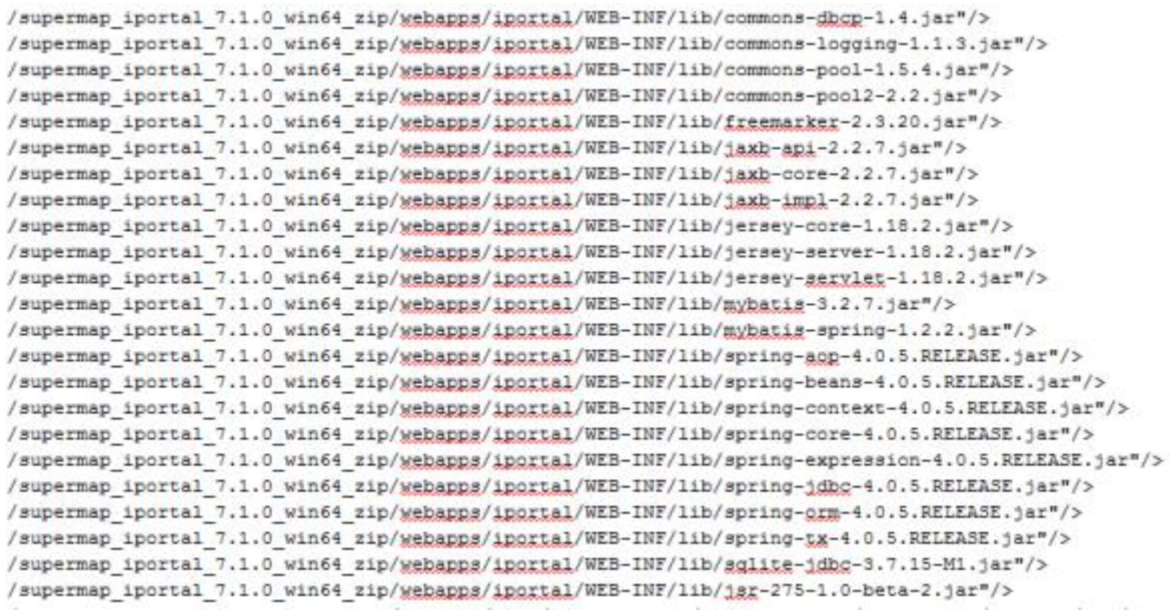
```
public class IportalCustomResource extends IportalJaxrsResourceBase
 {
NewsInfoComponent newsInfoComponent = (NewsInfoComponent)
 beanFactory
.getBean("newsInfoComponent");
@GET
@Template(name = "newsInfo.ftl")
public NewsInfo getResource(@Context HttpServletRequest
 request) {
System.out.println("Access to resources... ...") ;
NewsInfo newsInfo = new NewsInfo();
int id=2;
//newsInfo = newsInfoComponent.getNewsInfo(id);
return newsInfo;
}
```

- Send a GET request for http: // localhost: 8091 / iportal / helloworld / custom.rjson with the result:

```
{
  "addTime": null,
  "addUser": "admin",
  "commentInfo": null,
  "content": null,
  "id": null,
  "modifyTime": null,
  "modifyUser": null,
  "newsIcon": null,
  "newsTitle": "test1"
}
```

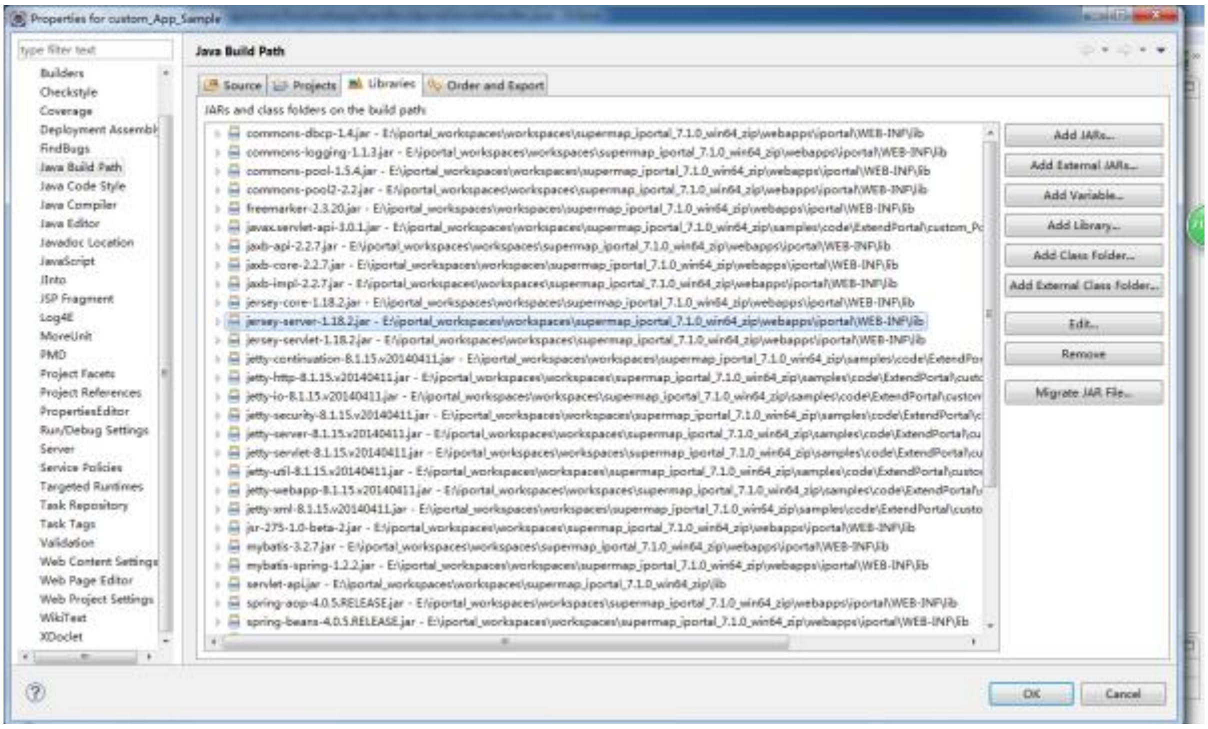
Case 2. Customize new resources based on Jersey

- 1. Find the customized project: Custom_App_Sample in the path: %SuperMap iPortal_HOME%\samples\code\CustomPortal\
- 2. Open the customized develop project in Eclipse (Notes: before this step, please first run LibReplace.bat under path: %SuperMap iPortal_HOME%\samples\code\. You only need to run this file for one time. If it is running, there is no need to start again), click File, then click Import. Select General and expand it on the Import tab, click Existing Projects into Workspace, then click Next, find Custom_App_Sample under: %SuperMap iPortal_HOME%\samples\code\CustomPortal\, click Finish to complete it.
- About how to configure the custom project, please refer to [Configure Projects](). If you have already done the project configuration, you can proceed directly to the following third step.
- 3. Configurate the relying jar package. In Eclipse, right click Custom_App_Sample, select Build Path > Configure Build Path, click the tab: Libraries in the dialog box, all the jar needed will be shown in the down side, if is shown in this item, it means this jar is not found, you need to click the Add External JARs, select the jar under path: %SuperMap iPortal_HOME%\samples\code\CustomPortal\Custom_Portal \WebContent\WEB-INF\lib, or servlet-api.jar under path: %SuperMap iPortal_HOME%\lib, or jar under path: %SuperMap iPortal_HOME%\webapps\iportal\WEB-INF\lib, shown as below in the figure:



```
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/commons-dbcp-1.4.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/commons-logging-1.1.3.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/commons-pool-1.5.4.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/commons-pool2-2.2.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/freemarker-2.3.20.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jaxb-api-2.2.7.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jaxb-core-2.2.7.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jaxb-impl-2.2.7.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jersey-core-1.18.2.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jersey-server-1.18.2.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jersey-servlet-1.18.2.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/mybatis-3.2.7.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/mybatis-spring-1.2.2.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-aop-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-beans-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-context-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-core-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-expression-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-jdbc-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-orm-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/spring-tx-4.0.5.RELEASE.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/sqlite-jdbc-3.7.15-M1.jar"/>
/supermap_iportal_7.1.0_win64_zip/webapps/iportal/WEB-INF/lib/jsr-275-1.0-beta-2.jar"/>
```

- Click OK, as shown in the figure below:



Properties for custom_App_Sample

Java Build Path

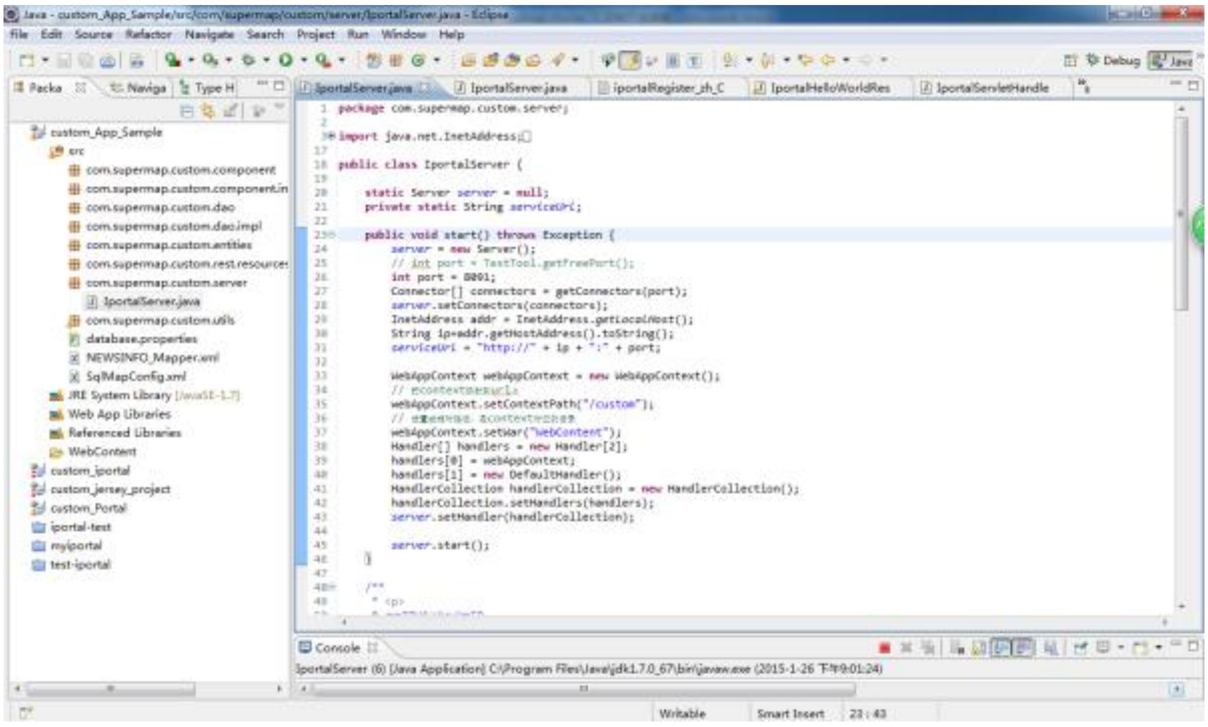
Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

```
commons-dbcp-1.4.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
commons-logging-1.1.3.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
commons-pool-1.5.4.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
commons-pool2-2.2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
freemarker-2.3.20.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
javax.servlet-api-3.0.1.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\custom_Po
jaxb-api-2.2.7.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jaxb-core-2.2.7.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jaxb-impl-2.2.7.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jersey-core-1.18.2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jersey-server-1.18.2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jersey-servlet-1.18.2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
jetty-continuation-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPor
jetty-http-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\custo
jetty-io-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\custom
jetty-security-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\c
jetty-server-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\cu
jetty-servlet-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\cu
jetty-util-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\custo
jetty-webapp-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\
jetty-xml-8.1.15.v20140411.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\samples\code\ExtendPortal\custo
jsr-275-1.0-beta-2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
mybatis-3.2.7.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
mybatis-spring-1.2.2.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
servlet-api.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\lib
spring-aop-4.0.5.RELEASE.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
spring-beans-4.0.5.RELEASE.jar - E:\iportal_workspaces\workspaces\supermap_iportal_7.1.0_win64_zip\webapps\iportal\WEB-INF\lib
```

- 4. In Eclipse, right click project： Custom_App_Sample， chose Build Path > Configure Build Path, in the dialog box, click Libraries tab, drag the down side scroll bar and select JRE System Library, then click the right side Edit button, chose configurated JDK version of your PC, click Finish.

- 5. Open IportalServer.java (Java Resources > src > com.supermap.custom.server) of project Custom_App_Sample in Eclipse, shown as below:



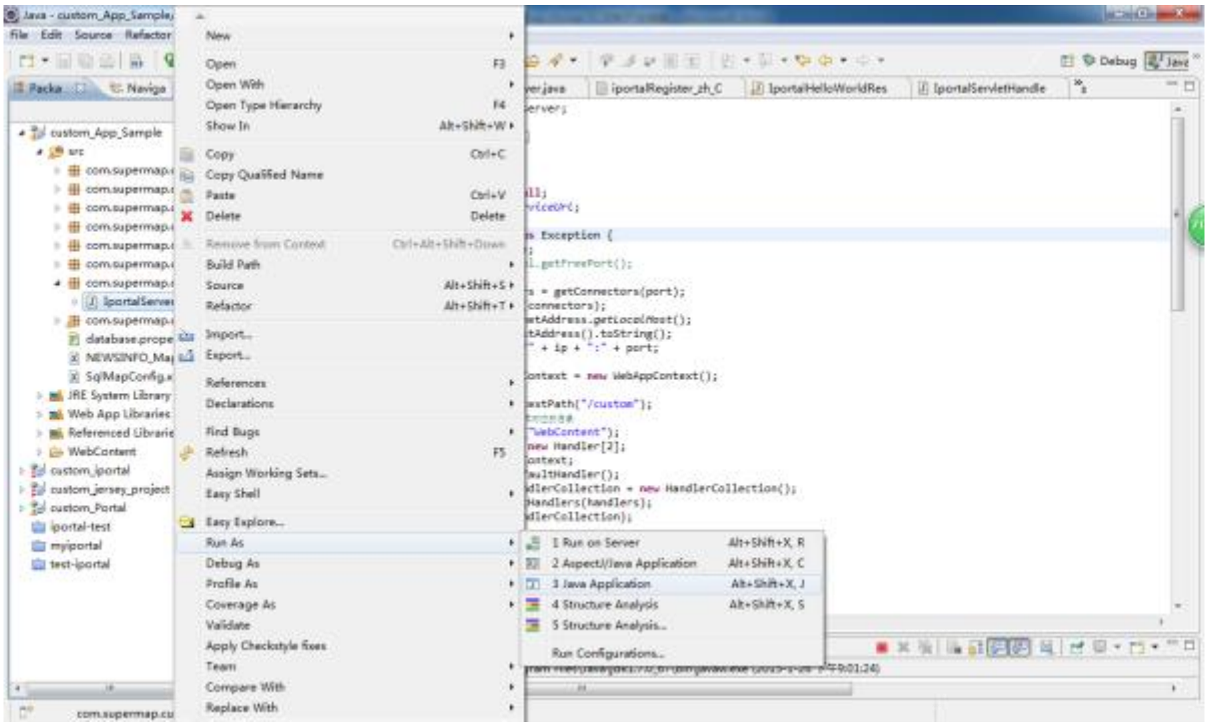
Java - custom_App_Sample/src/com/supermap/custom/server/IportalServer.java - Eclipse

```
package com.supermap.custom.server;

import java.net.InetAddress;

public class IportalServer {

    static Server server = null;
    private static String serviceUrl;

    public void start() throws Exception {
        server = new Server();
        // int port = TestTool.getFreePort();
        int port = 8081;
        Connector[] connectors = getConnectors(port);
        server.setConnectors(connectors);
        InetAddress addr = InetAddress.getLocalHost();
        String ip=addr.getHostAddress().toString();
        serviceUrl = "http://" + ip + ":" + port;

        WebAppContext webAppContext = new WebAppContext();
        // [illegible]
        webAppContext.setContextPath("/custom");
        // [illegible]
        webAppContext.setWar("WebContent");
        Handler[] handlers = new Handler[2];
        handlers[0] = webAppContext;
        handlers[1] = new DefaultHandler();
        HandlerCollection handlerCollection = new HandlerCollection();
        handlerCollection.setHandlers(handlers);
        server.setHandler(handlerCollection);

        server.start();
    }

    /**
     * <p>
```

```
IportalServer (6) [Java Application] C:\Program Files\Java\jdk1.7.0_67\bin\javaw.exe (2015-1-26 下午9:01:24)
```

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

- Right click IportalServer.java, then click Run as > Java Application , successfully started the customized service.



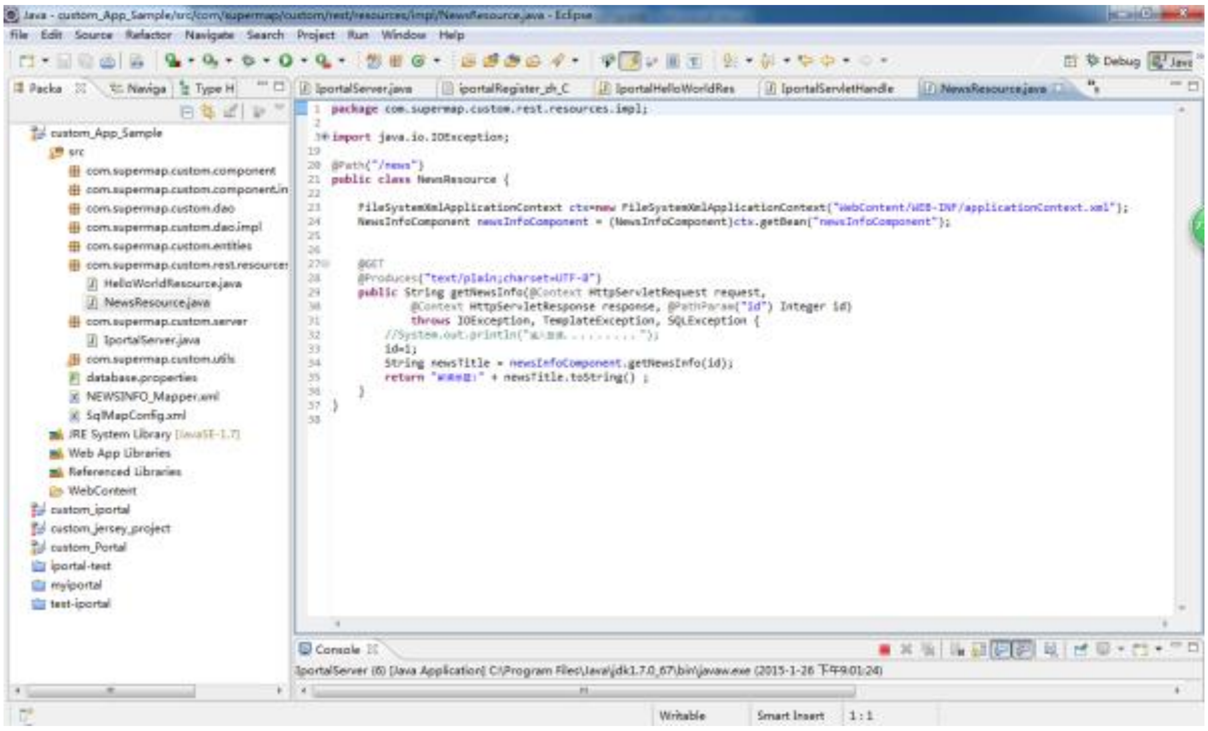
- 6. Modify the directed path of the database, copy the custom.db under：%SuperMap iPortal _HOME%\samples\code\CustomPortal\Custom_App_Sample\WebContent\WEB-INF to a fixed path, then find \Custom_App_Sample\WebContent\WEB-INF\classes\applicationContext.xml, directed to the path where the database files are, shown as below:

```
<property name="url" value="jdbc:sqlite:E:\custom.db"></property>
```

- **Please restart the customized service.**
- 7. Configure the root path of the service as custom in the class: IportalServer.java

```
webAppContext.setContextPath("/custom");
```

- The root path of the service is: http://localhost：8091/custom.
- 8. NewsResource.java (Java Resources > src > com.supermap.custom.rest.resource.impl) is the newly added customized resource management tool class, click http://localhost:8091/custom/news to direct the resource: NewsResource, as shown below:



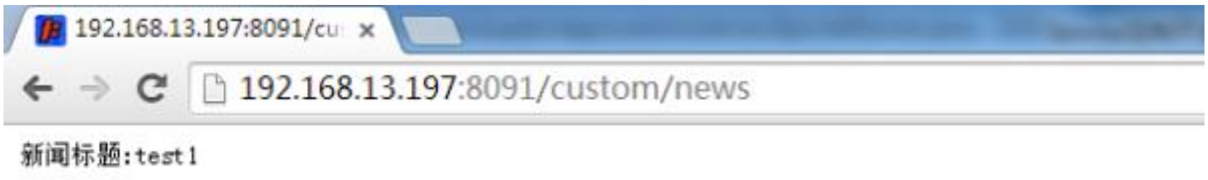
```
package com.supermap.custom.rest.resources.impl;

import java.io.IOException;

@Path("/news")
public class NewsResource {

    FileSystemXmlApplicationContext ctx=new FileSystemXmlApplicationContext("WebContent/WEB-INF/applicationContext.xml");
    NewsInfoComponent newsInfoComponent = (NewsInfoComponent)ctx.getBean("newsInfoComponent");


    @GET
    @Produces("text/plain;charset=UTF-8")
    public String getNewsInfo(@Context HttpServletRequest request,
            @Context HttpServletResponse response, @PathParam("id") Integer id)
            throws IOException, TemplateException, SQLException {
        //System.out.println("进入资源.........");
        id=1;
        String newsTitle = newsInfoComponent.getNewsInfo(id);
        return "新闻标题:" + newsTitle.toString() ;
    }
}
```

```
IportalServer (6) [Java Application] C:\Program Files\Java\jdk1.7.0_67\bin\javaw.exe (2015-1-26 下午9:01:24)
```

- You can send the GET request: http://localhost:8091/custom/news and assign it to the GET resource of the NewsResource. As shown below:

```
@Path("/news")
public class NewsResource {
FileSystemXmlApplicationContext ctx=new
FileSystemXmlApplicationContext("WebContent/WEB-INF/applicationContext.xml");
    NewsInfoComponent newsInfoComponent
 = (NewsInfoComponent)ctx.getBean("newsInfoComponent");
@GET
@Produces("text/plain;charset=UTF-8")
public String getNewsInfo(@Context HttpServletRequest
 request,
@Context HttpServletResponse response,  @PathParam("id")
 Integer id)
throws IOException,  TemplateException,  SQLException
 {
//System.out.println("Enter resource...") ;
id=1;
String newsTitle = newsInfoComponent.getNewsInfo(id);
return "News Title:" + newsTitle.toString() ;
}
}
```

- Responding result is as below:



```
192.168.13.197:8091/custom/news

新闻标题:test1
```

-