Extending existing REST resource

JAX-RS-based extension

SuperMap iServer not only provides REST services, publishing a large number of GIS functions as resource, but also provides an extension mechanism, adding users' applications to SuperMap iServer server, which can implement the custom resources to be a part of REST services.

SuperMap iServer has two ways to provide REST services: based on Restlet and JAX-RS. When extending, you should use different extending methods according to the implementation of function modules.

The REST resource module of SuperMap iServer implemented based on the JAX-RS is the spatial analyst module.

JAX-RS introduction

JAX-RS--Java API for RESTful Web Services, is a standard defined by the SUN Microsystems, providing the Java programmers with a set of fixed interfaces (Java API) to develop RESTful WEB service applications, which helps avoid relying on the third-party framework. Meanwhile, JAX-RS, based on the POJO programming model, uses annotations to simplify the development and deployment of web service clients and endpoints. REST is a lightweight Web service architecture and JAX-RS regulates the interfaces for REST application development

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

The common JAX-RS implementation methods are as follows:

- Jersey--The JSR311 reference implementation released by the SUN Microsystems with the JSR311 release;
- CXF--The merging of XFire and Celtix
- RESTEasy--The JAX-RS project of JBoss

Jersey is the JSR311 implementation released by the SUN Microsystems with the JSR311 release; Under the support of SUN, the usability of Jersey is much better than that of the other two methods at present, and is used more widely. SuperMap iServer refers to Jersey for resource implementation.

Samples

Construct the root resource HelloWorldResource, which is a simple Web resource. Its URI is /helloworld and it supports the GET method and the representation in "text/plain". The 3 main elements of JSR311 consists of the resource URI, the HTTP request method and the output format. After the resource has been deployed as the corresponding Web service, you can access it through http://localhost:8090/helloworld.

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

```
package com.sun.ws.rest.samples.helloworld.resources;

import javax.ws.rs.GET;
import javax.ws.rs.Produces;
import javax.ws.rs.Path;
// The Java class can be located by URI "/helloworld", such as http://localhost:8090/helloworld
@Path("/helloworld")
public class HelloWorldResource {
  // Use the HTTP request method GET
  @GET
  // The Java method will return the value of the media type here.
  // The return type is "text/plain".
  @Produces("text/plain")
  public String getClichedMessage() {
     // Return the string.
     return "Hello World";
  }
}
```

Introduction of main annotations

JAX-RS provides a series of annotations, and encapsulates the resource classes and the corresponding methods into the Web resource. The main annotations are as follows:

- @Path, annotates the relative path of the resource class or the method;
- @GET, @PUT, @POST and @DELETE, annotate the HTTP request method type;
- @Produce, annotates the returned data format (MIME type), namely the type of the returned value of the method;
- @Consumes, annotates the acceptable request data format (MIME type), namely the parameter type decoded by the resource;
- @PathParam, @QueryParam, @HeaderParam, @CookieParam, @MatrixParam, and @FormParam, annotate the where of the HTTP request does the parameter of

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

the methode come from. For example, @PathParam represents URL path, @QueryParam represents URL query parameter, @HeaderParam represents the header of the HTTP request, and @CookieParam represents the Cookie of the HTTP request.

Class-level annotations @Path and @Produces are applicable to each method, unless the method has its own @Path and @Produces.

Parameter passing method

There are two types of resource parameters: the URI query parameter annotated with @PathParam and the parameter in the request body annotated with @FormParam.

1. The URI query parameter is annotated with @PathParam, and userName is a URI parameter.

```
1 @Path("/users/{username}")
2 public class UserResource {
3
4    @GET
5    @Produces("text/xml")
6    public String getUser(@PathParam("username") String userName) {
7        ...
8    }
9 }
```

The parameter in the request body is annotated with @FormParam. Taking the POST request as an example, its definition method is as follows. Parameter name is one of this type.

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

```
1 @POST
2 @Consumes("application/x-www-form-urlencoded")
3 public void post(@FormParam("name") String name) {
4     // Request the information
5     }
```

SuperMap iServer provides the resource extension mechanism based on the JAX-RS framework (refer to **JAX-RS Introduction**), which is used for the REST service extension:

1. It provides the resource base class: JaxrsResourceBase and JaxAlgorithResultSetResource<T>, for users to extend and implement the new resource.

2. SuperMap iServer provides several output formats like xml, json and rjson. Resource created based on the base class support the output formats by default.

3. It provides the @Template annotation, and a series of FreeMarker variables, for users to customize the *.ftl template and configurate it to the resource. Therefore, the resource can support the html output format.

4. It provides the JSON decoder. The default identification type of the resource created based on the resource base class is the JSON format.

5. It provides the service interface context (InterfaceContext). The resource created based on the base class can get the service component and other information through InterfaceContext.

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

6. The domain functionalities are tightly attached to the three-layer architecture of SuperMap iServer, which induces the creation of domain resources through the combination with the current GIS functionalities.

Therefore, the REST resource extensions based on JAX-RS include:

- **Extending a new Resource**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > JAX-RS-based extension > Extending a Resource

- **Extending an Encoder**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > JAX-RS-based extension > Extending an Encoder

- **Extending a Decoder**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > JAX-RS-based extension > Extending a Decoder

The extend flow is as follows:

1. Code implementation.

New resources implement based on com.supermap.services.rest.resources.JaxAlgorithResultSetResource or com.supermap.services.rest.resources.JaxrsResourceBase;

The Encoder implement based on javax.ws.rs.ext.MessageBodyWriter;

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

The Decoder implement based on javax.ws.rs.ext.MessageBodyReader.

2. Configuration

- Extending the Encoder (consistent with **Extending the Encoder for existing modules)**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > JAX-RS-based extension > Extending an Encoder

- Extending the Decoder (consistent with **Extending the Decoder for existing modules**)

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > JAX-RS-based extension > Extending a Decoder

Here it introduces FakeKMLEncoder, FakeJsonDecoder and myBuffer. The source codes locate in: %SuperMap iServer_HOME%\samples\code\ExtendExist_JSR. It is an overall project file. Users only need to import this project and compile a JAR (See **extendexist_jsr.jar**). Then put Jar into %SuperMap iServer_HOME%\webapps\iserver\WEB-INF\lib,and restart iServer.

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