Extending existing REST resource

Restlet-based extension

SuperMap iServer not only provides REST services and publishes a large number of GIS functionalities as resource, but also provides a suit of extension mechanism, which can help users add their own applications to SuperMap iServer, implement custom resources and publish domain services.

Currently, SuperMap iServer has two methods to provide REST service, namely the mechanism based on Restlet and JAX-RS. When performing the extension, you should adopt the different extension methods according to the implementation method of functional modules.

The modules that implement REST resource based on the Restlet mechanism have: map module, data module, transporation analyst module and 3D module. SuperMap iServer provides the following extensions:

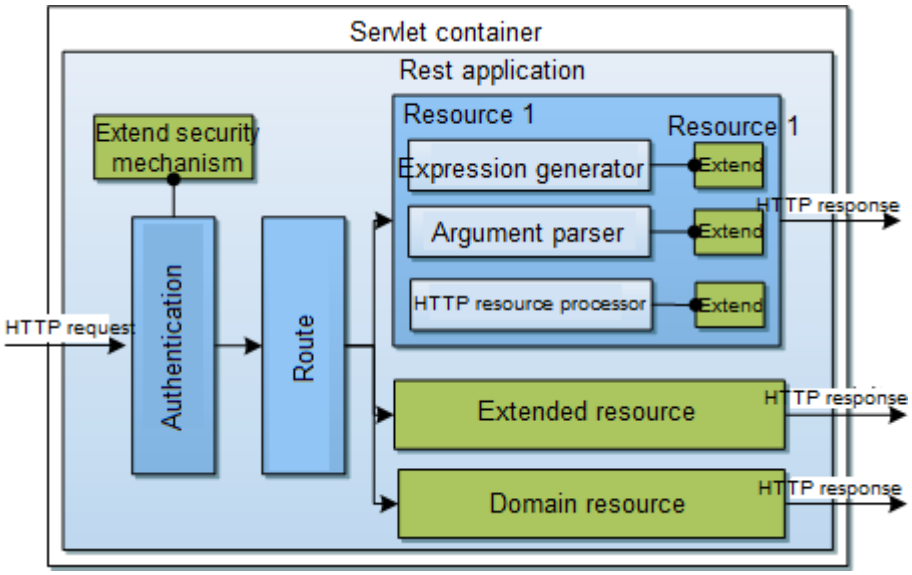
- Extending resource by using REST SDK and inheriting the abstract resource class provided in SDK.
- Extending the encoder. The resource on the server can be published in new output formats.
- Extending the decoder. The server can recgonize new parameter passing formats.

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

- Extending HTTP Handler. The processing procedure of the HTTP request on the server can be customized.

- Extending security. Users can configure their own security mechanism to the server.

Below shows how an HTTP request sent by a client to access SuperMap iServer REST service is processed on SuperMap iServer. In green is places where the extension can be performed.



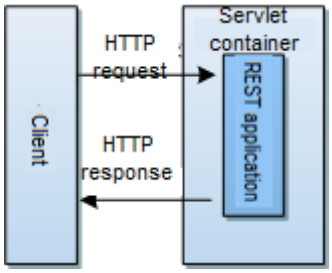
Encoder extension, decoder extension, and HTTP request extension can be performed for original resources on the server and those resources formed by extending through REST SDK or publishing domain components.

Follow the links below to know how to extend REST services.

- **REST service publishing mechanism**

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

SuperMap iServer publishes REST services through Restlet. Firstly, construct a REST application, and then deploy the application object in an HTTP environment, such as Servlet container, OSGI core, Spring, Guice Ioc container, independent JVMS, Groovy, Scala, JAIN/SLEE, etc. Currently, the application object is deployed in the Servlet container. After that, SuperMap iServer can publish REST services through the REST application object, as illustrated below.

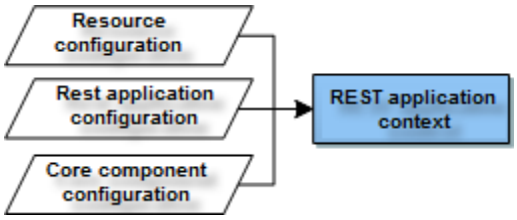


As shown above, an HTTP request firstly arrives at the HTTP environment, then the HTTP environment passes the request to the REST application object, the corresponding class of which inherits from Application in Restlet. The HTTP request will be handled according to the context and the response will be sent back to the client through the HTTP environment.

When an HTTP request arrives at the REST application object, the application object will search the resource configuration information in the REST application context and compare the URI for the HTTP request with those templates. The best matched resource will be found and the implementation of the resource is expected to handle the HTTP request. The procedure is as follows:

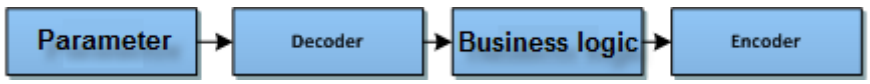
Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

The REST application context contains the information about the resource configuration, REST application configuration, core component configuration, etc. needed by the REST application. The REST application context is obtained from the configuration file when starting the server.



The resource configuration information includes the resource ID, resource type, URI template, implementation class, etc. Resources are implemented through the REST application object and are published in the form of the corresponding URI templates.

The implementation class of a resource is used to handle all operations on the resource. Each implementation class has an HTTP MethodHandler, which aims at managing the HTTP request handling. The HTTP request handling has the following steps:



1. Extract parameters from the HTTP request (it is possible that there are no parameters in the HTTP request);

2. Get the correct Decoder according to the HTTP request header, parsing the parameters in the HTTP request into Java objects;

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

3. Handle the business logic according to the parameters (there can be no parameters) and return the result;

4. Obtain the correct Encoder according to URI, encode the result and write the result into the HTTP response object.

In short, when an HTTP request arrives at the REST application object, a proper resource will be found to handle the request. After that the result will be returned to the client through the HTTP environment as an HTTP response.

- **Extending a Simple Algorithm**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > Restlet-based extension > Extending a Simple Algorithm

- **Extending an Encoder**

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

- **Extending a Decoder**

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

- **Extending an HTTP Request Handler**

To view: Home > Developer guide > Extending iServer > Extending existing REST resource > Restlet-based extension > Extending an HTTP Request Handler

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

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