

Grails + Metro - Contract First Web Service

We've been trying to get full fledged web services working in Grails built in contract first approach, it felt like it should be easy, as all the bits were there, but all of the plugins to date were more proof of concept vs Enterprise ready (e.g. no control over namespaces). After looking at many technologies, decided to give Metro a try.

So, the target was to get Metro running with the following key requirements:

- WSDL First
- Minimal change required to configure the web service
- Bound to Grails services so we can re-use business logic and interact with domain classes

Step 1: Grails App + Metro

Create a simple Grails application (working from the ground up is the best way to explain it, rather than trying to explain our existing application).

Download Metro Grails plugin: <https://jax-ws-commons.dev.java.net/grails/> and install it in grails app.

I used grails-metro-1.0.zip. If you are building the web service bottom-up, you can follow the example on the page above and create web services without further steps.

Since the plugin doesn't have the utility to create java artifacts from the wsdl, download Metro [<https://metro.dev.java.net/>] and unzip it. Note that the bin folder has wsimport scripts which does the reverse engineering.

Step 2: Generate java source files

From the bin folder execute

```
wsimport -s <srcFolder> -d <outputClassesFolder> <WSDLFile>
```

For Example: `wsimport -s src -d classes HelloWorld.wsdl`

This generates all the java classes with the annotations required for the contract first web service. Move the java classes from the src folder to grails app src/java folder.

Step 3: Generate groovy service

Generate the groovy service

```
grails create-service Test
```

Step 4: Add annotations in service groovy file

```

import javax.annotation.Resource;
import javax.jws.WebService;
import javax.xml.ws.WebServiceContext;

@WebService(serviceName = "HelloWorld",
            portName="HelloWorldPort",
            endpointInterface =
"sample.HelloWorld",
            targetNamespace =
"http://mycompany.com/HelloWorld/1.0",
            wsdlLocation =
"http://localhost:<port>/HelloWorld.wsdl")

class TestService implements
sample.HellWorld {

    @Resource
    private WebServiceContext context;

    public sample.HelloWorldRs process(
        sample.HelloWorldRq payload){
        sample.HelloWorldRs res = new
sample.HelloWorldRs();
        //Web service code here
        return res;
    }
}

```

Step 5: Run Grails application

Run the rails application

```
rails run-app
```