



[SOA Governance in Action](#)

REST and WS- Architectures*

By Jos Dirksen

This article based on chapter 2 of [SOA Governance in Action](#) covers some SOA Governance basics by showing how to store a WS-* based service in a registry and then query the registry.

To save 35% on your next purchase use Promotional Code **dirksen0235** when you check out at www.manning.com.

[You may also be interested in...](#)

Setting Up the SOA Registry

In this article, we'll look at how we can store a WS-* based service in a registry and how we can query the registry. The first thing we're going to do is download and install the registry from the WSO2 site.

Running the SOA registry for the first time

Before we can do anything with the registry, we first need to log in. Start the WSO2 registry from `<GOV_HOME>/wso2greg-3.6.0/bin` and point your browser to `https://localhost:9443`. This will start the application and you'll be shown the login screen (see figure 1).

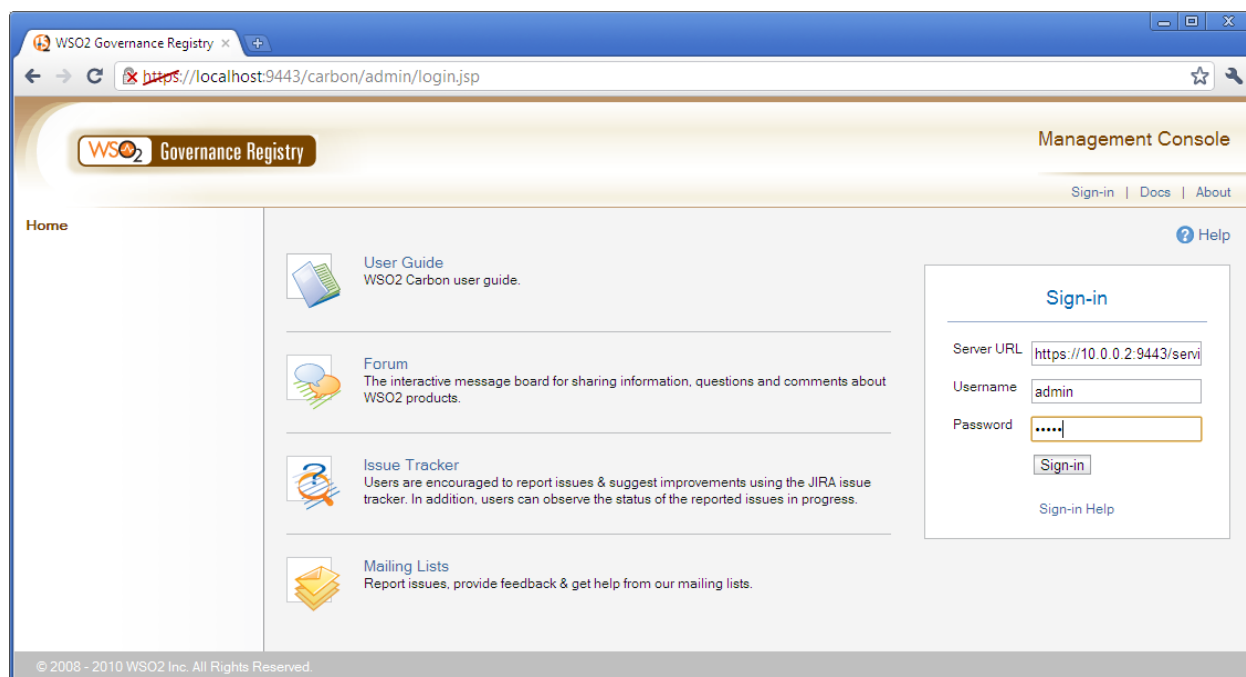


Figure 1 The login page for the WSO2 service registry

For Source Code, Sample Chapters, the Author Forum and other resources, go to <http://www.manning.com/dirksen/>

Enter "admin" for both the username and the password and click on the **Sign-In** button. This will bring you to the governance registry home screen (see figure 2) from where we'll show you how to register your service in this registry.

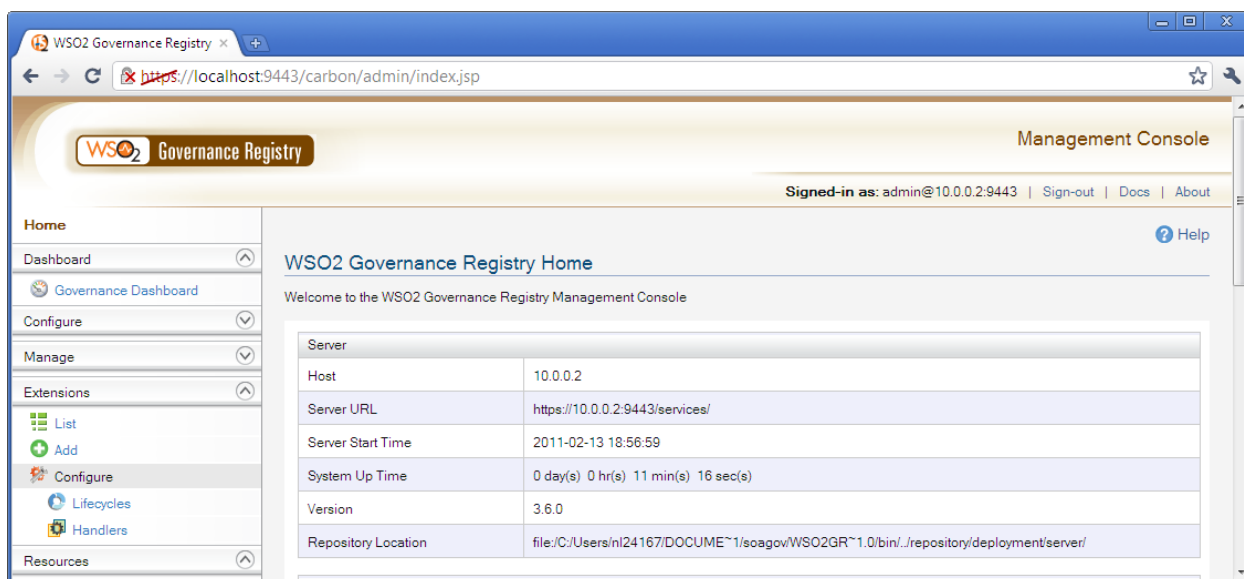


Figure 2 The WS02 Governance registry Home screen

Registering a service manually in the registry

Suppose you created a WSDL that defines your contract. In this part, you'll register this service into the registry so that it can be easily found and reused by others. From the start screen of the registry (see figure 2), select the **Add WSDL** option from the left side (it's under **Metadata**). We use the **Add WSDL** option instead of the **Add Service** option since our WSDL defines our service. So, if we add the WSDL, the registry will automatically define a related service element and an XSD element. In the **Add WSDL** screen (see figure 3), use the option **Import WSDL from url**. In the URL field, add the URL of the `accountService` we created: `http://localhost:9001/accountService?wsdl`. The name will be automatically filled in. Now, click on the **Add** button to add the WSDL and its related resources to the registry.

Figure 3 Add a WSDL to the SOA registry

We'll now have the following three things registered in our registry:

- **Service:** a service with the name `accountService`, the name we specified in the WSDL, is now registered under the **metadata/list/services** menu option.

For Source Code, Sample Chapters, the Author Forum and other resources, go to <http://www.manning.com/dirksen/>

- **WSDL**: the WSDL itself is of course registered. You can find this one in the **metadata/list/WSDLs** menu option.
- **XSD**: we created our schema separately from the WSDL. The registry has detected this from our WSDL and created an entry for this. You can see this in the **metadata/list/schemas**.

If this were a real work scenario, we would now start adding documentation about this service in the registry so that other people would know what our service does, how it should be used or for instance how it should be accessed for a test environment. For now, we'll keep at it this, and look at how we can access this repository, besides using the provided web application.

Accessing the WSO2 Governance registry

In this part, we'll look at the most simple way you can access the information from the repository. This is done by using the client provided by WSO2 for this. This client is called the `WSRegistryServiceClient`. Before we dive into the Java code, let's quickly look at the registry again. Start up the WSO2 Governance registry and log in at <https://localhost:9443>. On the left side of the overview screen, you'll see a tab with the name **Resources**. In the **Resources** tab, click on the **Browse** option and you'll see a tree layout of the repository. All the data you store in the registry will be accessible through this tree. When you open it, you won't see much except a `_system` folder. Open this folder and then click on the `governance` subfolder (see figure 4).



Figure 4 How to browse through the SOA Governance registry

In the `governance` folder, you'll find all the resources related to WSDLs, services, schemas, and so on. Now, click a bit further until you see the WSDL we uploaded. If you click on this resource, you'll find its location (its path) at the top of the screen. For this WSDL this will be something like: `/_system/governance/wsdls/urn:trafficavoidance:wSDL:account:service-1/0/accountService.wsdl`. We'll use this path to look up this WSDL in the example in listing 1.

Listing 1 Shows how you can access the WSO2Registry from java code using the WSRegistryServiceClient

```
public static void main(String[] args) throws Exception {
    WSRegistryServiceClient client;           |#1
    client = initialize();                     |#1
}
```

For Source Code, Sample Chapters, the Author Forum and other resources, go to <http://www.manning.com/dirksen/>

```

// get the governance folder
Resource governanceFolder = client.get("/_system/governance"); #2
System.out.println("Folder description: " + governanceFolder.getDescription()); #2

// get the WSDL folder resource (use the url we browsed to)
String wsdlUrl = "/_system/governance/wsdl:urn: " + #3
    "trafficavoidance:wsdl:account:service-1" + #3
    "/0/accountService.wsdl"; #3
Resource wsdlResource = client.get(wsdlUrl);

// output the content of the wsdl
System.out.println(new String((byte[])wsdlResource #4
    .getContent())); #4
}

```

#1: Initializes the client before using it

#2: Shows the description of a resource

#3: Gets the WSDL using its path

#4: Prints out the content of the WSDL

In this example, you can see that we use the `WSRegistryClient` (#1) to connect to a registry. If you want to access a resource you use the `get` operation from this client. This operation takes a `String`, which is the path you want to access, as a parameter and returns this as a resource. In #2, we use this to print out the description of the `/_system/governance` folder. In #3, we use this same operation to get our WSDL resource and print out its content (#4). Besides its content through the `wsdlResource` object, we also have access to any tags and properties that have been added to this resource.

Summary

We showed you how to run an SOA registry for the first time, register a WS-* based service manually in the registry, and access the WSO2 Governance registry.

Here are some other Manning titles you might be interested in:



[Open Source SOA](#)
Jeff Davis



[SOA Patterns](#)
Arnon Rotem-Gal-Oz, Eric Bruno, and Udi Dahan



[SOA Security](#)
Ramarao Kanneganti and Prasad A. Chodavarapu

Last updated: January 29, 2012