Practical Exercise: Getting Started with WSO2 Enterprise Integrator

Training Objective

Introduce the tabs and menus of the ESB Profile's Management Console, learn how to manage users and roles, and list and test the deployed services of WSO2 EI out-of-the-box through integrated tools.

Business Objectives

Get an understanding of the user management features in WSO2 EI.

High Level Steps

- Install WSO2 EI 6.1.1
- Familiarize yourself with the tabs and options in the WSO2 EI console
- Create a "tester" User
- Create a Role Tester and assign Role
- Test echo service as "tester" User

Detailed Instructions

Install WSO2 EI 6.1.1

Before installing the product, ensure that the installation prerequisites have been fulfilled. Refer the documentation [1] for details. If prerequisites are fulfilled, instructions on installing the product can be found for: Linux or OS X at [2] and Windows [3].

[1] https://docs.wso2.com/display/EI611/Installation+Prerequisites
[2] https://docs.wso2.com/display/EI611/Installing+on+Linux
[3] https://docs.wso2.com/display/EI611/Installing+on+Windows

Start the ESB profile

1. Open a command line interface.
2. Start the ESB profile by navigating to the <EI_HOME>/bin directory and running integrator.bat (on Windows) or integrator.sh (on Linux)
3. Log in to the Management Console of the ESB profile, which is available by default at https://localhost:9443/carbon. You can log in to the console using the default admin/admin credentials.

Familiarize yourself with the Management Console of the ESB profile

The Management Console provides a set of options aggregated into four main tabs:

- Main
- Monitor
- Configure
- Tools

Under the Main tab, we can find a set of options to view the following EI resources that are deployed in the server instance:

- **Services**: WSO2 EI deployed services.
- **Proxy Services**: Virtual services that appear as regular, fledged web services to the external clients. There are proxy services of the following types:
  - **Pass Through Proxy**: A simple proxy service on a specified endpoint that does not perform any processing on the messages.
  - **Secure Proxy**: A proxy that will process WS-Security on incoming requests and forward them to an unsecured backend service.
  - **WSDL Based Proxy**: A proxy service out of a WSDL of an existing Web service.
  - **Logging Proxy**: A proxy service that logs all the incoming requests and forwards them to a given endpoint.
  - **Transformer Proxy**: A proxy service which transforms all the incoming requests using XSLT and then forwards them to a given endpoint.
  - **Custom Proxy**: Launches the proxy service creation wizard.
- **Data Service:** Exposes a backend database as a data service.
- **Sequences**: An array of mediators assembled as a message flow.
- **Inbound Endpoints**: Message entry points that inject messages directly into the mediation layer.
- **Scheduled Tasks**: Jobs deployed in the WSO2 EI runtime for periodic execution.
- **Templates**: Prototypes of Endpoints or Sequences that can be used as a base for new objects.
- **Endpoints**: Definitions of external services endpoints and any attributes or semantics that should be followed when communicating with them.
- **Message Processors**: Units of execution that have the ability to connect to a message store and perform message mediation or required data manipulations.
- **Local Entries**: Local configuration registry/repository for resources such as WSDLs, schemas, scripts, etc.
- **Message Stores**: Units of storage (queues) for messages/data exchanged during WSO2 EI runtime. Queues can refer to the local WSO2 EI in memory store implementation or an external MQ product.

- **Priority Executors**: Resources used in high load scenarios where user wants to execute different sequences at different priority levels.
- **APIs**: Virtual Web applications that provide a convenient approach for filtering and processing HTTP traffic through the service bus.
- **Source View**: WSO2 EI's full XML configuration (except resources that are stored in registry).
- **Connectors**: Zip modules with components to connect to external applications.
- **Secure Vault Tool**: Tool for encrypting and storing passwords.
- **Carbon Applications**: Applications developed for WSO2's product platform (*.car files).
- **Modules**: Modules developed over WSO2's product platform (*.mar files). These extensions can contain processors, executors, etc.
- **Topics**: Group of messages on queues.
- **ESB Artifacts**: Custom extensions that can be deployed in WSO2 EI. These extensions can contain custom mediators, tasks, configuration factories and serializers.
- **Registry**: Browse the registry configuration resources.

Under the Monitor tab, we can find a set of options to check the following WSO2 EI resources:

- **Application Logs**: Logs of WSO2 EI's deployed services.
- **System Statistics**: System's statistical data such as average response time, total requests served, memory allocated.
- **System Logs**: Logs of WSO2 EI operations.
- **Message Flows**: Graphical view of internal message in & out flowing.

Under the Configure tab, we can find a set of options to manage the following WSO2 EI resources:

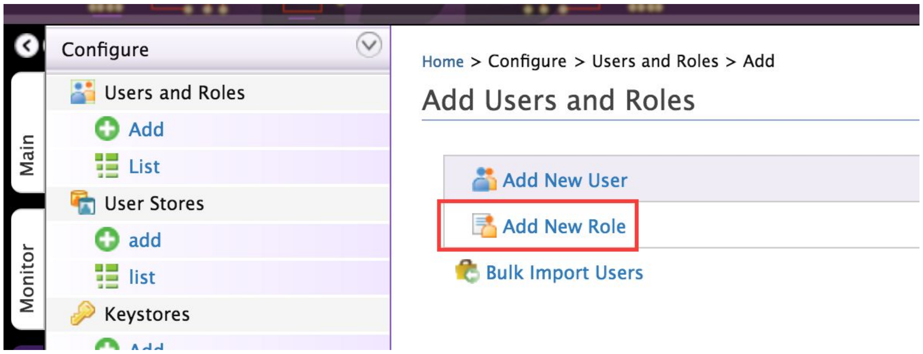
- **Users and Roles**: Users and permissions for actions.
- **User Store Management**: Secondary user stores.
- **KeyStores**: Stores for secret, private and trusted certificate keys.
- **Cloud Gateway Agent**: Selectively publish services and data to the cloud through the WSO2 Cloud Gateway Agent.
- **Features**: Installed WSO2 features.
- **Logging**: WSO2 EI's log4j configuration.
- **Data Sources**: Definition of connections to databases that can be used by WSO2 EI resources.
- **BAM Server Profile**: Resources where the user can define a set of event streams that can be used by the WSO2 DAS mediator declaration time.
- **Server Roles**: Security server roles.
- **Event Sinks:** Information about transport endpoints provided by other systems. Events can be published to these endpoints.
- **Multi Tenancy**: Multiple WSO2 EI resource instances on a single WSO2 EI installation.

Under the Tools tab, we can find a set of options to manage the following WSO2 EI resources:

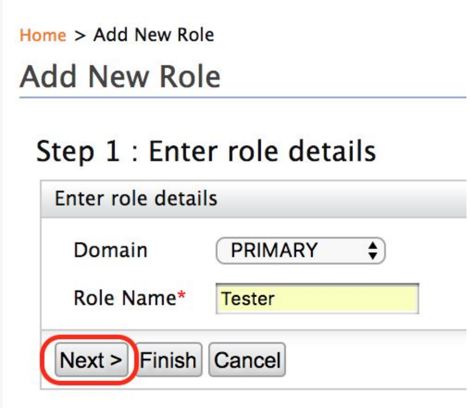
- **WSDL2Java**: Generates Java code to provide/consume a web service for a given WSDL.
- **Java2WSDL**: Bottom-up Java Web Service builder.
- **Try It**: Internet tool for service testing.
- **WSDL Validator**: Tool for validating WSDL files.

Define Roles

1. Navigate **Configure** → **Users and Roles** → **Add** and click **Add New Role**.



2. Add the role by providing **Tester** as the **Role Name** and click **Next**.


Home > Add New Role

Add New Role

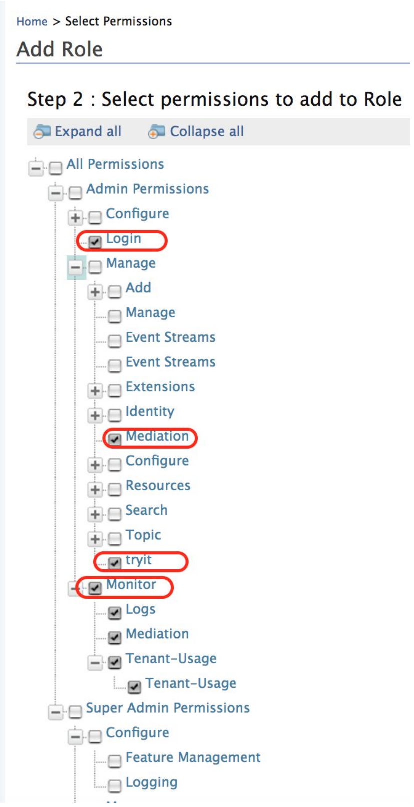
Step 1 : Enter role details

Enter role details

Domain: PRIMARY

Role Name*: Tester

3. Select **Login, Mediation, Tryit** and **Monitor** permissions and click **Finish**.

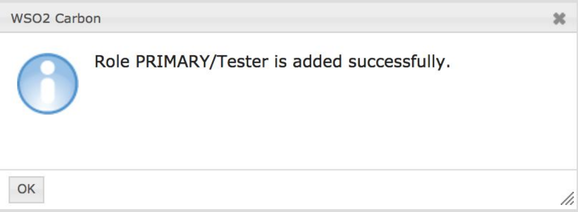


Home > Select Permissions

Add Role

Step 2 : Select permissions to add to Role

- □ All Permissions
  - □ Admin Permissions
    - □ Configure
    - ☑ Login
    - □ Manage
      - □ Add
      - □ Manage
      - □ Event Streams
      - □ Event Streams
      - □ Extensions
      - □ Identity
      - ☑ Mediation
      - □ Configure
      - □ Resources
      - □ Search
      - □ Topic
      - ☑ tryit
    - ☑ Monitor
      - ☑ Logs
      - ☑ Mediation
      - ☑ Tenant-Usage
        - ☑ Tenant-Usage
  - □ Super Admin Permissions
    - □ Configure
      - □ Feature Management
      - □ Logging

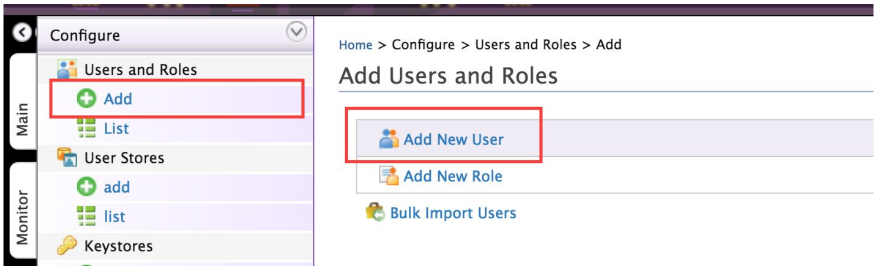


WSO2 Carbon

Role PRIMARY/Tester is added successfully.

Create a user with the ‘Tester’ role

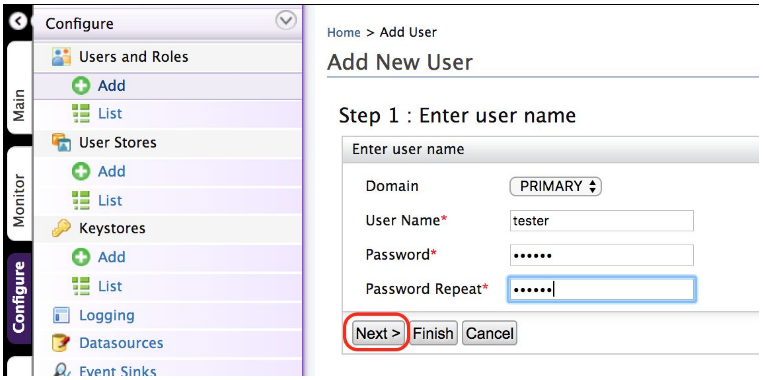
1. Navigate **Configure → Users and Roles → Add** and click **Add New User**.



Home > Configure > Users and Roles > Add

Add Users and Roles

2. Create the user with tester/tester as the name and password and click **Next** to complete user creation.



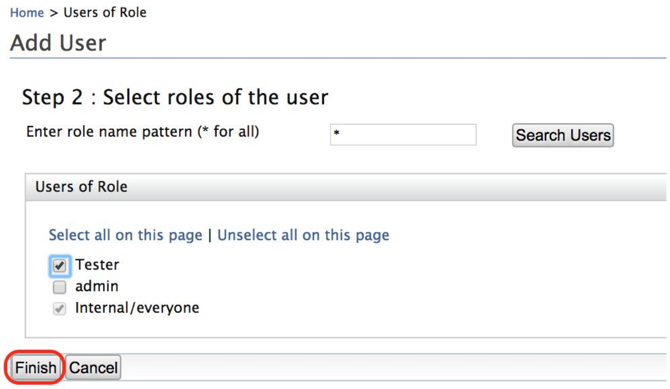
Add New User

Step 1 : Enter user name

Enter user name

Domain	PRIMARY
User Name*	tester
Password*	••••••
Password Repeat*	••••••

3. Select **Tester** from the list of roles and click **Finish**.



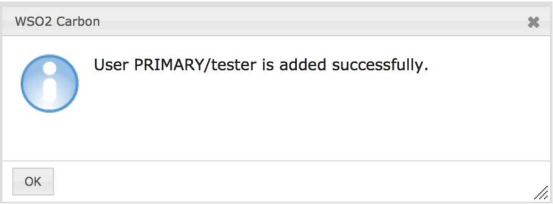
Add User

Step 2 : Select roles of the user

Enter role name pattern (* for all) *

Users of Role

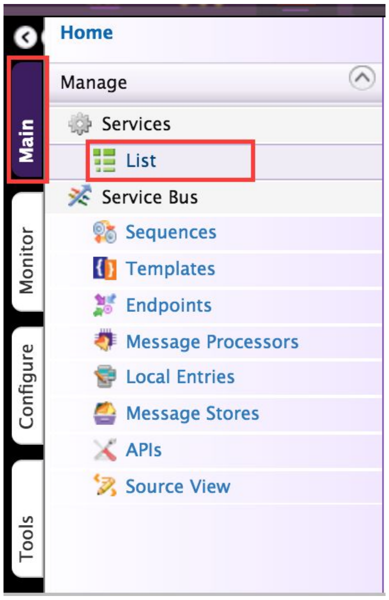
☑ Tester
☐ admin
☑ Internal/everyone



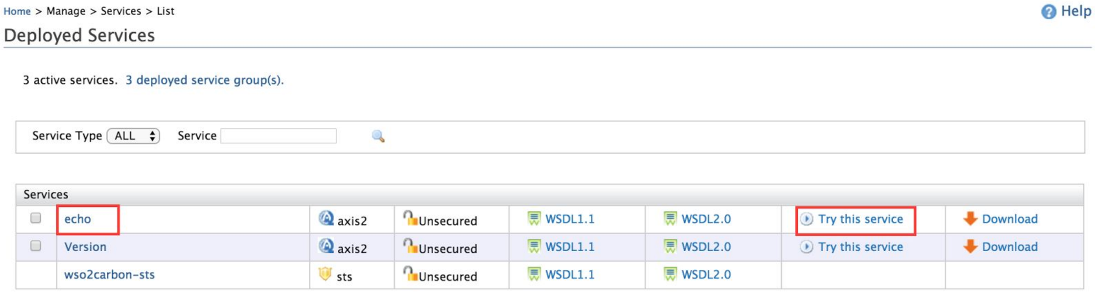
User PRIMARY/tester is added successfully.

Test the ‘echo’ service as the ‘tester’ user

1. Log in to the Admin console as tester/tester.
2. Select **Main** → **Services** → **List**.



3. Select the **echo** service from the list and click **Try this service** for the echo service.



Home > Manage > Services > List

Help

Deployed Services

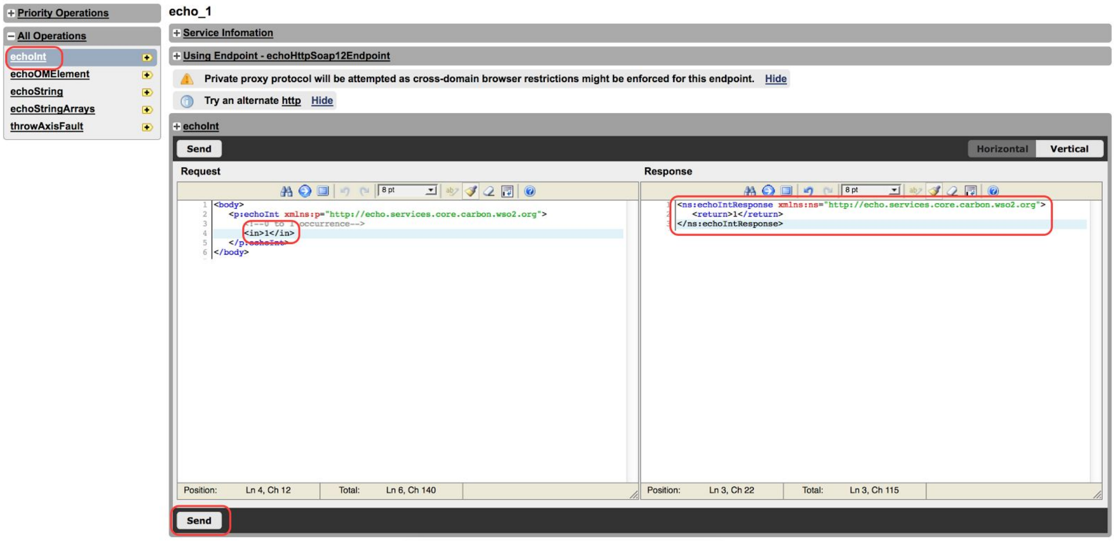
3 active services. 3 deployed service group(s).

Service Type ALL Service

Services

□	echo	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
□	Version	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
	wso2carbon-sts	sts	Unsecured	WSDL1.1	WSDL2.0		

4. Select echo Init operation, introduce a number between <in> and </in> tags of the XML payload body.
5. Click the **Send** button.
6. The service executes and returns the original number sent.



echo_1

Private proxy protocol will be attempted as cross-domain browser restrictions might be enforced for this endpoint. Hide

Try an alternate http Hide

Request

```
<body>
   <p:echoInt xmlns:p="http://echo.services.core.carbon.wso2.org">
      <!--0 to 1 occurrence-->
      <in>1</in>
   </p:echoInt>
</body>
```

Position: Ln 4, Ch 12 Total: Ln 6, Ch 140

Response

```
<ns:echoIntResponse xmlns:ns="http://echo.services.core.carbon.wso2.org">
   <return>1</return>
</ns:echoIntResponse>
```

Position: Ln 3, Ch 22 Total: Ln 3, Ch 115