talend Connecting the Data-Driven Enterprise >

Lab Guide

ESB Administration

Version 6.2

Talend Inc.
800 Bridge Parkway, Suite 200
Redwood City, CA 94065
United States
+1 (650) 539 3200

Welcome to Talend Training

Congratulations on choosing a Talend training course.

Working through the course

You will develop your skills by working through use cases and practice exercises using live software. Completing the exercises is critical to learning!

If you are following a self-paced, on-demand training (ODT) module, and you need an answer to proceed with a particular exercise, use the help suggestions on your image desktop. If you can't access your image, contact customercare@talend.com.

Exploring

You will be working in actual Talend software, not a simulation. We hope you have fun and get lots of practice using the software! However, if you work on tasks beyond the scope of the training, you could run out of time with the environment, or you could mess up data or Jobs needed for subsequent exercises. We suggest finishing the course first, and if you have remaining time, explore as you wish. Keep in mind that our technical support team can't assist with your exploring beyond the course materials.

For more information

Talend product documentation (help.talend.com)

Talend Community (community.talend.com)

Sharing

This course is provided for your personal use under an agreement with Talend. You may not take screenshots or redistribute the content or software.

This page intentionally left blank to ensure new chapters start on right (odd number) pages.

CONTENTS

LESSON 1

ESB Infrastrucure

This chapter discusses the following.

Lab Overview

ESB infrastructure

Talend ESB consists of several different software applications working together to provide a distributed development environment. These include Talend Administration Center (TAC), Talend Studio, CommandLine, Nexus artifact repository, Talend Runtime, and Service Activity Monitoring (SAM). Talend Administration Center connects to Nexus artifact repository to retrieve and expose the deployment artifacts for the Talend Runtime container. TAC simplifies the deployment and management of routes and services within Talend Runtime. This lesson introduces you to the Talend ESB server and its infrastructure. In addition, this course introduces new features: ESB Conductor and ESB Runtime, which can be accessed from TAC.

As for data integration features, the following high-level development life cycle applies to ESB features: **Develop > Store > Publish > Deploy**:

Note: This is a typical progression, not the only one. In this course, you will learn more about Nexus and Talend Runtime, as well as the deployment process and related Talend applications (such as ESB Conductor and Publisher).

Objectives

After completing this lesson, you will know more about the Talend ESB infrastructure and Talend Runtime.

Next Steps

First we will make sure the training environment is configured and ready for ESB features.

Preparing Talend Administration Center and the VM Environment

Overview

You access Talend Administration Center (TAC) using your web browser. From TAC you manage users, projects, and Job deployment and scheduling, among other functions. TAC is available only for Talend subscription products. You can configure how TAC operates directly from its Web interface.

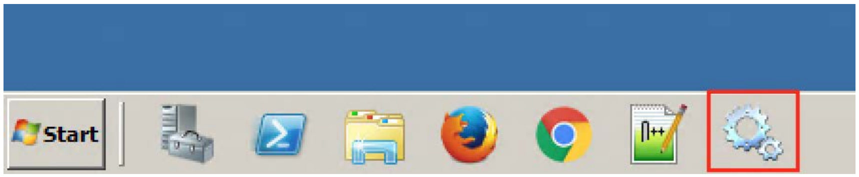
If you ran the Talend DI Administration training on the same VM as this training, you already configured all mandatory TAC parameters and can move on to the next section of this course.

If this is your first time connecting to your VM, please run the following routine. Bear in mind that a good understanding of TAC and Talend administrative tasks is required for this ESB Administration course.

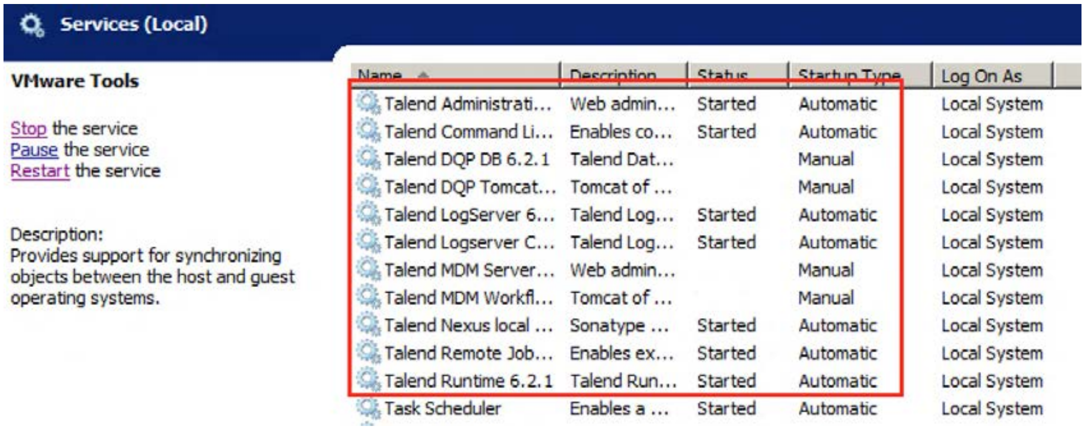
These instructions take you through identifying Talend services and checking their statuses. If they are not running (e.g. Stopped), you need to start them. If they are running (Started), you can continue without taking any action. Here is a high-level view of common administrative tasks you can do from TAC:

Start services

1. In the Windows taskbar, right-click the **Services** icon.



2. In the **Services** window, scroll down to view Talend installed services:



Services (Local)

VMware Tools

Stop the service
Pause the service
Restart the service

Description:
Provides support for synchronizing objects between the host and guest operating systems.

Name	Description	Status	Startup Type	Log On As
Talend Administrati...	Web admin...	Started	Automatic	Local System
Talend Command Li...	Enables co...	Started	Automatic	Local System
Talend DQP DB 6.2.1	Talend Dat...		Manual	Local System
Talend DQP Tomcat...	Tomcat of ...		Manual	Local System
Talend LogServer 6...	Talend Log...	Started	Automatic	Local System
Talend Logserver C...	Talend Log...	Started	Automatic	Local System
Talend MDM Server...	Web admin...		Manual	Local System
Talend MDM Workfl...	Tomcat of ...		Manual	Local System
Talend Nexus local ...	Sonatype ...	Started	Automatic	Local System
Talend Remote Job...	Enables ex...	Started	Automatic	Local System
Talend Runtime 6.2.1	Talend Run...	Started	Automatic	Local System
Task Scheduler	Enables a ...	Started	Automatic	Local System

Make sure the following Talend services are started (start them if they are not)

Talend Administration Center

Talend Command Line

Talend LogServer

Talend LogServer Collector

Talend Nexus local

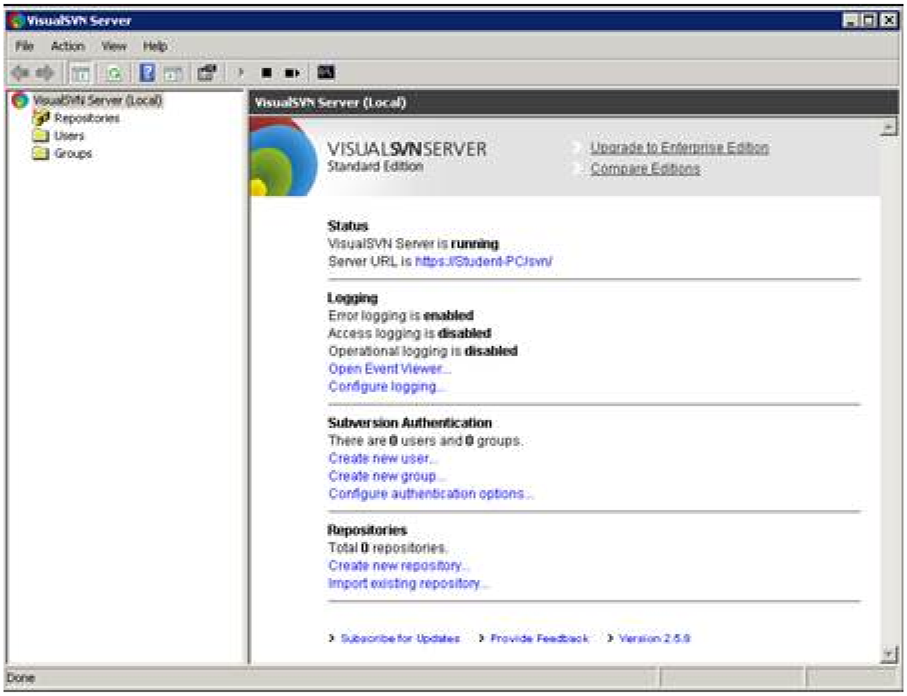
Talend Remote Jobserver

Talend Runtime

Configure SVN

Create an SVN repository for projects

1. From the Windows **Start** menu, start **VisualSVN Server Manager**:



Status
VisualSVN Server is **running**
Server URL is https://Student-PC/svn/

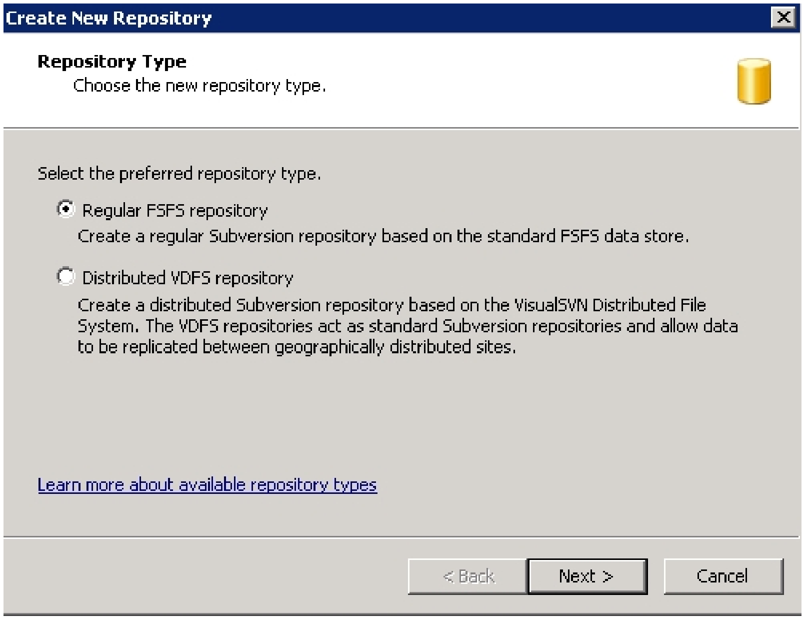
Logging
Error logging is **enabled**
Access logging is **disabled**
Operational logging is **disabled**
Open Event Viewer...
Configure logging...

Subversion Authentication
There are **0** users and **0** groups.
Create new user...
Create new group...
Configure authentication options...

Repositories
Total **0** repositories.
Create new repository...
Import existing repository...

2. Right-click **Repositories** and click **Create New Repository**.

3. Select **Regular FSFS repository** and click **Next**.



Create New Repository

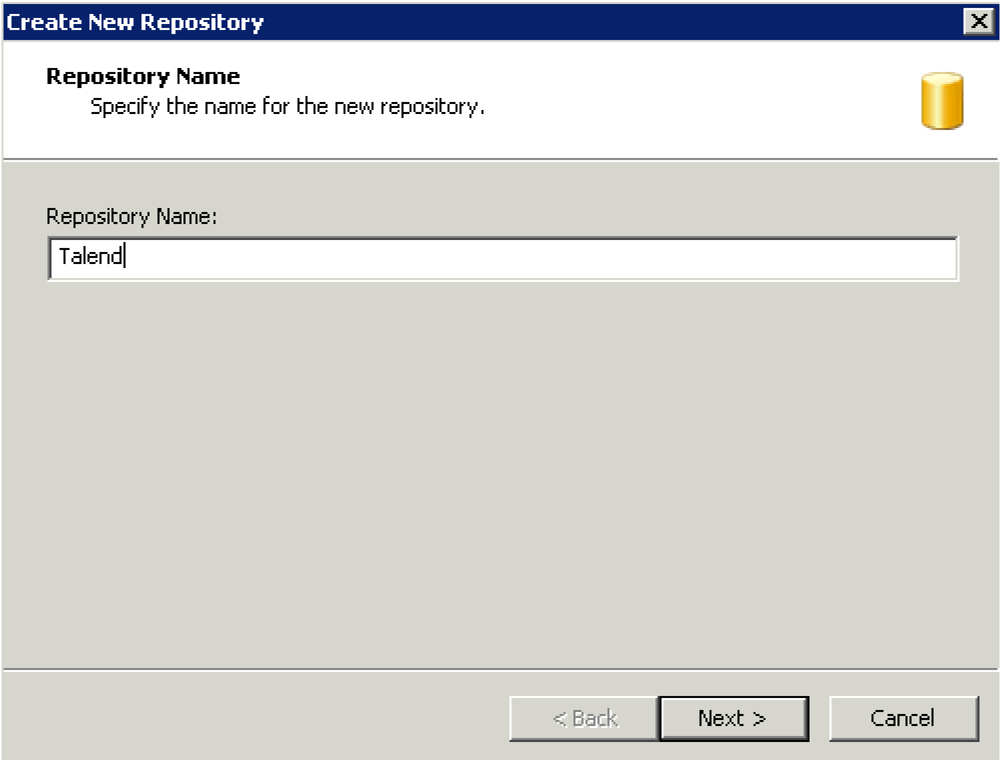
Repository Type
Choose the new repository type.

Select the preferred repository type.

● Regular FSFS repository
Create a regular Subversion repository based on the standard FSFS data store.

○ Distributed VDFS repository
Create a distributed Subversion repository based on the VisualSVN Distributed File System. The VDFS repositories act as standard Subversion repositories and allow data to be replicated between geographically distributed sites.

4. In the **Repository Name** field, enter *Talend* and click **Next**.

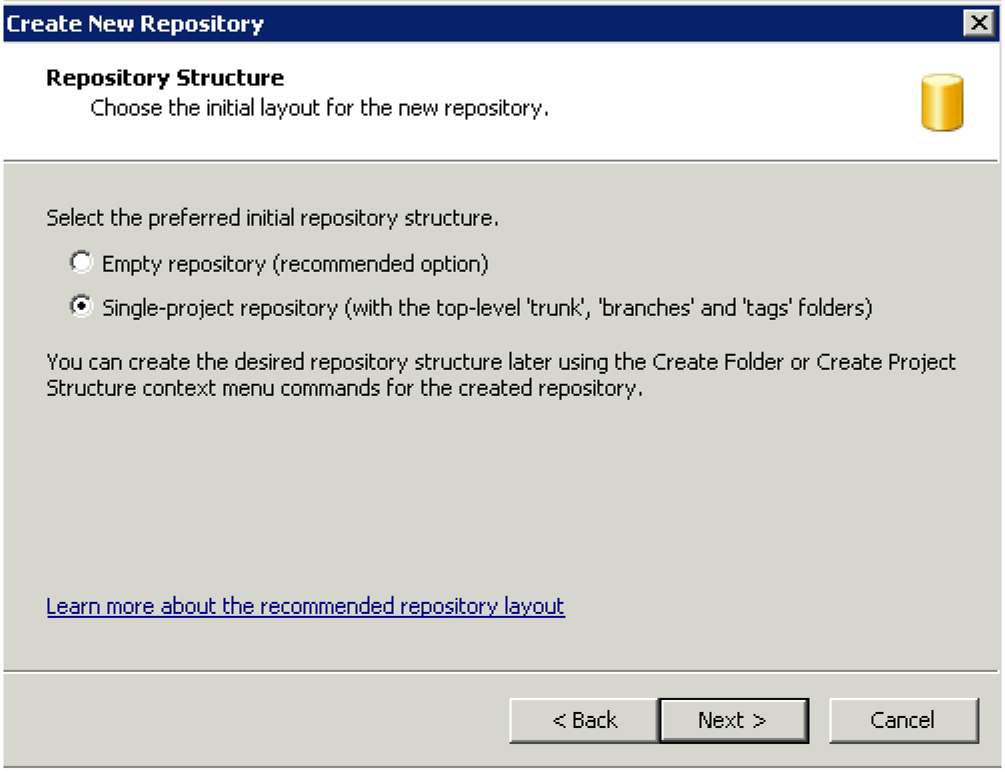


Create New Repository

Repository Name
Specify the name for the new repository.

Repository Name: Talend

5. Select **Single-project repository (with the top-level 'trunk', 'branches' and 'tags' folders)**.


Create New Repository
Repository Structure
Choose the initial layout for the new repository.

Select the preferred initial repository structure.

○ Empty repository (recommended option)

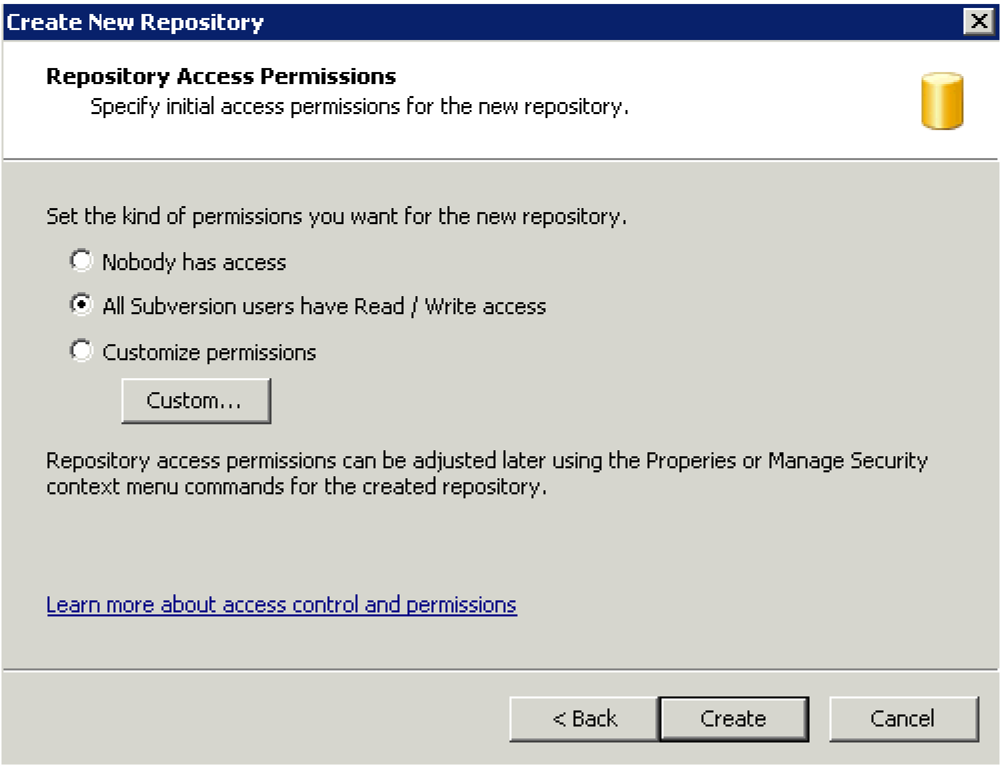
● Single-project repository (with the top-level 'trunk', 'branches' and 'tags' folders)

You can create the desired repository structure later using the Create Folder or Create Project Structure context menu commands for the created repository.

Learn more about the recommended repository layout
< Back
Next >
Cancel

Click **Next**.

6. Select **All Subversion users have Read / Write access**.
   You will be able to adjust permissions for different users later in the process.



Create New Repository

Repository Access Permissions

Specify initial access permissions for the new repository.

Set the kind of permissions you want for the new repository.

○ Nobody has access

● All Subversion users have Read / Write access

○ Customize permissions

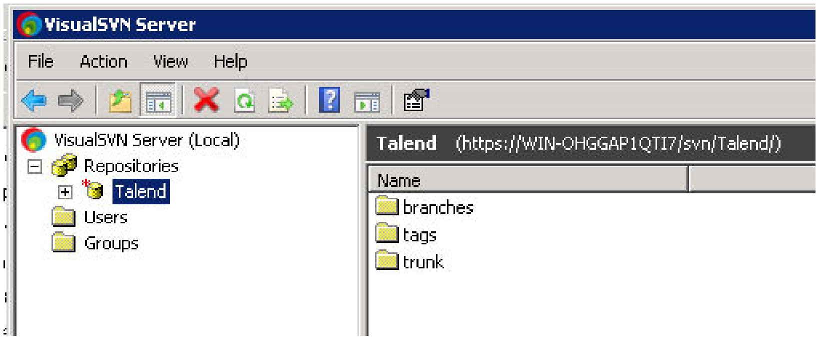
Repository access permissions can be adjusted later using the Properies or Manage Security context menu commands for the created repository.

Learn more about access control and permissions

7. Click **Create**.

8. Click **Finish**.

 The new repository appears:



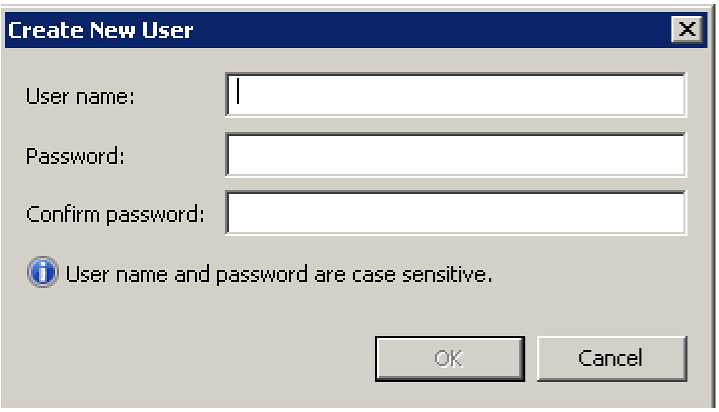
Talend (https://WIN-OHGGAP1QTI7/svn/Talend/)

Name
branches
tags
trunk

You can use this repository with the TAC to create shared projects for multiple developers.

Configure the SVN repository

1. Still in VisualSVN Server Manager, right-click **Users** and click **Create New User**:



Create New User

User name:

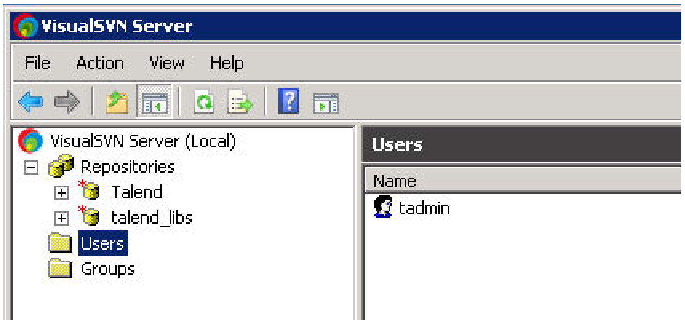
Password:

Confirm password:

User name and password are case sensitive.

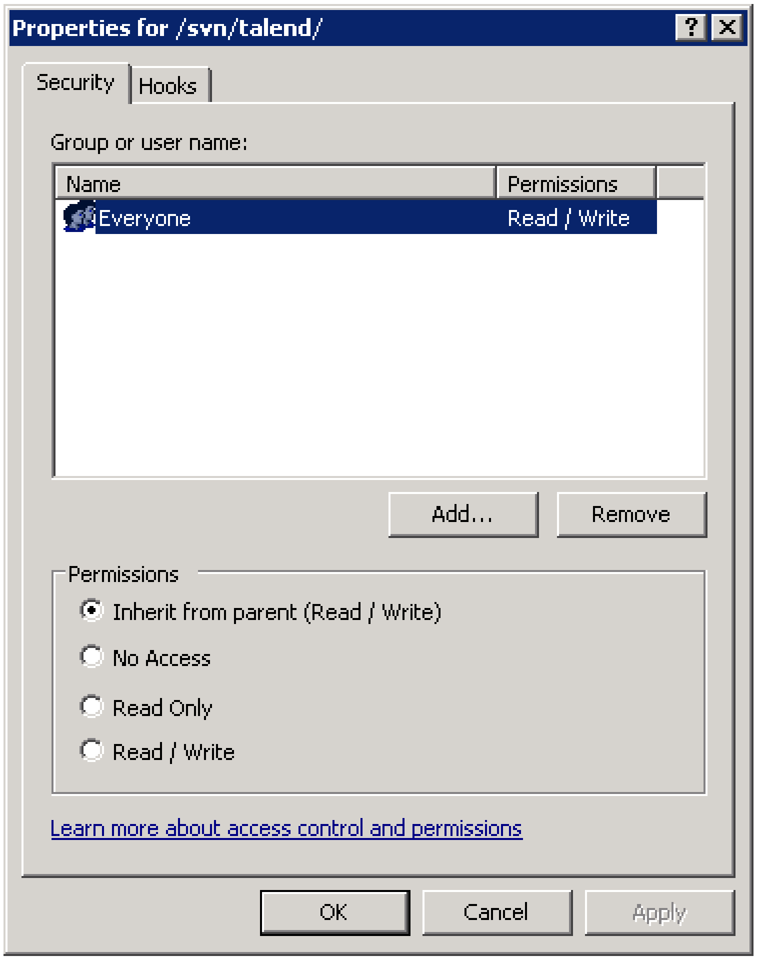
You are creating a user account to use to restrict access to this repository.

2. In all three boxes, enter *tadmin* and click **OK**:



Again, the name and password are not significant, but you must remember them.

3. Right-click the **Talend** repository and then click **Properties**:


Properties for /svn/talend/
?
Security
Hooks
Add...
Remove
Learn more about access control and permissions
OK
Cancel
Apply

Group or user name:

Name	Permissions
Everyone	Read / Write

Permissions

● Inherit from parent (Read / Write)

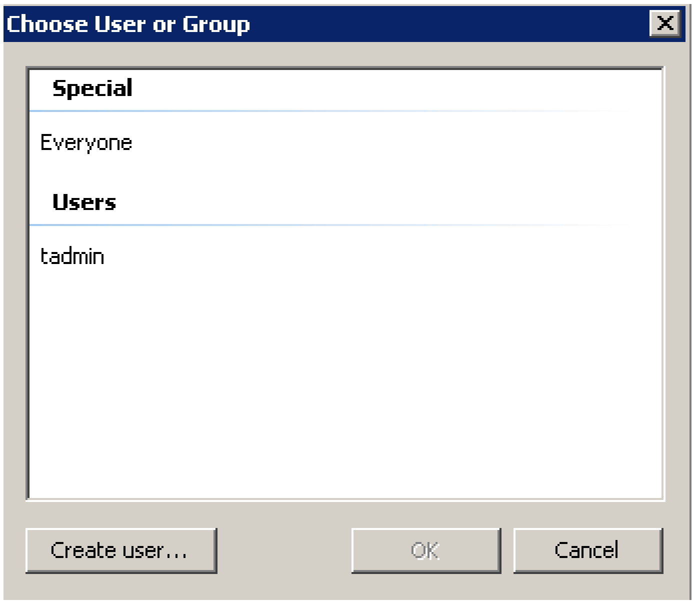
○ No Access

○ Read Only

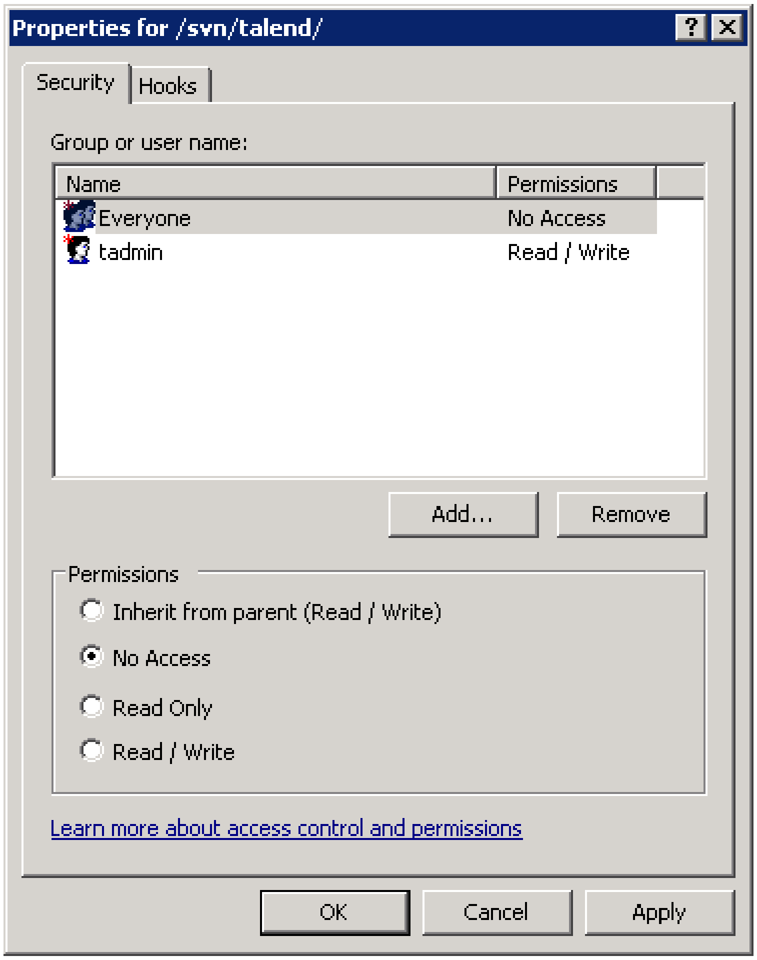
○ Read / Write

By default, everyone has full permissions for this repository.

4. Click **Add**.



5. To specify that you want to grant permissions to the tadmin user, click **tadmin** and **OK**.
6. Select **Everyone** and the **No Access** radio button. Select **tadmin** to ensure that the **Read / Write** radio button is selected:



Properties for /svn/talend/

Group or user name:

Name	Permissions
Everyone	No Access
tadmin	Read / Write

Permissions

○ Inherit from parent (Read / Write)

● No Access

○ Read Only

○ Read / Write

Now only the tadmin user has read/write access to the Talend repository.

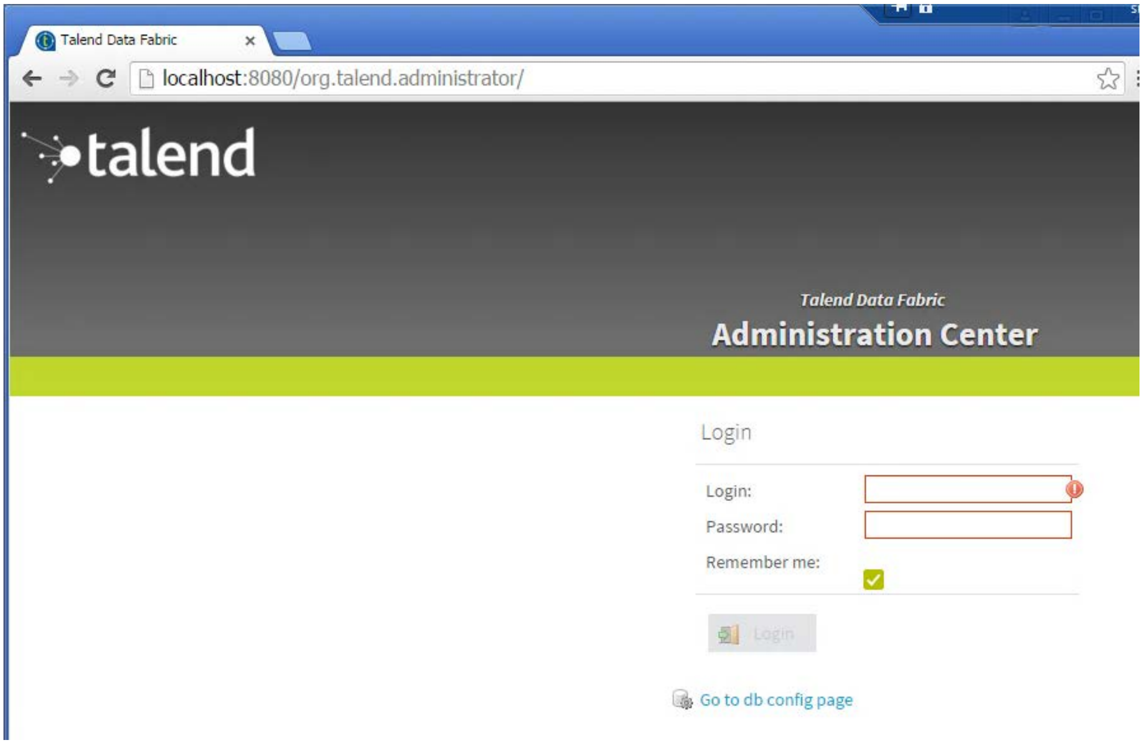
7. Click **OK**.

The SVN repository configuration is complete. Leave VisualSVN Server Manager running because you will need to return to it one more time.

Configure TAC

Create users

1. In a web brower, go to **http://localhost:8080/org.talend.administrator**:

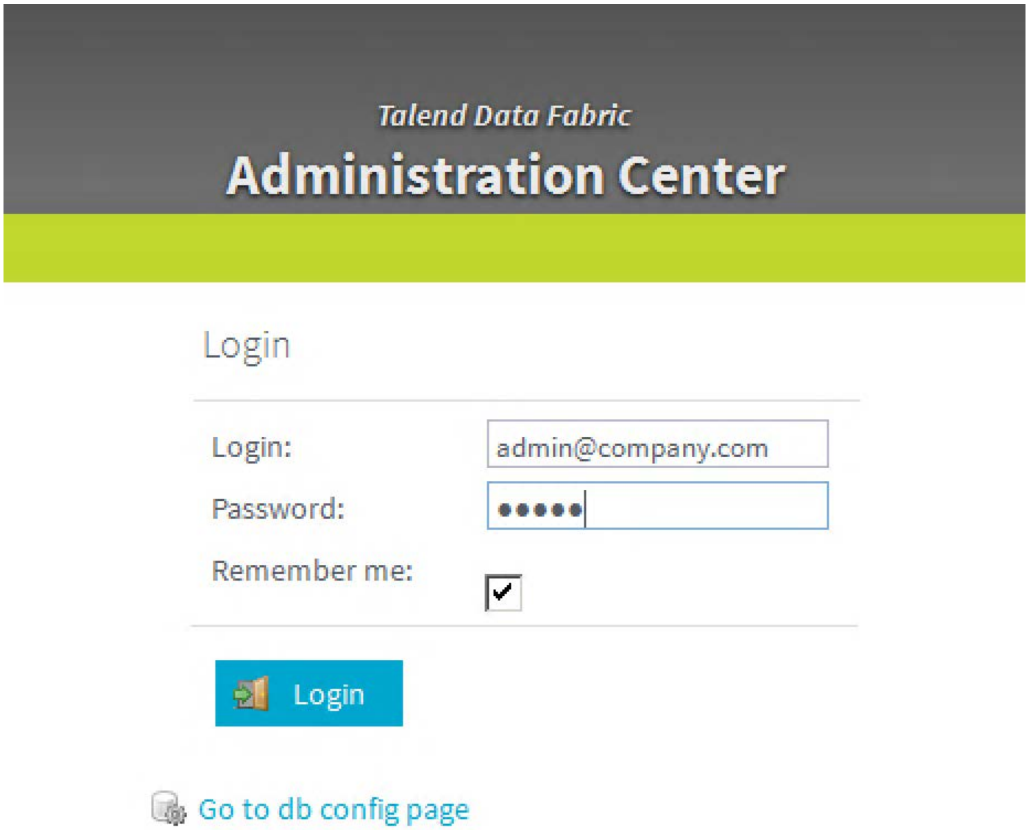


2. Use the following credentials to log in:

   Login: *admin@company.com*

   Password: *admin*

Tip: Select the **Remember me** check box to avoid having to remember these credentials.

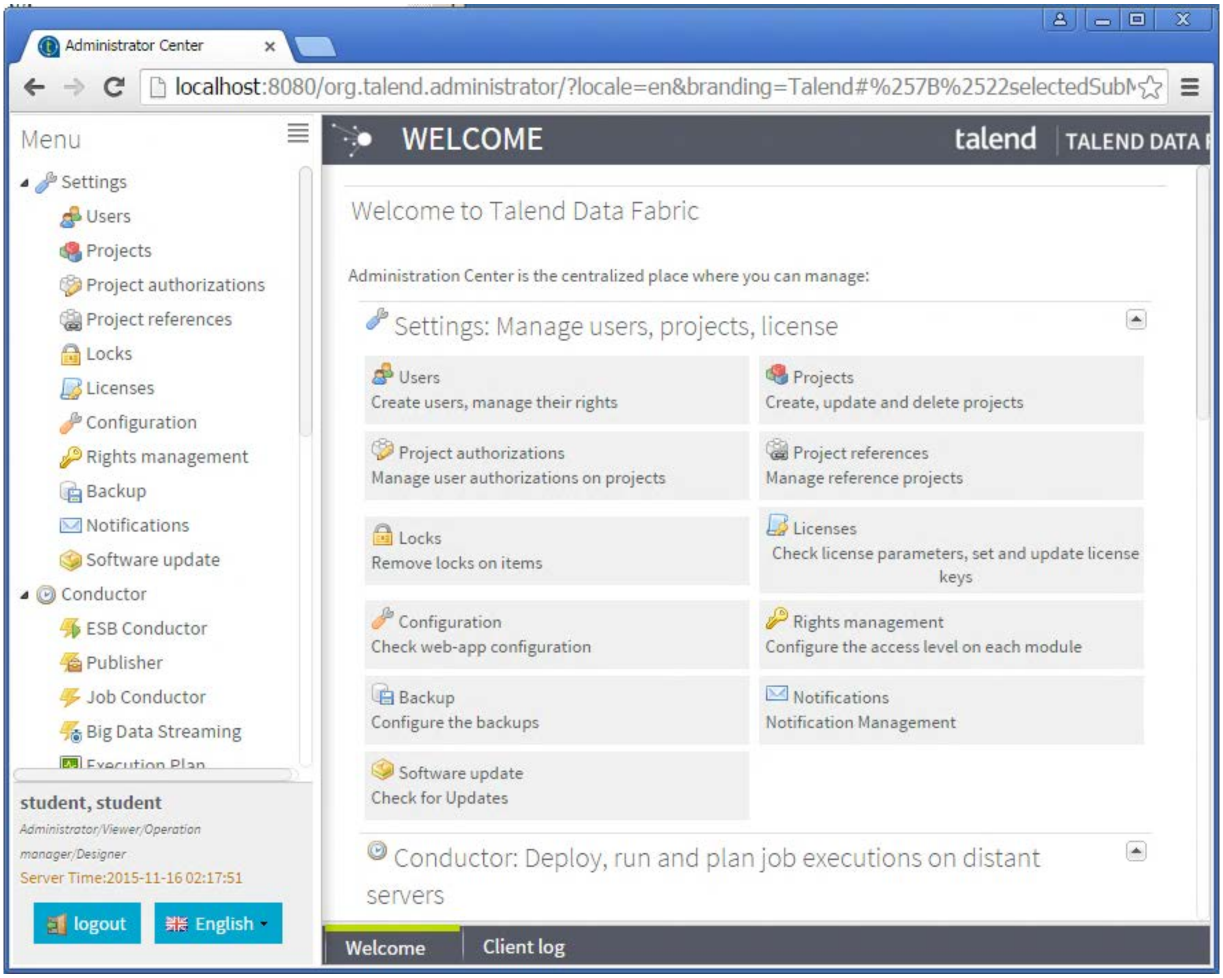


Login: admin@company.com

Password: •••••

Remember me: ☑

3. Click **Login**:



Welcome to Talend Data Fabric

Administration Center is the centralized place where you can manage:

Settings: Manage users, projects, license

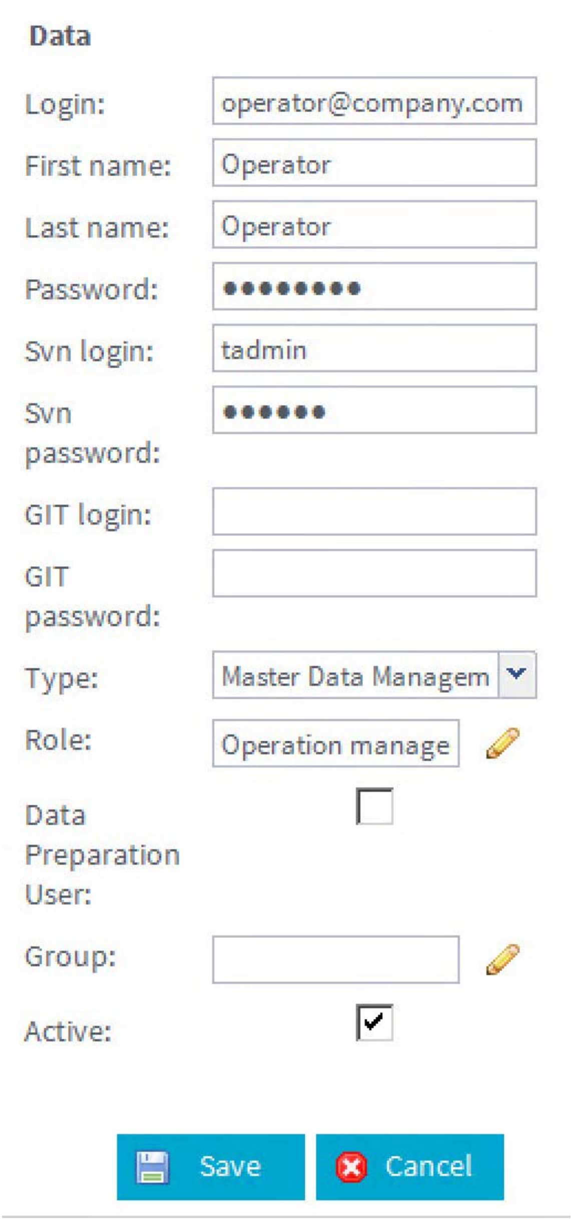
Users Create users, manage their rights	Projects Create, update and delete projects
Project authorizations Manage user authorizations on projects	Project references Manage reference projects
Locks Remove locks on items	Licenses Check license parameters, set and update license keys
Configuration Check web-app configuration	Rights management Configure the access level on each module
Backup Configure the backups	Notifications Notification Management
Software update Check for Updates	

Conductor: Deploy, run and plan job executions on distant servers

The TAC Welcome page displays menus and items in accordance with your TAC edition and user role: administrator, operations manager, designer, or viewer.

Note: Specific messaging on the Welcome screen may vary, depending on the final license used for the training environment. The license also determines which features are available, so your company test and/or production environments will likely vary.

1. In the menu on the left, click **Users**. At the top of the center section, click the **Add** button.
2. Create a new user with the following parameters:
   Login: *operator@company.com*
   First Name: *Operator*
   Last Name: *Operator*
   Password: *operator*
   SVN login: *tadmin*
   SVN password: *tadmin*
   Type: *Master Data Management*



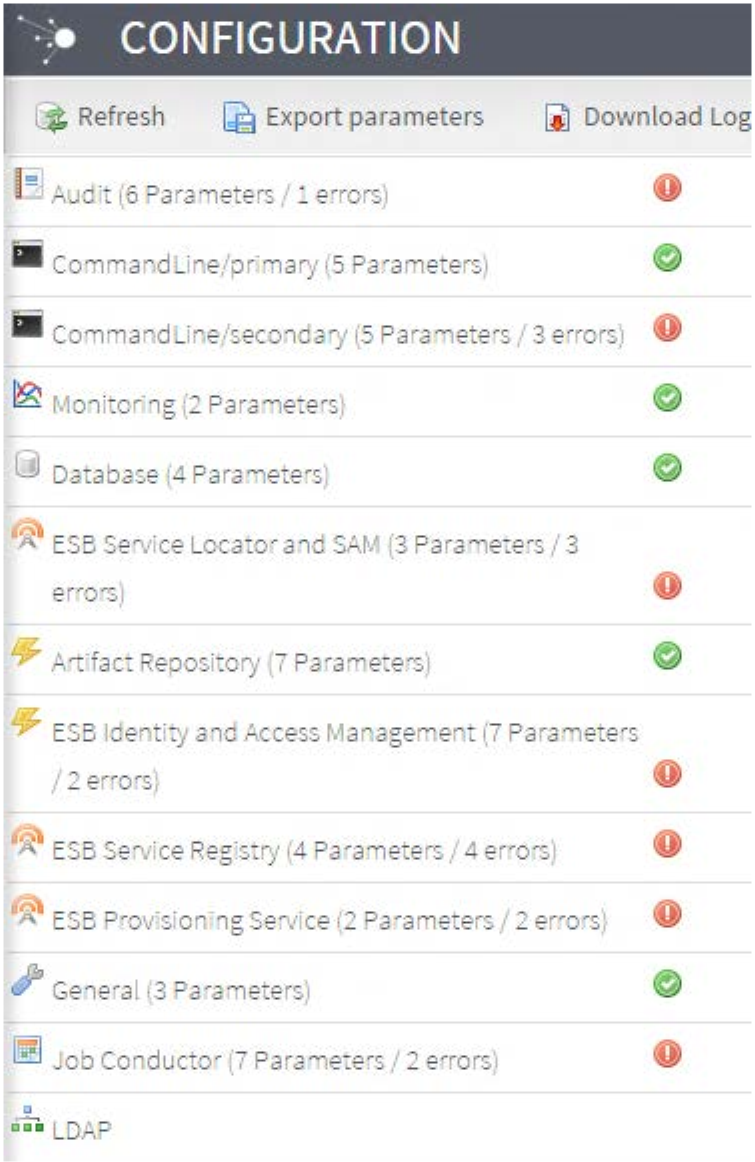
Data

Login:	operator@company.com	
First name:	Operator	
Last name:	Operator	
Password:	••••••••	
Svn login:	tadmin	
Svn password:	••••••	
GIT login:		
GIT password:		
Type:	Master Data Managem	
Role:	Operation manage	
Data Preparation User:	□	
Group:		
Active:	☑	

3. Next to **Role**, click the button marked with a pencil.
4. Select **Operation manager** and click **Validate**.
5. At the bottom of the list, click **Save**. The new user appears on the list.

Configure SVN access in TAC

1. On the TAC menu, navigate to **Settings** > **Configuration**:



CONFIGURATION

Refresh | Export parameters | Download Log

- Audit (6 Parameters / 1 errors) ❗
- CommandLine/primary (5 Parameters) ✓
- CommandLine/secondary (5 Parameters / 3 errors) ❗
- Monitoring (2 Parameters) ✓
- Database (4 Parameters) ✓
- ESB Service Locator and SAM (3 Parameters / 3 errors) ❗
- Artifact Repository (7 Parameters) ✓
- ESB Identity and Access Management (7 Parameters / 2 errors) ❗
- ESB Service Registry (4 Parameters / 4 errors) ❗
- ESB Provisioning Service (2 Parameters / 2 errors) ❗
- General (3 Parameters) ✓
- Job Conductor (7 Parameters / 2 errors) ❗
- LDAP

2. The Configuration page has details and values for applications that are available from TAC (including database connections). This is a good place to troubleshoot problems and view Talend log files. Three status icons are displayed for each configuration item:

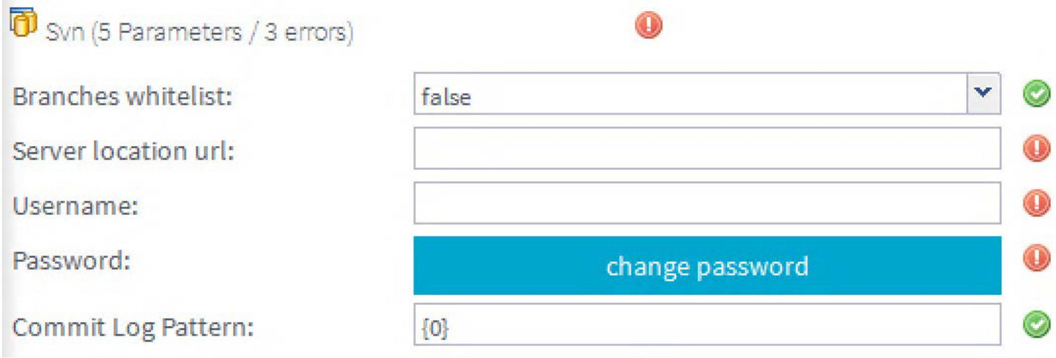
   ✓ Valid. Correctly configured and ready for use.

   ❗ Failed. Not properly configured. Often bad credentials or no system access.

   ⚡ Check. TAC Configuration status is still being checked.

3. Return to the VisualSVN Server Manager application. In **Repositories**, right-click **Talend** and click **Copy URL to Clipboard**.

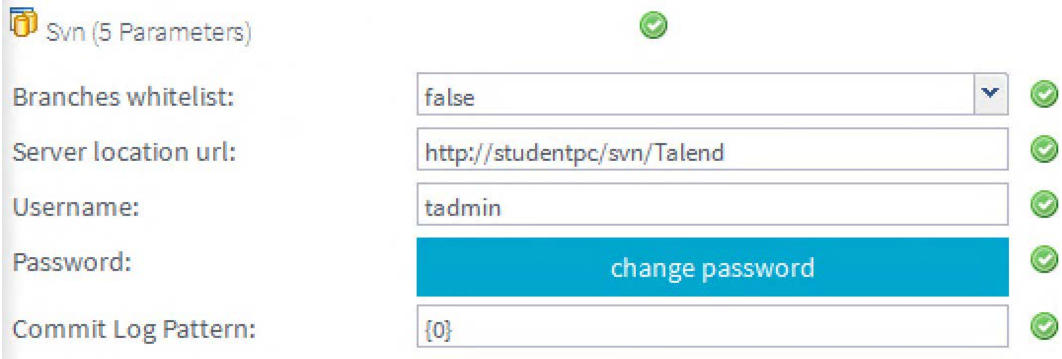
4. On the TAC configuration page, click the **Svn** section to expand it:



Svn (5 Parameters / 3 errors) ❗

Branches whitelist:	false	✓
Server location url:		❗
Username:		❗
Password:	change password	❗
Commit Log Pattern:	{0}	✓

5. Paste the URL in the **Server location url** field and enter *tadmin* in the **Username** and **Password** fields (remember, that is the account name and password you set for the SVN repository).
Note: Your server location URL may differ because you are in a training environment.



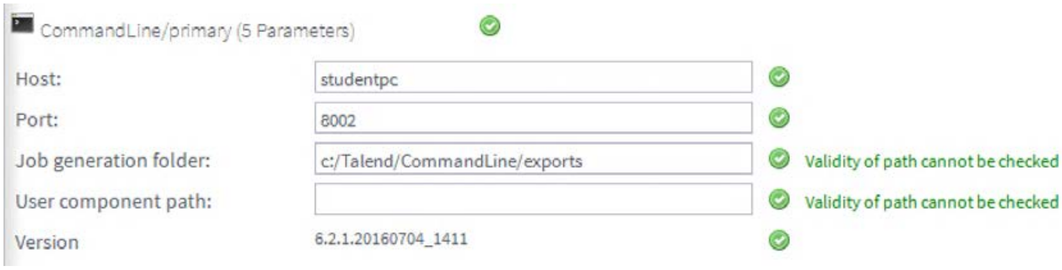
Svn (5 Parameters) ✓

Branches whitelist:	false	✓
Server location url:	http://studentpc/svn/Talend	✓
Username:	tadmin	✓
Password:	change password	✓
Commit Log Pattern:	{0}	✓

Now the SVN settings show as OK with green check marks.

Configure CommandLine in TAC

1. On the TAC **Configuration** page, click **CommandLine/primary** to expand the section.



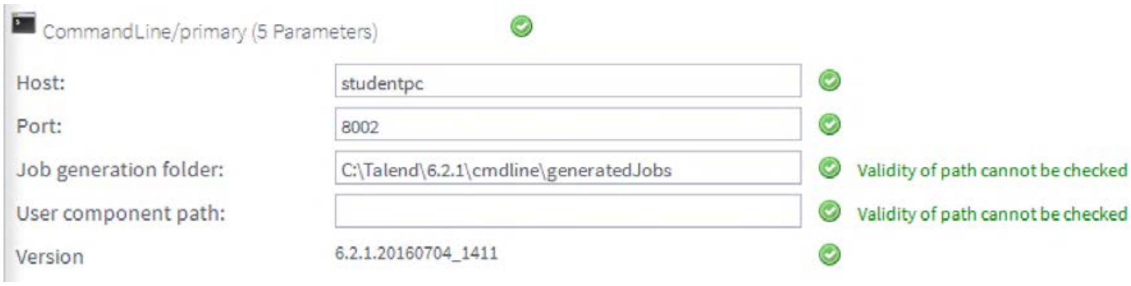
CommandLine/primary (5 Parameters) ✓

Host:	studentpc	✓	
Port:	8002	✓	
Job generation folder:	c:/Talend/CommandLine/exports	✓	Validity of path cannot be checked
User component path:		✓	Validity of path cannot be checked
Version	6.2.1.20160704_1411	✓	

All parameters seem OK, but for two, "**Validity of path cannot be checked**". These parameters must point to existing directories. If directories do not exist, CommandLine creates them when needed (if there are enough privileges). It is always a good practice to change the Job generation folder so generated artifacts are stored in a folder that you will be checking regularly. The User component path should only be filled in if development team members add custom components to their DI Jobs (in which case, they should tell you).

2. To control the folder in which CommandLine stores the generated jobs, open a file explorer and navigate to **C:\Talend\6.2.1\cmdline**.
This folder contains all CommandLine files. In it, create a new folder called *generatedJobs*.

3. Back in TAC, update the Job generation folder parameter and set it to **C:\Talend\6.2.1\cmdline\generatedJobs**. When CommandLine generates an artifact, it will be stored in this new folder.



CommandLine/primary (5 Parameters) ✓

Host: studentpc ✓

Port: 8002 ✓

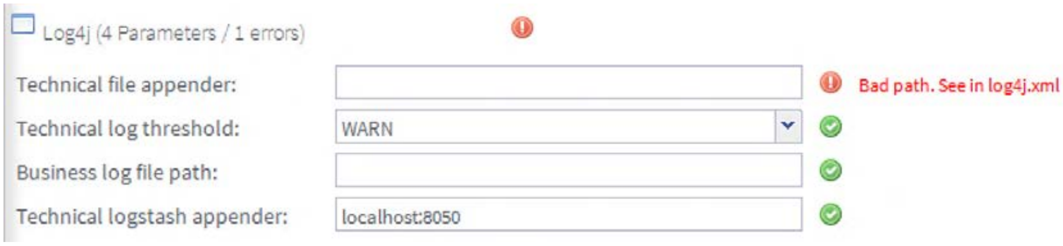
Job generation folder: C:\Talend\6.2.1\cmdline\generatedJobs ✓ Validity of path cannot be checked

User component path: ✓ Validity of path cannot be checked

Version 6.2.1.20160704_1411 ✓

Configure Log4j in TAC

1. On the TAC **Configuration** page, click **Log4j** to expand the section.



Log4j (4 Parameters / 1 errors) ⓘ

Technical file appender: ⓘ Bad path. See in log4j.xml

Technical log threshold: WARN ✓

Business log file path: ✓

Technical logstash appender: localhost:8050 ✓

2. In a file browser, navigate to **C:\Talend\6.2.1** and create a new folder called *logs*.
3. Back on the TAC **Configuration** page, fill in the Log4j parameters as follows:
   Technical file appender: *C:\Talend\6.2.1\logs\technical.log*
   Business log file path: *C:\Talend\6.2.1\logs\business.log*
4. The Logstash appender parameter should point to your Logstash server and the port listening to your logs. We will cover this in the lab on Talend Log Server. Just remember that no URL in your installation should include localhost. You can replace this parameter with **host_name:8050** where **host_name** is the name of the server on which Log Server is installed. For this course, you can leave the value as **localhost:8050**.

Continue to the next section of this course.

Exploring Web Services

Overview

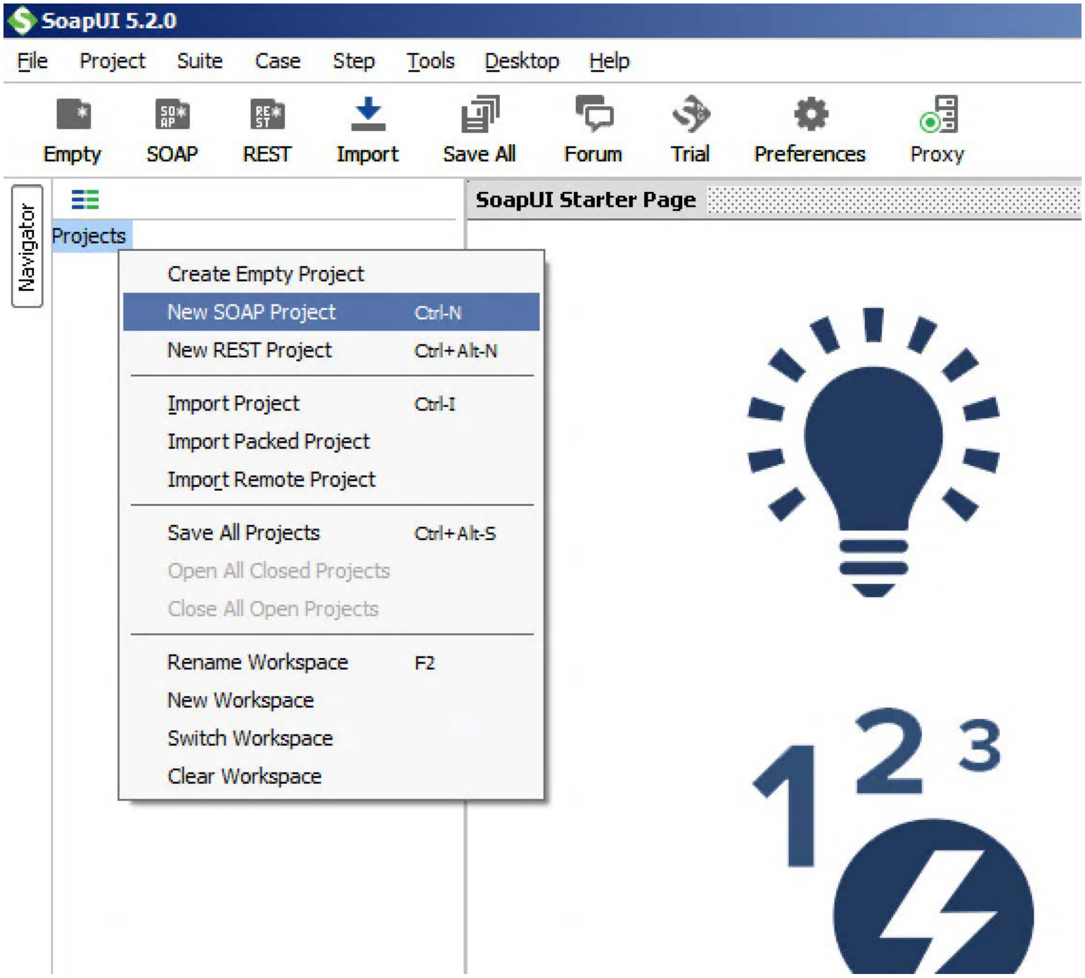
This section shows you how to test and query web services by using a thick-client tool. It also helps you differentiate between SOAP and REST web service standards.

Two web services have already been deployed on the Talend Runtime instance of your training VM. By using the SoapUI client, you will learn what a WSDL (Web Services Definition Language) file is, how to send a SOAP query, and how to access a REST service.

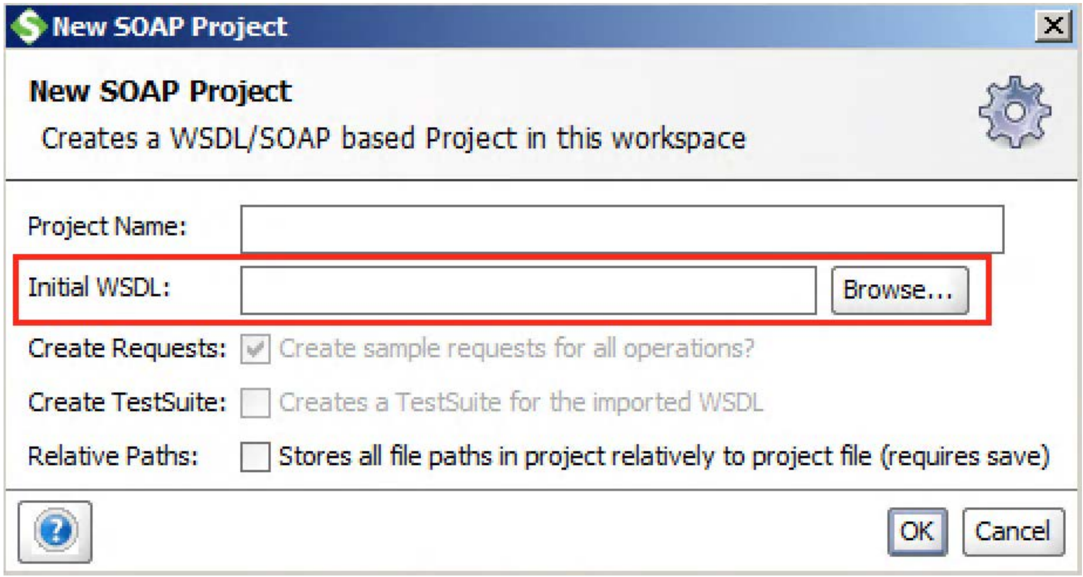
SOAP service

The testSOAPWebService service you are about to test does not do much. It receives a message and sends an acknowledgement message. Its sole purpose is to show you how to use SoapUI to query a SOAP web service.

1. On your VM desktop, double-click the **SoapUI 5.2.0** icon.
   This opens the SoapUI client. This tool is used by developers to test web services. It mocks the behavior of a web service client: it connects to services, sends them requests, and receives answers from them.
2. In SoapUI, in the left pane, right-click **Projects**. To create a new SOAP test project, from the menu, select **New SOAP Project**.



3. The **New SOAP Project wizard** window opens. It needs one mandatory parameter to access a SOAP service: a path to the service's WSDL file.



New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name:

Initial WSDL: Browse...

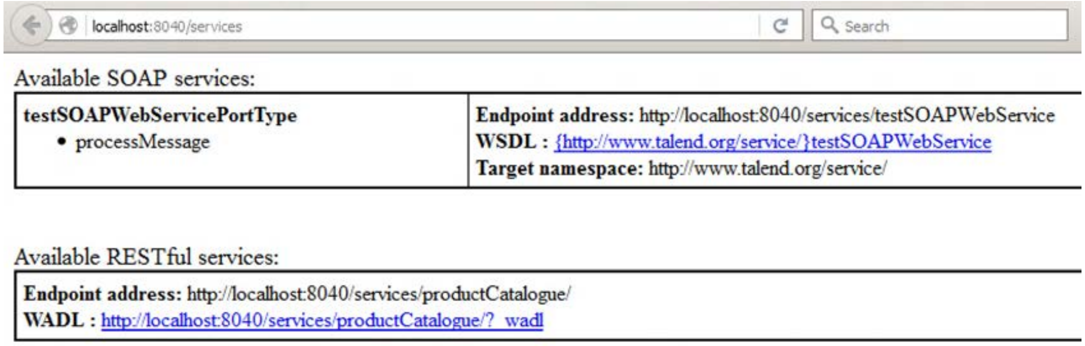
Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: □ Creates a TestSuite for the imported WSDL

Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

The WSDL file describes everything a SOAP service can do: it has the list of operations available, as well as the schema of the request message and the response message for each operation.

4. Open a web browser and navigate to **http://localhost:8040/services**. This is the endpoint where Talend Runtime deploys web services by default. This page lists all the SOAP and REST services deployed on the runtime instance.



Available SOAP services:

testSOAPWebServicePortType • processMessage	**Endpoint address:** http://localhost:8040/services/testSOAPWebService **WSDL :** {http://www.talend.org/service/}testSOAPWebService **Target namespace:** http://www.talend.org/service/

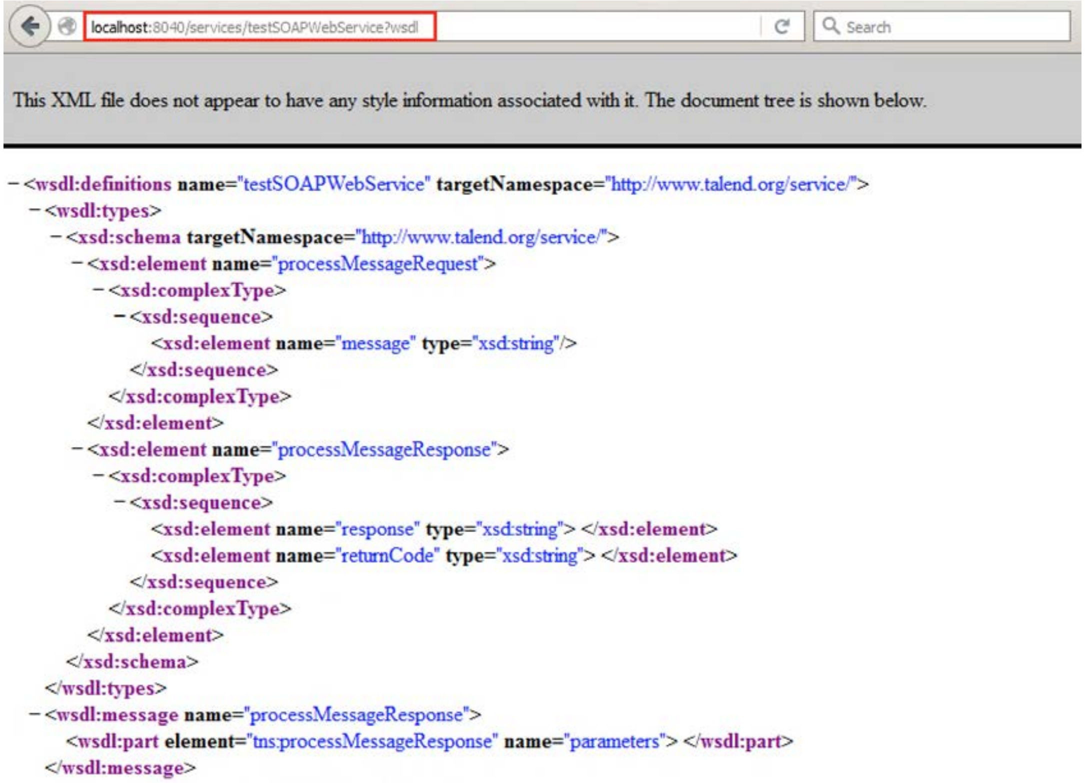
Available RESTful services:

Endpoint address: http://localhost:8040/services/productCatalogue/
WADL : http://localhost:8040/services/productCatalogue/?_wadl

As you can see, two services are already deployed on the runtime instance: one SOAP service called **testSOAPWebService** and one REST service called **productCatalogue**.

5. Access the WSDL file of the SOAP service by clicking the **WSDL** link on the services web page. If you see an XML message in your browser, this is the WSDLfile content and it proves the web service is deployed. If an error message is displayed, your services has failed to deploy..

Copy the WSDLURL file:

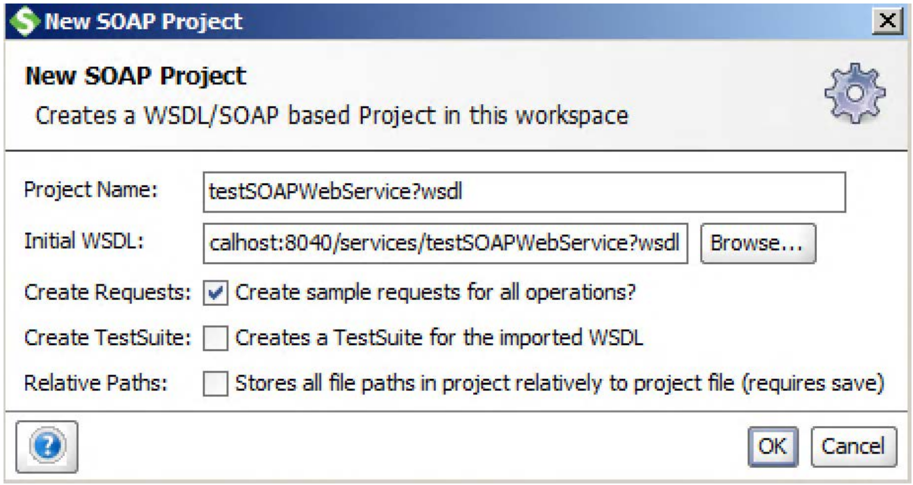


localhost:8040/services/testSOAPWebService?wsdl

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <wsdl:definitions name="testSOAPWebService" targetNamespace="http://www.talend.org/service/">
  – <wsdl:types>
    – <xsd:schema targetNamespace="http://www.talend.org/service/">
      – <xsd:element name="processMessageRequest">
        – <xsd:complexType>
          – <xsd:sequence>
              <xsd:element name="message" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      – <xsd:element name="processMessageResponse">
        – <xsd:complexType>
          – <xsd:sequence>
              <xsd:element name="response" type="xsd:string"> </xsd:element>
              <xsd:element name="returnCode" type="xsd:string"> </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:schema>
    </wsdl:types>
  – <wsdl:message name="processMessageResponse">
      <wsdl:part element="tns:processMessageResponse" name="parameters"> </wsdl:part>
    </wsdl:message>
```

6. Back in SoapUI, paste the WSDL URL in the **New SOAP Project** wizard, in the **Initial WSDL** field. Click **OK**.



New SOAP Project

New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: testSOAPWebService?wsdl

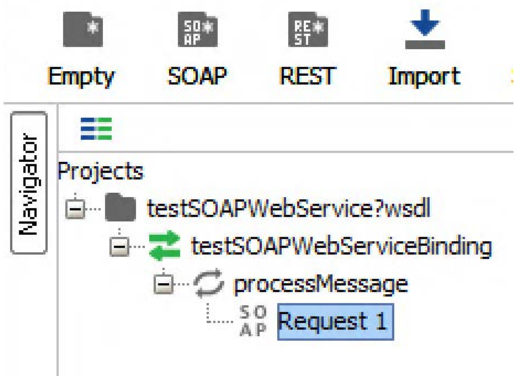
Initial WSDL: calhost:8040/services/testSOAPWebService?wsdl

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: □ Creates a TestSuite for the imported WSDL

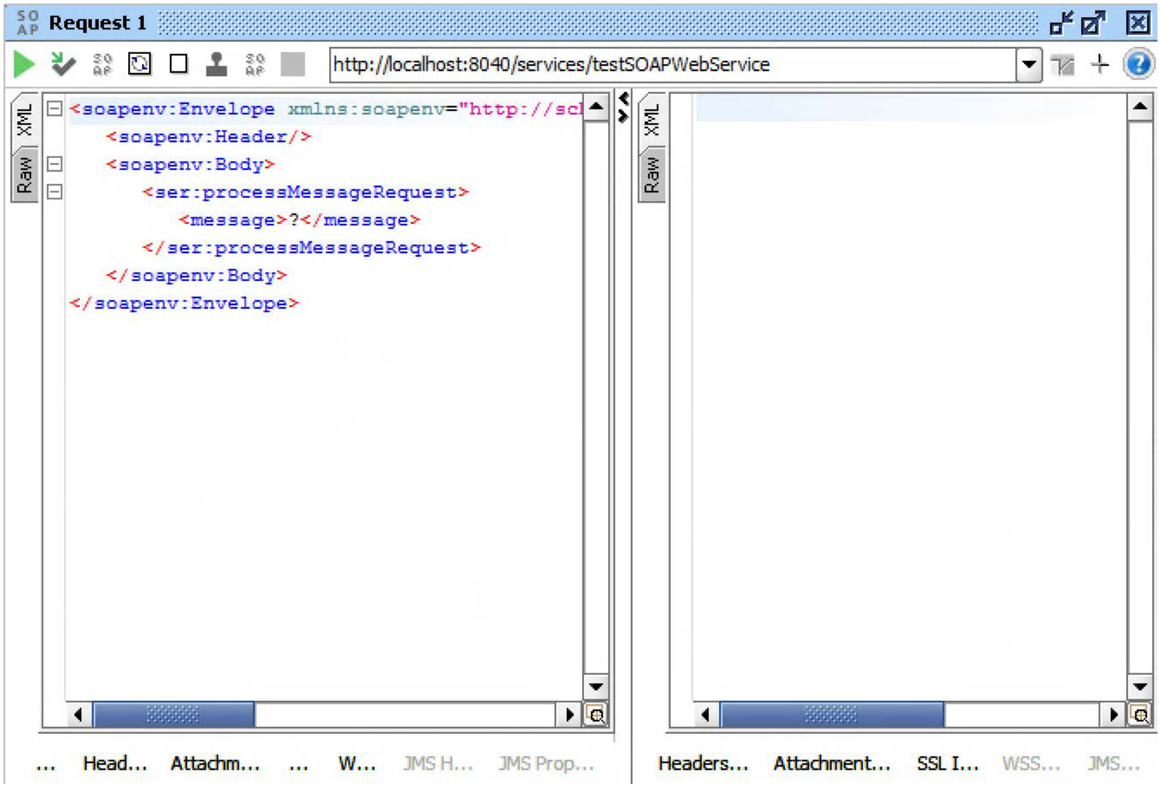
Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

7. SoapUI retrieves the WSDL file and connects to the SOAP web service, then creates a test case for each available operation.



As you can see, only one operation is available for **testSOAPWebService**, called **processMessage**. Expand the operation to reveal **Request 1**.

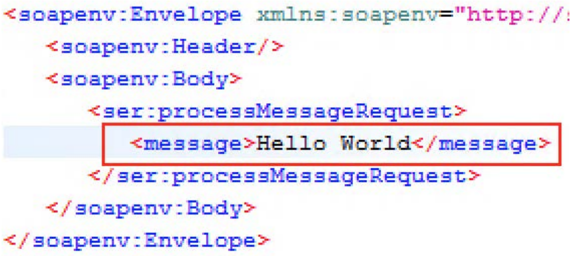
Double-click **Request 1**. This opens a new testing window for the **processMessage** operation. This window is divided into two parts:
The left side is dedicated to the web service request message, which SoapUI sends to the service.
The right side is reserved for the web service response message, which SoapUI receives from the service.



```
<soapenv:Envelope xmlns:soapenv="http://scl
   <soapenv:Header/>
   <soapenv:Body>
      <ser:processMessageRequest>
         <message>?</message>
      </ser:processMessageRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

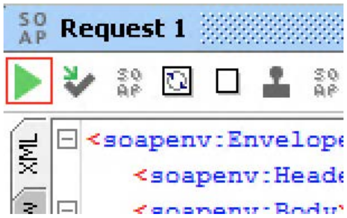
Note that parameters in the request are given a question-mark value by default.

1. In the left-side window, change the request and give the message parameter any value.

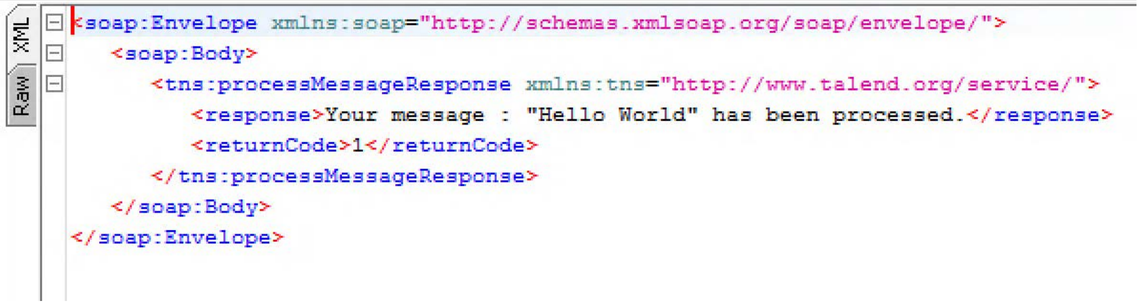


```
<soapenv:Envelope xmlns:soapenv="http://
   <soapenv:Header/>
   <soapenv:Body>
      <ser:processMessageRequest>
         <message>Hello World</message>
      </ser:processMessageRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

2. To send your request to the web service, click the **Play** button.



3. The web service answers in the right-side part of the window. Your answer should look like this:

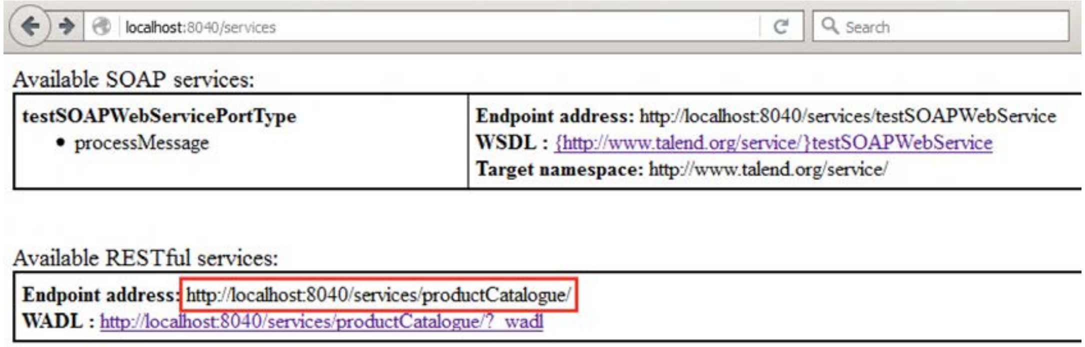


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <tns:processMessageResponse xmlns:tns="http://www.talend.org/service/">
         <response>Your message : "Hello World" has been processed.</response>
         <returnCode>1</returnCode>
      </tns:processMessageResponse>
   </soap:Body>
</soap:Envelope>
```

REST service

The productCatalogue service you are about to test is a simple REST web service that returns a list of products along with its catalog information.

1. In SoapUI, in the left pane, right-click **Projects.** To create a new REST test project, on the menu, select **New REST Project**.
2. REST services do not have WSDL files. SoapUI requires only that the service URL be connected in order to create a test project. In the web browser, navigate to **http://localhost:8040/services**. The endpoint URL of the REST service is displayed on the page. Copy the URL.



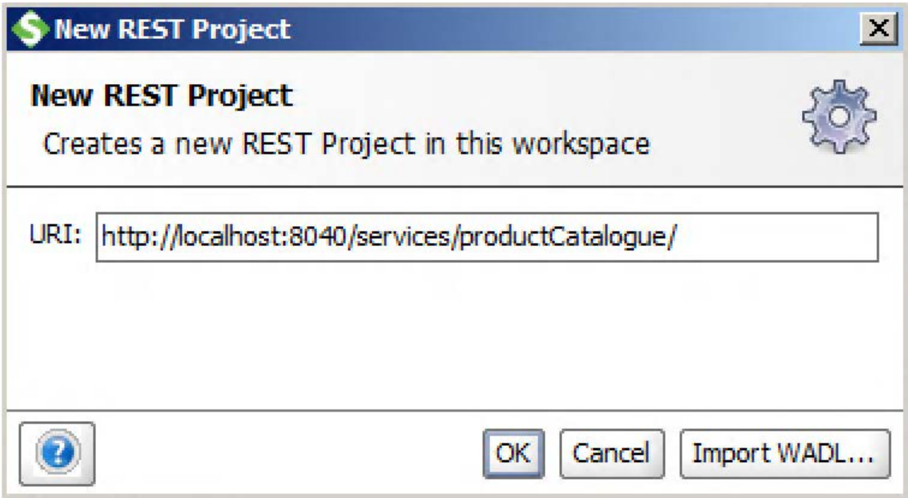
Available SOAP services:

testSOAPWebServicePortType • processMessage	**Endpoint address:** http://localhost:8040/services/testSOAPWebService **WSDL :** {http://www.talend.org/service/}testSOAPWebService **Target namespace:** http://www.talend.org/service/

Available RESTful services:

Endpoint address: http://localhost:8040/services/productCatalogue/
WADL : http://localhost:8040/services/productCatalogue/?_wadl

3. In SoapUI, in the **New REST Project** window, paste the service URL and click **OK**.

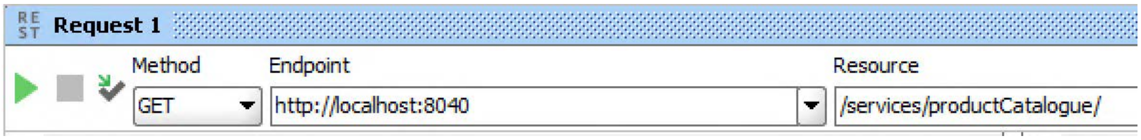


New REST Project

Creates a new REST Project in this workspace

URI: http://localhost:8040/services/productCatalogue/

4. A new REST project opens. As you can see at the top of the main window, a REST service is queried using three parameters:

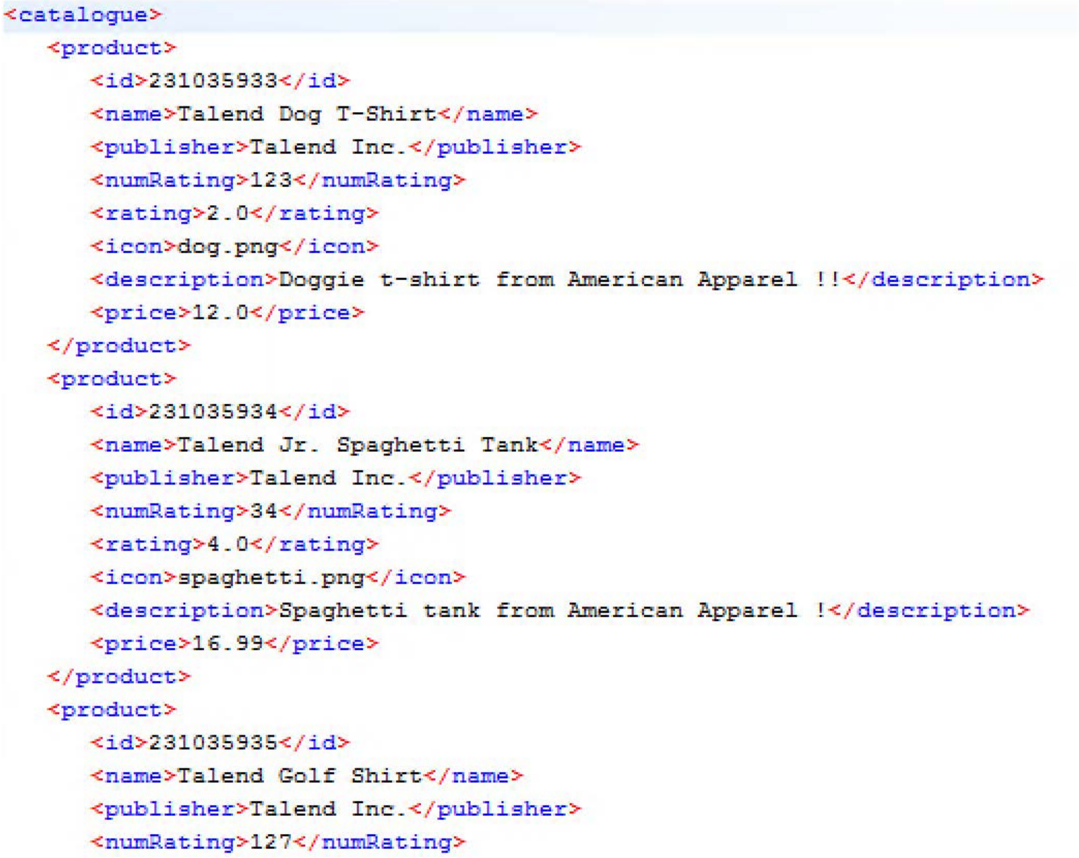


Method	Endpoint	Resource
GET	http://localhost:8040	/services/productCatalogue/

- The **Endpoint** is the server name and port hosting the service
- The **Resource** is the part of the URL that describes which service you access
- The **Method** is the action you take on the resource. With Talend ESB, there are four possible actions: **GET, POST, PUT**, and **DELETE**. **GET** retrieves data from the resource, whereas **POST** writes data to the resource, **PUT** updates data, and **DELETE** deletes data.

5. Make sure the **GET** method is selected, then choose an action from the **Method** drop-down list.

6. Click the **Play** button. The service gets data from the resource **productCatalogue** and should return an XML message with product information. This message appears in the SoapUI main window.



```
<catalogue>
   <product>
      <id>231035933</id>
      <name>Talend Dog T-Shirt</name>
      <publisher>Talend Inc.</publisher>
      <numRating>123</numRating>
      <rating>2.0</rating>
      <icon>dog.png</icon>
      <description>Doggie t-shirt from American Apparel !!</description>
      <price>12.0</price>
   </product>
   <product>
      <id>231035934</id>
      <name>Talend Jr. Spaghetti Tank</name>
      <publisher>Talend Inc.</publisher>
      <numRating>34</numRating>
      <rating>4.0</rating>
      <icon>spaghetti.png</icon>
      <description>Spaghetti tank from American Apparel !</description>
      <price>16.99</price>
   </product>
   <product>
      <id>231035935</id>
      <name>Talend Golf Shirt</name>
      <publisher>Talend Inc.</publisher>
      <numRating>127</numRating>
```

Now that you know how to test both SOAP and REST services from the SoapUI interface, you can move on to the next section.

Messaging with Apache ActiveMQ

Overview

Talend ESB embeds its own messaging platform: Apache ActiveMQ. If your company already uses a JMS standards-based message-oriented middleware solution, you are free to use it in your routes and services. And if message management is new to your company, Apache ActiveMQ is both efficient and robust enough to satisfy your business needs.

This section will guide you through the basic features of ActiveMQ.

Starting ActiveMQ

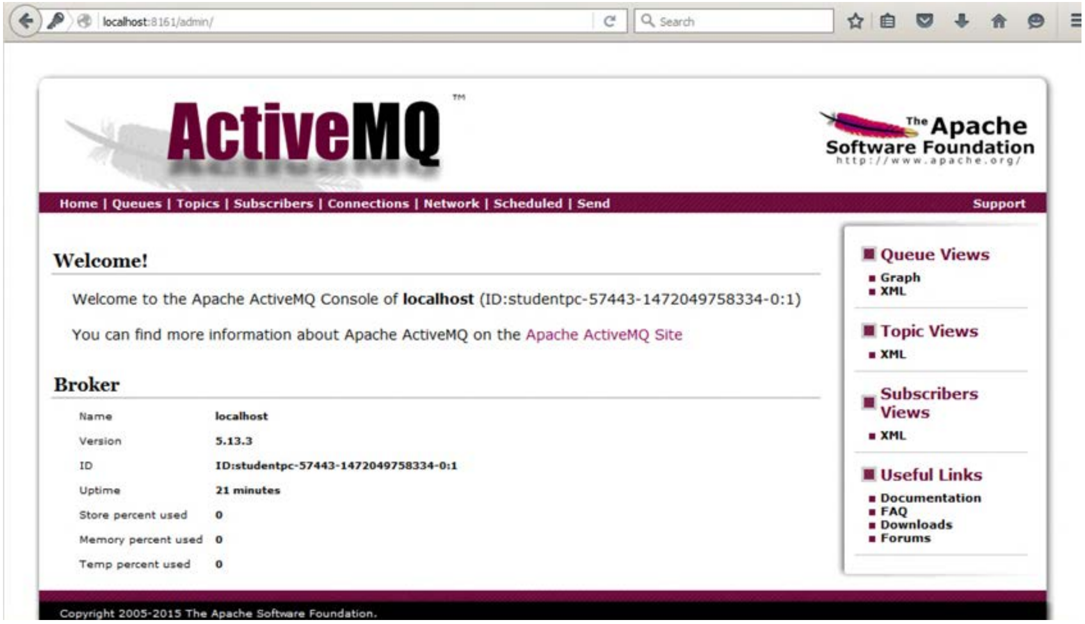
Unlike all the other Talend modules, ActiveMQ is not installed as a service, so on your training VM, you need to start it manually.

1. ActiveMQ is part of the ESB package. To find ActiveMQ, navigate to **C:\Talend\6.2.1\esb\activemq**.
2. In the **.\bin** folder, you will find one directory per operating system. Navigate to **.\bin\win64**.
3. This folder holds several bat files. One of them can be used to install ActiveMQ as a service, but if you just want to start the ActiveMQ server, double-click **activemq.bat**. A terminal opens and logs the ActiveMQ starting tasks. Leave this terminal window running.

Administering ActiveMQ

ActiveMQ offers an administration web interface that lets you manage queues and topics, browse messages, declare subscribers, and send test messages.

1. Open a web browser and go to **http://localhost:8161/admin**.
2. ActiveMQ is password protected. To connect, enter the following:
   user: *admin*
   password: *admin*
3. On the menu, click **Queues**, then **Topics**, and explore the pages. Each page displays information such as queues/topics created, number of pending messages, and number of connected consumers/subscribers. To explore these options, you will now send a test message.



Welcome!

Welcome to the Apache ActiveMQ Console of **localhost** (ID:studentpc-57443-1472049758334-0:1)

You can find more information about Apache ActiveMQ on the Apache ActiveMQ Site

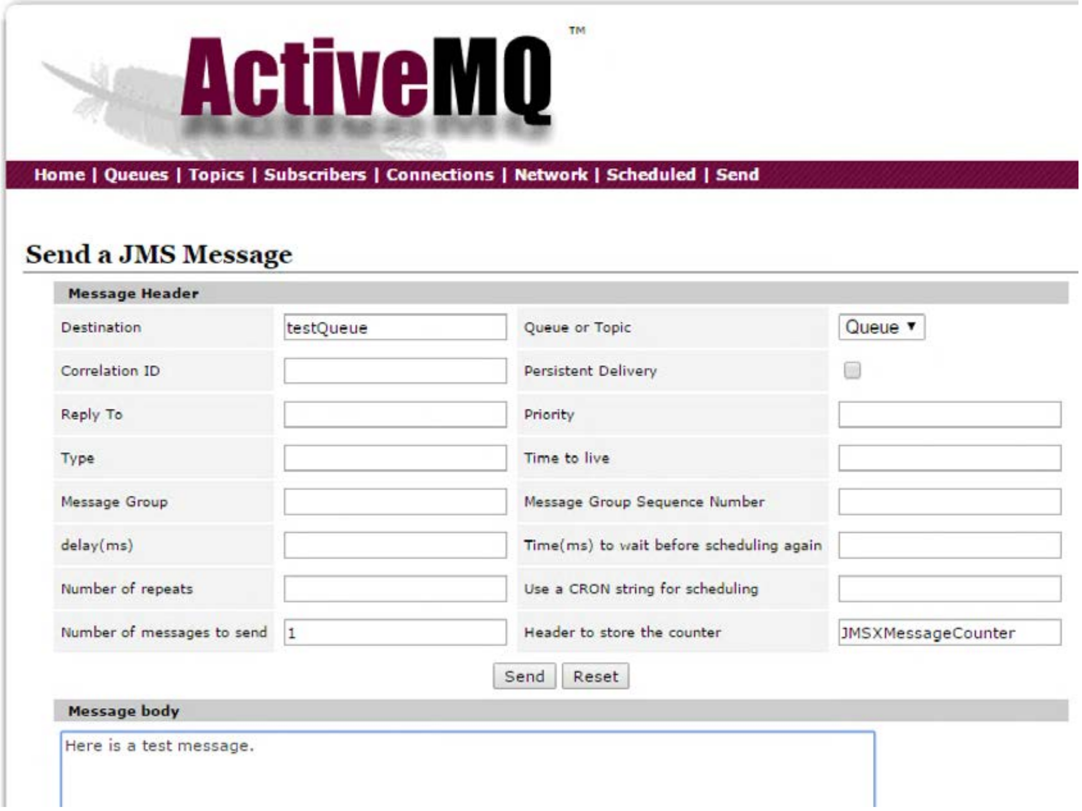
Broker

Name	localhost
Version	5.13.3
ID	ID:studentpc-57443-1472049758334-0:1
Uptime	21 minutes
Store percent used	0
Memory percent used	0
Temp percent used	0

Copyright 2005-2015 The Apache Software Foundation.

4. On the top menu bar, click **Send**.
5. This page allows you to send a test message to a queue or topic, with a wide range of custom parameters. This feature is very useful if you need to test a specific option on your queues/topics or test any ESB interface triggered by messages.
   To send a test message to a new queue, fill in the fields as follows:
   Queue or Topic: *Queue*

Destination: *testQueue*
Message body: *Here is a test message.*


ActiveMQ™
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Send a JMS Message

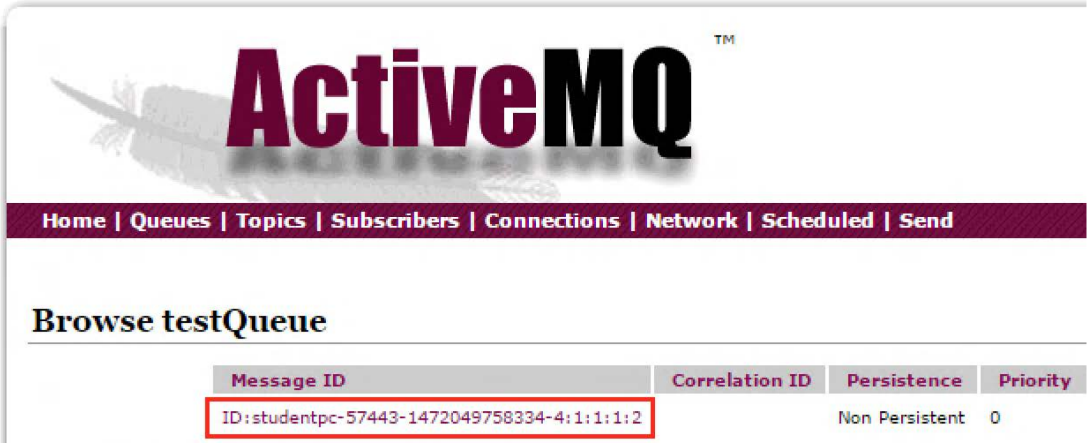
Message Header			
Destination	testQueue	Queue or Topic	Queue ▾
Correlation ID		Persistent Delivery	☐
Reply To		Priority	
Type		Time to live	
Message Group		Message Group Sequence Number	
delay(ms)		Time(ms) to wait before scheduling again	
Number of repeats		Use a CRON string for scheduling	
Number of messages to send	1	Header to store the counter	JMSXMessageCounter

Message body

Here is a test message.

Click **Send**.

6. In the menu, click **Queues**.
   The Queues administration page shows the new queue, testQueue, to which you sent your message. As you can see from the table displayed, this queue has one pending message and no consumers connected.
7. To drill down, click on the queue **testQueue**.
8. The browse testQueue page displays all pending messages. Each message has a unique ID (usually generated by the system of origin). You should see one message, the one you sent earlier. To drill down to the message level, click the message ID.


ActiveMQ™
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Browse testQueue

Message ID	Correlation ID	Persistence	Priority
ID:studentpc-57443-1472049758334-4:1:1:1:2		Non Persistent	0

9. At the message level, you can see all message configuration parameters, as well as the message content.

Next step

You have almost completed this lesson. Continue to the Wrap-up section for a review of the concepts we covered.

Wrap-up

In this lesson, you learned about the Talend ESB infrastructure and features, including web services and ActiveMQ.

You can now describe and test both SOAP and REST web services. You know how to start an ActiveMQ server, and how to administer it through its administration web interface. You can send test messages to mock queues and topics, and you know how to explore queues, topics, and browse messages in the web interface.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let’s continue >**.

This page intentionally left blank to ensure new chapters
start on right (odd number) pages.

LESSON 2

SOAP and REST Services

This chapter discusses the following.

Lab overview

Talend Runtime is the execution environment for all ESB features--services and routes. It easily integrates with any prior Talend infrastructure and adds real-time synchronous and asynchronous capabilities, as well as network features that a JobServer cannot handle.

This lesson shows how to access, declare, configure, and use Talend Runtime in a professional environment.

Objectives

After completing this lesson, you will be able to:

Add a new instance of Talend Runtime to an existing Talend infrastructure

Connect to the runtime with an SSH client

Edit runtime configuration files

Run basic runtime commands

Next steps

Now you can connect to TAC and add the runtime to your architecture.

Declaring a New Runtime Instance in TAC

Overview

This section shows how you can easily add a new runtime instance to your Talend infrastructure.

Checking the status of a runtime instance

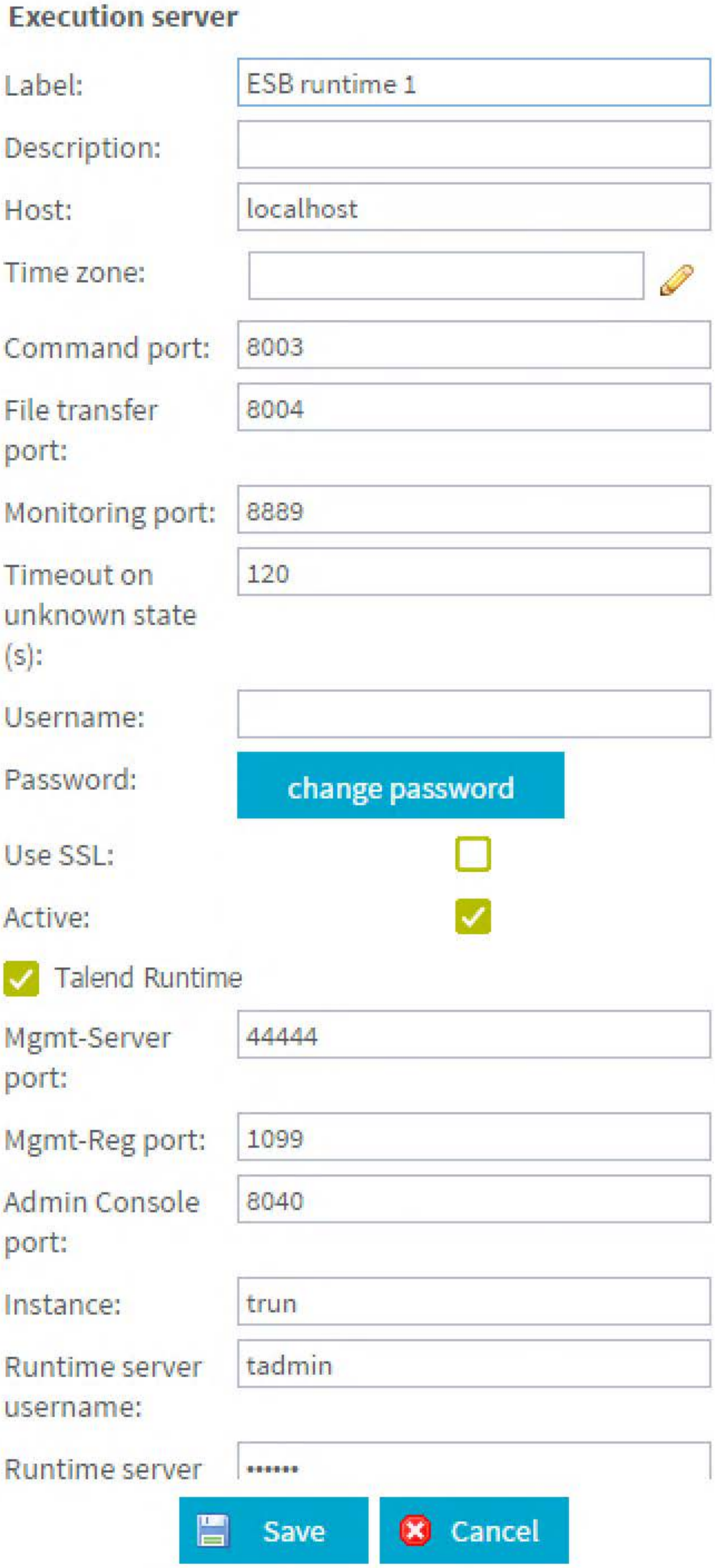
A Talend runtime instance has been installed on your training VM. Before using it, make sure this instance is up and running:

1. From the taskbar, open the **Services** window.
2. Confirm that the service **Talend Runtime 6.2.1** is running. If not, start it.

Declaring a new server in TAC

1. Connect to TAC via the standard URL, http://localhost:8080/org.talend.administrator, as user *operator@company.com* (password: *operator*).
2. To open the server management page, on the TAC left menu, under **Conductor**, click **Server**.
3. To declare a new runtime instance in the TAC, on the **Server** page, click the **Add** button, then **Add Server**.
4. Fill in the Execution Server parameters as follows:
   Label: *ESB runtime 1*
   Host: *localhost*
   Command port: *8000*
   File transfer port: *8001*
   Monitoring port: *8888*
5. Select the **Talend Runtime** check box. This indicates to TAC that you are declaring a new ESB runtime instance and not a new JobServer. The default additional runtime parameters should be OK. Confirm that they correspond to the following:
   Mgmt-Server port: *44444*
   Mgmt-Reg port: *1099*
   Admin Console port: *8040*
   Instance: *trun*

6. Click **Save**. Your runtime instance is now declared and monitored in TAC, ready to receive artifacts for deployment.



Execution server

Label:	ESB runtime 1
Description:	
Host:	localhost
Time zone:	
Command port:	8003
File transfer port:	8004
Monitoring port:	8889
Timeout on unknown state (s):	120
Username:	
Password:	change password
Use SSL:	□
Active:	☑

☑ Talend Runtime

Mgmt-Server port:	44444
Mgmt-Reg port:	1099
Admin Console port:	8040
Instance:	trun
Runtime server username:	tadmin
Runtime server	••••••

You will now explore Talend Runtime by accessing it directly and learning some basic commands.

Accessing Talend Runtime and Sending Basic Commands

Overview

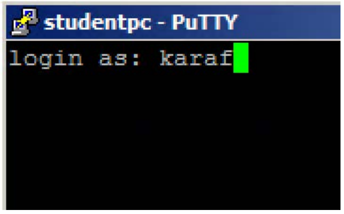
Talend Runtime troubleshooting is usually faster and more efficient if you know where to look for logs, configurations, and system status checks. You can do most administrative operations by directly connecting to a runtime instance.

This section shows how to connect to a runtime instance via an SSH client and use basic runtime commands.

Connecting to Talend Runtime via SSH

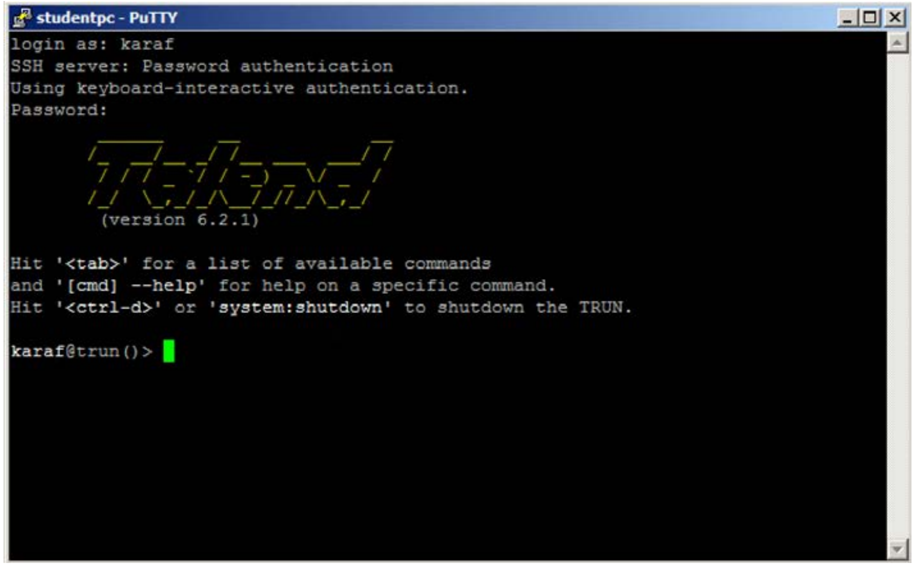
To access Talend Runtime, you can use any SSH client. On the Windows training VM, you use a PuTTy client, which is both simple and reliable.

1. On the desktop of your VM, to open the client, double-click the **PuTTy** icon.
2. In the **PuTTy configuration** window, enter the following parameters:
   host: *localhost*
   port: *8101*
   To open a new connection to the runtime instance, click the **Open** button.
   Note: By default, any runtime instance can be reached on port 8101.
3. Access to the Talend Runtime console is password protected. To log in, enter *karaf* as the username and password.

   ```
   login as: karaf
   ```

   You are now logged into the console.

   ```
   login as: karaf
   SSH server: Password authentication
   Using keyboard-interactive authentication.
   Password:

       ______      __                 __
      /_  __/___ _/ /__  ____  ____/ /
       / / / __ `/ / _ \/ __ \/ __  /
      / / / /_/ / /  __/ / / / /_/ /
     /_/  \__,_/_/\___/_/ /_/\__,_/
         (version 6.2.1)

   Hit '<tab>' for a list of available commands
   and '[cmd] --help' for help on a specific command.
   Hit '<ctrl-d>' or 'system:shutdown' to shutdown the TRUN.

   karaf@trun()>
   ```

Running basic commands

Before you run commands in Talend Runtime, consider these basic guidelines:

You can get a list of all available commands by typing *help*

You can access any command documentation by typing *--help*

You can use the tab key for command completion

Talend Runtime implements some basic shell commands, such as 'grep' or the pipe character to chain commands

List command

1. The *list* command lists all features installed in your runtime instance. Run the command by typing *list*, then press Enter.
2. You can refine the *list* command by chaining a grep command to it. Use the following: *list | grep REST*.
   This selects the features from the list matching the term REST. You will see some system features as well as the testRESTWebService that you tested earlier.

Log command

1. You can access the runtime logs with any command beginning with *log*. In the runtime client interface, type *log*, then press TAB to get a list of all available log commands.
2. Try the *log:display* command. This displays all runtime logs at once.
3. Try the *log:tail* command. This is the equivalent of a tail on a file on Linux, but this one is specific to the runtime logs. To exit this command, press the Ctrl and C keys.

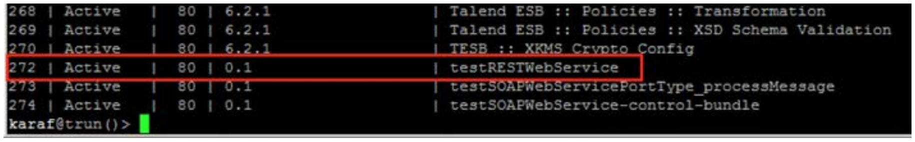
tesb:start- and tesb:stop- commands

1. You can start and stop Talend-specific runtime features with the *tesb:start* command. To get a list of the available modules, in the Runtime console, type *tesb:start*, then press the tabulation key. Try the following command: *tesb:start-locator*. This starts the Service Locator, which you will need later in this course.
2. To see if your command installed or activated any module, run the following command: *list | grep Locator* (capitalize "Locator").
   As you can see, several Locator modules were installed and activated.
3. You can stop any feature by using the command *tesb:stop*. Try *tesb:stop-locator*.
4. Run *list | grep Locator* again. As you can see, modules were either stopped or removed.
5. Finally, in order to prepare future labs, run these two commands:
   > *tesb:start-locator* to start the service locator module
   > *tesb:start-sam* to start the Service Activity Monitoring module

Start and stop bundles

It is very easy to start and stop bundles when accessing the Runtime console.

1. In the Runtime console, type the command *list* and locate the **testRESTWebService** bundle, currently listed as **Active**.



```
268 | Active   |  80 | 6.2.1                    | Talend ESB :: Policies :: Transformation
269 | Active   |  80 | 6.2.1                    | Talend ESB :: Policies :: XSD Schema Validation
270 | Active   |  80 | 6.2.1                    | TESB :: XKMS Crypto Config
272 | Active   |  80 | 0.1                      | testRESTWebService
273 | Active   |  80 | 0.1                      | testSOAPWebServicePortType_processMessage
274 | Active   |  80 | 0.1                      | testSOAPWebService-control-bundle
karaf@trun()>
```

2. Enter the command *stop testRESTWebService.*
3. Run a *list* command again. The testRESTWebService bundle is now listed as Resolved, which means it has been stopped.
4. In a web browser, go to **http://localhost:8040/services**. You can see that the REST web service is stopped; it is no longer listed on the runtime service page.
5. To restart the service, in the Runtime console, enter *start testRESTWebService*.

Installing runtime features

Talend Runtime embeds features, which are modules you can activate or deactivate according to your needs. For instance, we will start the ActiveMQ client. This feature installs all dependencies needed to connect to an ActiveMQ broker. You need to start this feature because in a future lab, you will send messages from the runtime instance to ActiveMQ.

1. In the Runtime console, enter *feature:list* | *grep -i activemq*. This lists all features related to ActiveMQ.
2. As you can see from the command result, all ActiveMQ features are uninstalled for now. The feature you want to install is **activemq-client**.
3. To install this feature, in the Runtime console, enter *feature:install activemq-client*.
4. Again enter the *feature:list* | *grep -i activemq* command to confirm that the activemq-client feature was installed and started.

You can run many more commands in the Runtime console. We recommended reading the Talend documentation and using the Talend Runtime embedded documentation if your administrator tasks require a deep understanding of these commands.

Configuration files

All runtime features have configuration parameters. These parameters are stored in files, and all the files are consistently located in the same folder so that configuration is easy to both find and update.

1. In a file explorer, navigate to **C:\Talend\6.2.1\runtime\etc**. All runtime configuration files are stored in the **etc** folder.
2. Notice that each file name is dedicated to just one feature. In **Notepad++,** open the file **org.ops4j.pax.web.cfg**.
3. This file holds the web configuration of the runtime instance. As you can see, this is where you can define the port on which services will be hosted, as well as SSL parameters. Change the service port from *8040* to *8041* and save the file.
4. In a browser, go to **http://localhost:8040/services**. You get an error message, since your services are now hosted at http://localhost:8041/services. The update in your configuration was applied when you saved the file. The runtime does not need restarting for most configuration updates.
5. In the **org.ops4j.pax.web.cfg** file, revert your update so that services are hosted on port *8040* and save the file.

You can find a description of configuration files in the documentation for the related module or feature. We recommend that you thoroughly read module documentation before changing any parameter.

Next step

You have almost completed this lesson. Continue to the Wrap-up section for a review of the concepts we covered.

Wrap-up

In this lesson you learned how to add a runtime instance to a Talend infrastructure in TAC. You also accessed a runtime console via an SSH client, sent commands to read logs, started/stopped features, and got a full list of deployed bundles, along with their statuses. Finally, you edited a configuration file and saw the effect applied to the runtime instance.

For more information on runtime commands and configuration, refer to the Talend ESB documentation.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let’s continue >**.

This page intentionally left blank to ensure new chapters start on right (odd number) pages.

LESSON 3

Deploying Services Manually - ESB Runtime

This chapter discusses the following.

Deploying Services Manually–ESB Runtime

Overview

In this lab, you will use several methods to deploy web services on your runtime instance. You already have a solid background in Talend DI administration, so this chapter focuses mostly on the slight differences between DI Admin and ESB Admin when deploying ESB features. However, the overall ESB deployment process is very similar to the DI process.

This chapter shows how to:

Deploy a service manually

Use Talend Administration Center (TAC) to deploy a service from an artifact stored on Nexus

Publish a service on Nexus and deploy it from TAC

Publish a route on Nexus and deploy it from TAC

Moreover, now that you know how to test web services using SoapUI, you can check every service deployed in Talend Runtime.

Objectives

After completing this lesson, you will be able to:

Deploy an ESB feature on Talend Runtime without using TAC

Deploy an ESB feature on Talend Runtime from Nexus by using TAC

Check the status and test your web services

Check the status and test your routes

Make sure Talend Runtime and TAC are still running as Windows services, then go to the first lab section.

Manual Deployment

Overview

In some ESB infrastructures, strong security policies lead designers to isolate runtimes from the rest of the infrastructure. In such cases, runtimes are connected to neither the TAC server nor the Nexus repository. As a result, it is impossible to follow the recommended deployment process that you know from your DI experience.

If a runtime instance has no way of contacting other machines in the information system, deploying using TAC is not possible. This is why Talend Runtime offers another manual deployment option.

Manually deploying a package

Developers can build and export packages from Talend Studio. Depending on the nature of the feature they contain, these packages are either JAR, .ZIP, or .KAR files.

1. On your VM, open a file explorer and navigate to the Runtime repository **C:\Talend\6.2.1\runtime**.
2. Open the **deploy** folder.
3. This folder holds two interesting files: *testRESTWebService-0.1.jar* and *testSOAPWebService-0.1.kar*. These are the two web services you tested earlier with SoapUI. They were deployed manually. Indeed, the **deploy** folder has a specific behavior: any package you drop in here will be automatically and immediately deployed by the runtime.
4. Cut the file *testRESTWebService-0.1.jar* and paste it in the parent folder (**C:\Talend\6.2.1\runtime**).
5. In your web browser, go to **http://localhost:8040/services** and note that the REST service has disappeared. This means one of two things: it is stopped (in Resolved state) or uninstalled.
6. One way to know exactly what happened is to connect to the runtime console. Open a new **PuTTy** client window and connect to the runtime instance (on *localhost*, on port *8101*; when prompted, enter *karaf* as the log-in ID and password).
7. To check the REST web service, in the console, run the *list* command.
   The testRESTWebService has disappeared. This means it has not only been stopped, but also uninstalled: when an item is removed from the deploy folder, it is uninstalled from the runtime instance.
8. Cut the file *testRESTWebService-0.1.jar* and paste it in the **deploy** folder.
9. In the Runtime console, run the *list* command again. The REST web service has been installed and deployed. It is now listed as active and it has a new bundle ID.

Manual deployment can be very useful when you cannot follow the best-practice deployment cycle due to security or infrastructure constraints. Keep in mind that you should only use manual deployment when no other option is available. When deploying manually, you lose many of the Talend administration features, such as Service Locator, Service Activity Monitoring, and Service Registry.

You can now move to the next section and learn about recommended deployment by using Nexus and TAC.

Deploying Services from TAC

Overview

In this lab, you will learn how a user with operations manager privileges can use the ESB Conductor page to deploy a service from Nexus to a runtime instance.

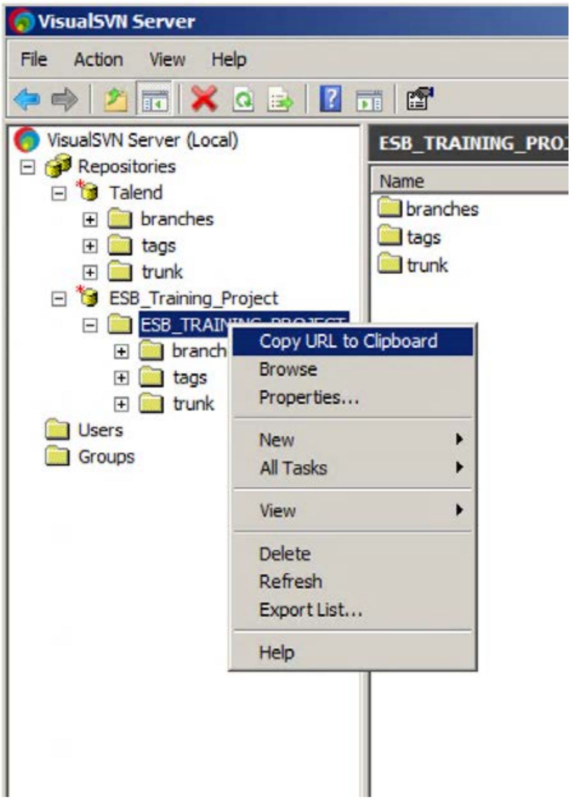
Preparing for deployment

As you know from your Talend DI experience, you can only deploy an artifact from TAC if it is linked to a project. An artifact is stored in Nexus for you to deploy on the runtime, and some sources are stored in an SVN dump file for further publishing during this course. To be able to use these resources, you need to load the SVN dump and create the corresponding project in TAC, then manage its privileges so your users can access it.

1. From the **Start** menu in Windows, open **VisualSVN Server Manager**.
2. Right-click **Repositories** and select **Import Existing Repository...**
3. In the **Import Existing Repository** wizard, select the option **Load repository from a dump file** and click **Next**.
4. Click the **Browse** button and go to **C:\StudentFiles\ESB_Training_Project.svn_dump**. Click **Open**, then **Next**.

The Import Existing Repository wizard suggests **ESB_Training_Project** as the **Destination repository name**. Approve by clicking **Next**.

1. As security is not a key issue in this training, and you already know how to manage permissions both on SVN and in the TAC, set the default permission to **All Subervion users have Read / Write access** and click **Import**.
2. Click **Finish**.
3. In **VisualSVN Server Manager**, in the **Repositories** browser, unfold the new **ESB_Training_Project**. Right-click the **ESB_TRAINING_PROJECT** folder and, on the contextual menu, click **Copy URL to Clipboard**.



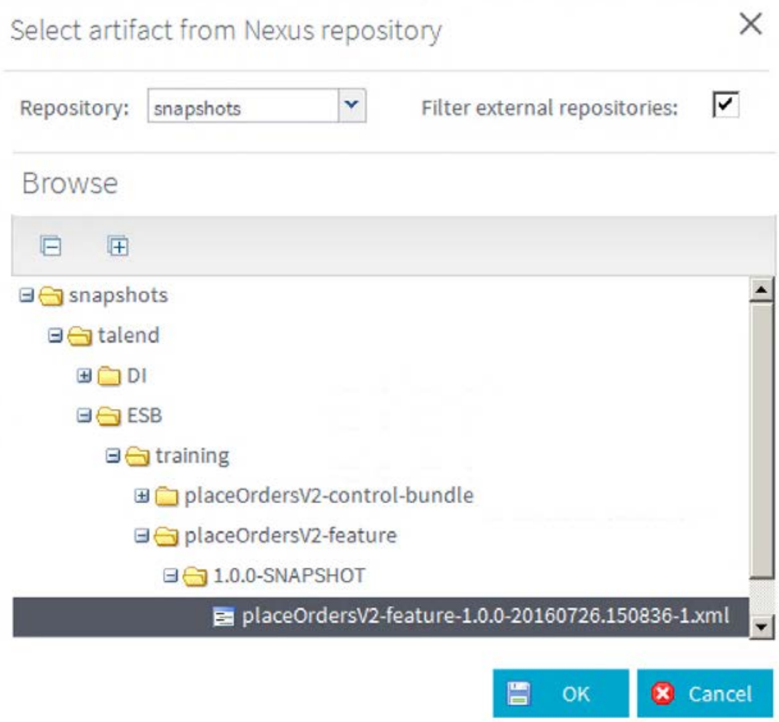
4. Connect to the TAC as the administrator user (*admin@company.com/admin*).
5. On the TAC menu, in **Settings**, click **Projects**.

6. To add a new project, click **Add**.
7. In the right-side **Project** window, fill in the fields as follows:
   Label: *ESB_Training_Project* (enter this exactly as capitalized here)
   Project type: *Data Integration/ESB*
   Storage: *SVN*
8. Select the **Advanced settings** check box and fill in the fields:
   Url: (paste in the repository URL you copied in Step 8)
   Login: *tadmin*
   Password: *tadmin*
9. To create the project, click **Save**.
10. On the TAC menu, in **Settings**, click **Project authorizations** and assign user **operator@company.com** read rights on the **ESB_Training_Project**.
11. Log out of the TAC.

Deploying a Nexus-hosted web service

A web service from the ESB_Training_Project is stored in Nexus. This service allows clients to place orders. Each new order is stored in a database table and sent in parallel as a JMS message to ActiveMQ. Prior to running the service, you need to make sure ActiveMQ is running so that the service can deliver its message to a queue.

1. If you closed the ActiveMQ application window, in a file explorer, navigate to **C:\Talend\6.2.1\esb\activemq\bin\win64** and execute **activemq.bat**. If ActiveMQ is still running on your VM, you do not need to change anything.
2. Connect to TAC as the operator user (*operator@company.com/operator*).
3. On the TAC menu, under **Conductor**, click **ESB Conductor**. This page is very similar in appearance and purpose to the Job Conductor page. It allows operators to deploy ESB artifacts.
4. To add a new task, click **Add > Task**.
5. In the **Label** field, enter *placeOrder*.
6. Click **Select feature** and browse to **snapshots > talend > ESB > training > placeOrdersV2-feature > 1.0.0-SNAPSHOT > placeOrdersV2-feature-1.0.0-20160726.150836-1.xml**. Click **OK**.


Select artifact from Nexus repository

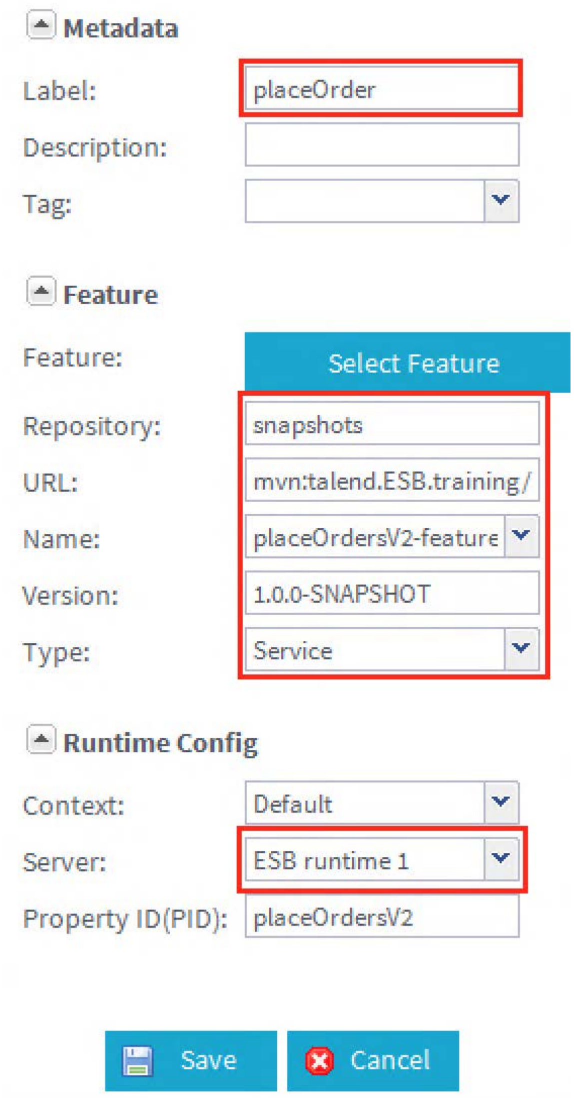
Repository: snapshots

Filter external repositories: ☑

Browse
snapshots
talend
DI
ESB
training
placeOrdersV2-control-bundle
placeOrdersV2-feature
1.0.0-SNAPSHOT
placeOrdersV2-feature-1.0.0-20160726.150836-1.xml
OK
Cancel

Note: The only element you need in order to deploy an artifact is its feature, which you can find in the **artifactName-feature** folder.

7. In the **Type** field, select **Service** instead of **Route**.
8. On the **Server** list, select **ESB runtime 1**. Your configuration should read as follows:



Metadata

Label: placeOrder

Description:

Tag:

Feature

Feature:

Repository: snapshots

URL: mvn:talend.ESB.training/

Name: placeOrdersV2-feature

Version: 1.0.0-SNAPSHOT

Type: Service

Runtime Config

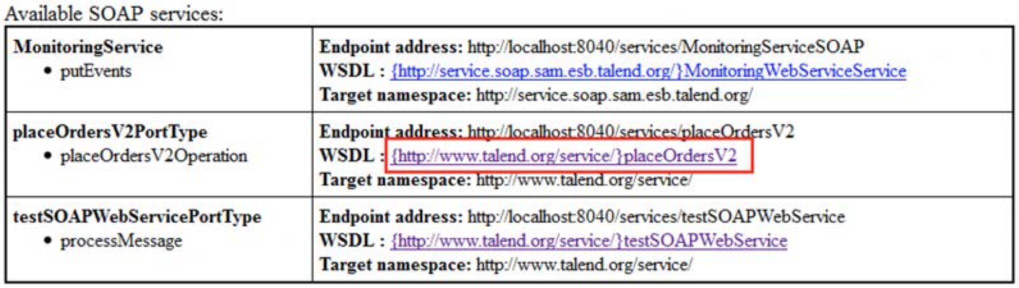
Context: Default

Server: ESB runtime 1

Property ID(PID): placeOrdersV2

9. Click **Save**. Your new task appears on the list.
10. Select **placeOrder** and click **Deploy**.
11. In a web brower, go to **http://localhost:8040/services** and confirm that the new service, **placeOrdersV2,** appears on the page.

12. Click the service **WSDL** link.



Available SOAP services:

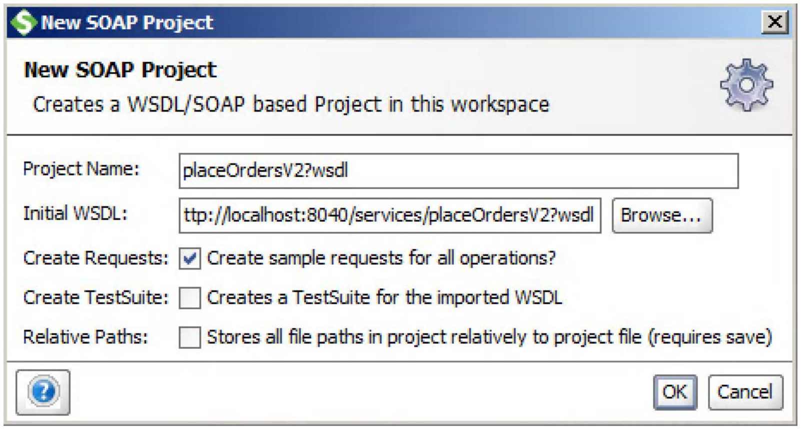
MonitoringService • putEvents	**Endpoint address:** http://localhost:8040/services/MonitoringServiceSOAP **WSDL :** {http://service.soap.sam.esb.talend.org/}MonitoringWebServiceService **Target namespace:** http://service.soap.sam.esb.talend.org/
placeOrdersV2PortType • placeOrdersV2Operation	**Endpoint address:** http://localhost:8040/services/placeOrdersV2 **WSDL :** {http://www.talend.org/service/}placeOrdersV2 **Target namespace:** http://www.talend.org/service/
testSOAPWebServicePortType • processMessage	**Endpoint address:** http://localhost:8040/services/testSOAPWebService **WSDL :** {http://www.talend.org/service/}testSOAPWebService **Target namespace:** http://www.talend.org/service/

13. Copy the URL of the WSDL page.

14. Open **SoapUI** and right-click **Projects**.

15. On the contextual menu, select **New SOAP Project**.

16. In the **New SOAP Project** window, in the **Initial WSDL** field, paste the WSDL URL. Click **OK**.



New SOAP Project

New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: placeOrdersV2?wsdl

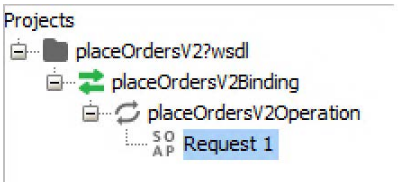
Initial WSDL: ttp://localhost:8040/services/placeOrdersV2?wsdl

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

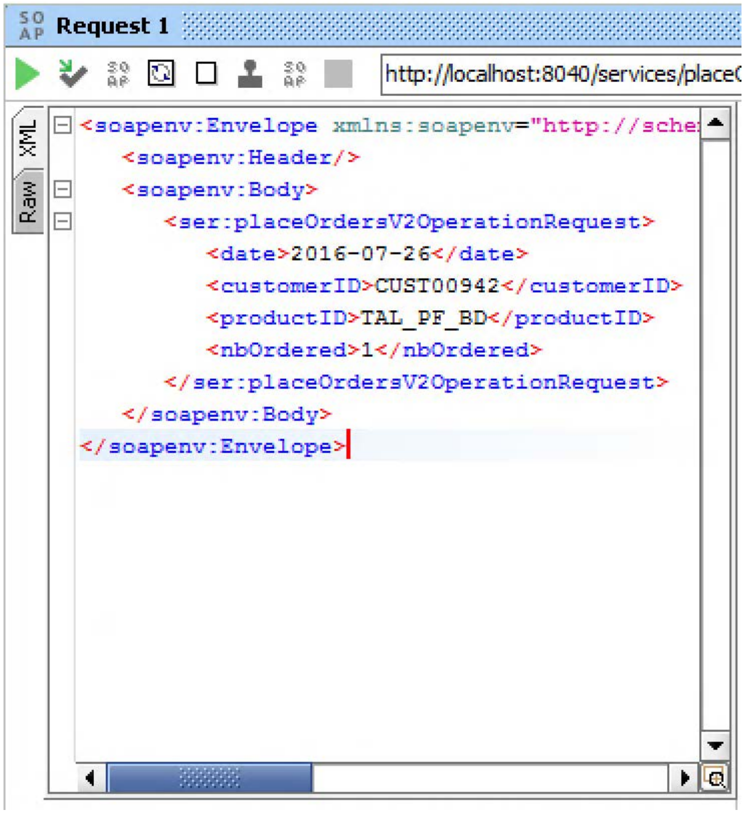
Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

17. A new test project is created. Double-click **Request 1**.



Projects
- placeOrdersV2?wsdl
  - placeOrdersV2Binding
    - placeOrdersV2Operation
      - Request 1

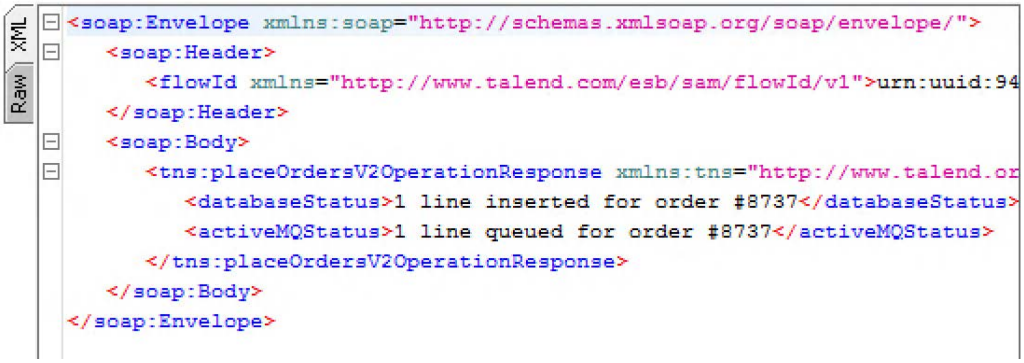
18. In the **Request 1** window, fill in the SOAP request parameters as follows:



```
<soapenv:Envelope xmlns:soapenv="http://sche
   <soapenv:Header/>
   <soapenv:Body>
      <ser:placeOrdersV2OperationRequest>
         <date>2016-07-26</date>
         <customerID>CUST00942</customerID>
         <productID>TAL_PF_BD</productID>
         <nbOrdered>1</nbOrdered>
      </ser:placeOrdersV2OperationRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

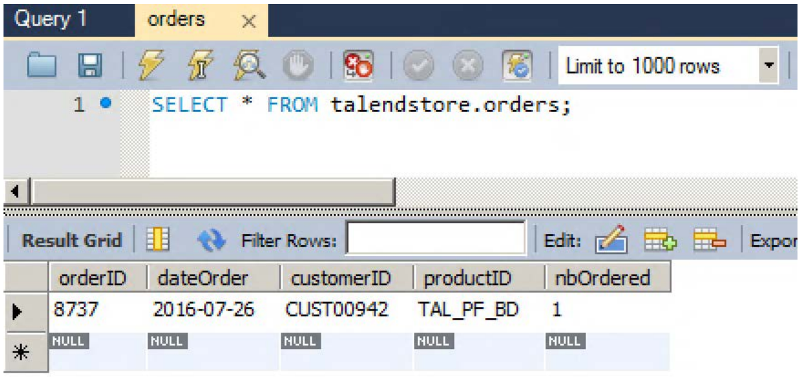
19. To send the request to the service, click the **Play** button.

20. The service should quickly answer with a message like this:



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Header>
      <flowId xmlns="http://www.talend.com/esb/sam/flowId/v1">urn:uuid:94
   </soap:Header>
   <soap:Body>
      <tns:placeOrdersV2OperationResponse xmlns:tns="http://www.talend.or
         <databaseStatus>1 line inserted for order #8737</databaseStatus>
         <activeMQStatus>1 line queued for order #8737</activeMQStatus>
      </tns:placeOrdersV2OperationResponse>
   </soap:Body>
</soap:Envelope>
```

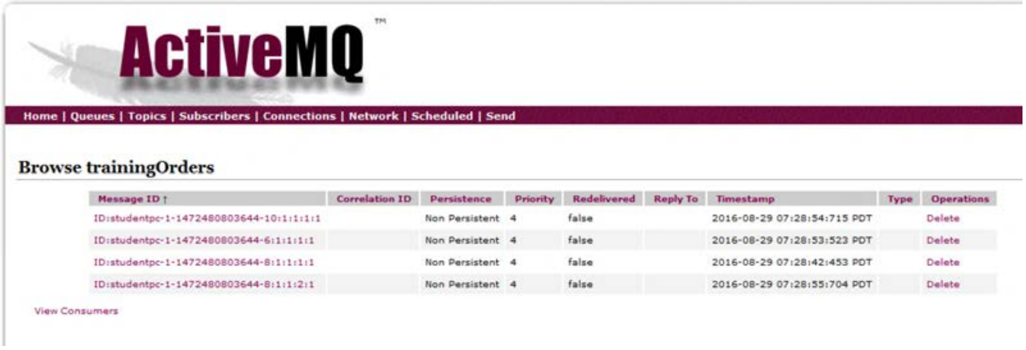
21. The service stores the order information in the MySQL database, in talendstore.orders. Using MySQL Workbench, make sure you can find the data in this table.



```
SELECT * FROM talendstore.orders;
```

orderID	dateOrder	customerID	productID	nbOrdered
8737	2016-07-26	CUST00942	TAL_PF_BD	1
NULL	NULL	NULL	NULL	NULL

22. The service also sends the order information as a JMS message to a queue called trainingOrders. On the ActiveMQ administration web console, look for your message.


ActiveMQ
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Browse trainingOrders

Message ID ↑	Correlation ID	Persistence	Priority	Redelivered	Reply To	Timestamp	Type	Operations
ID:studentpc-1-1472480803644-10:1:1:1:1		Non Persistent	4	false		2016-08-29 07:28:54:715 PDT		Delete
ID:studentpc-1-1472480803644-6:1:1:1:1		Non Persistent	4	false		2016-08-29 07:28:53:523 PDT		Delete
ID:studentpc-1-1472480803644-8:1:1:1:1		Non Persistent	4	false		2016-08-29 07:28:42:453 PDT		Delete
ID:studentpc-1-1472480803644-8:1:1:2:1		Non Persistent	4	false		2016-08-29 07:28:55:704 PDT		Delete

View Consumers

You have successfully deployed your first SOAP service with the ESB Conductor page and tested it. You can now go to the next section to publish your services and routes.

Publishing services and routes

Overview

Just as in the DI job life cycle, the ideal deployment follows best practices:

Developers develop and test ESB features (routes and data services)

Operators and administrators publish ESB features from SVN sources to Nexus

Operators and administrators deploy artifacts from Nexus to runtime instances

When needed, operators and administrators update or retire ESB features

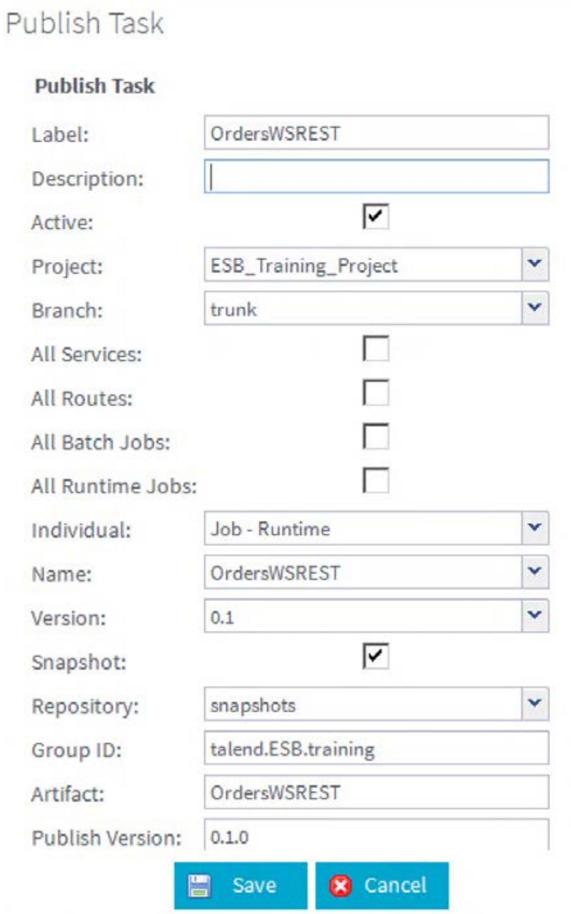
In this section, you will learn how to generate ESB features from TAC by using the Publisher module.

Publishing from TAC

As you may know from using the TAC in a DI context, you can use the Publisher module to generate compiled artifacts from SVN-stored sources. The Publisher engine, called CommandLine, can generate DI Jobs, services, and even routes.

A REST service and a route are stored on your VM SVN server. This section shows how to publish and deploy these ESB features.

1. Connect to the TAC as the operator user (*operator@company.com/operator*).
2. In the TAC menu, under **Conductor**, click **Publisher**.
3. Click **Add**.
4. Fill in the **Publish Task** panel as follows:
   Label: *OrdersWSREST*
   Project: *ESB_Training_Project*
   Branch: *trunk*
   Individual: *Job - Runtime*
   Name: *OrdersWSREST*
   Version: *0.1*
   Repository: *snapshots*
   Group ID: *talend.ESB.training*



Publish Task

Publish Task

Label:	OrdersWSREST
Description:	
Active:	☑
Project:	ESB_Training_Project
Branch:	trunk
All Services:	□
All Routes:	□
All Batch Jobs:	□
All Runtime Jobs:	□
Individual:	Job - Runtime
Name:	OrdersWSREST
Version:	0.1
Snapshot:	☑
Repository:	snapshots
Group ID:	talend.ESB.training
Artifact:	OrdersWSREST
Publish Version:	0.1.0

It may seem odd that you specify Job - Runtime to designate a REST service. Remember that from a developer's perspective, REST services are DI Jobs that use runtime-specific components (the REST consumer and producer endpoints), whereas a SOAP service is built like a service package around a WSDL contract and a Job.

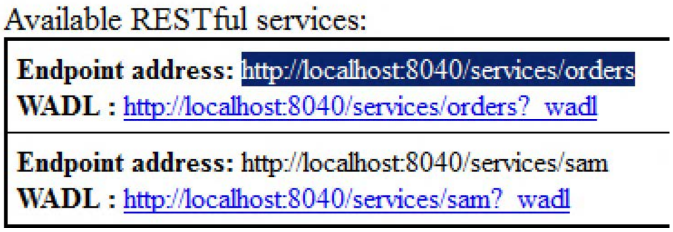
5. Click **Save**.
6. Click the **Publish** button to ask the CommandLine to compile the service and publish it on Nexus. Wait a few minutes for the task to be completed.

Deploying a TAC-published artifact

1. Still logged into TAC as the operator user, on the menu, click **ESB Conductor**.
2. To add a new task, click **Add > Task**. Unlike with the **Job Conductor**, there is no specific way to add a new task from a Nexus artifact, as the only valid way to deploy an ESB feature through TAC is by using Nexus.
3. Also, there is no specific interface for TAC-published artifacts. Creating your new task is the same exact process as in the previous section. In the Conductor Task **Label** field, enter *OrdersWSREST*.
4. Click the **Select Feature** button.
5. When publishing, you set the Group ID of our service to *talend.ESB.training* in the Snapshots repository. Navigate to **snapshots > talend > ESB > training > OrdersWSREST-feature > 0.1.0-SNAPSHOT > OrdersWSREST-feature-0.1.0-YYYMMDD-HHmmSS-1.xml**, then click **OK**.
6. Set the **Type** field to *Service*.
7. Set the **Server** to *ESB runtime 1*.
8. To create the task, click **Save**.
9. Click **Deploy**.

Testing your REST web service

1. In a web browser, go to **http://localhost:8040/services**.
2. Locate the newly deployed REST service **orders** and copy its endpoint URL.

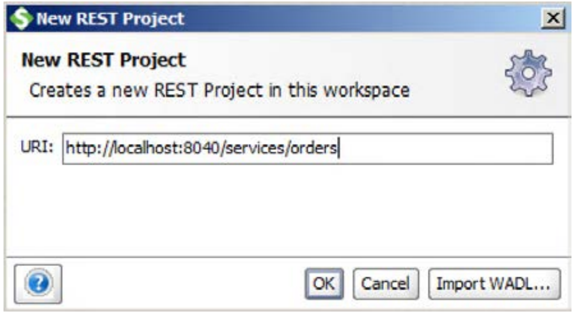


Available RESTful services:

Endpoint address: http://localhost:8040/services/orders
WADL : http://localhost:8040/services/orders?_wadl

Endpoint address: http://localhost:8040/services/sam
WADL : http://localhost:8040/services/sam?_wadl

3. In SoapUI, right-click **Projects.** On the contextual menu, select **New REST Project**.
4. In the **New REST Project** window, in the **URI** field, paste the endpoint URL and click **OK**.

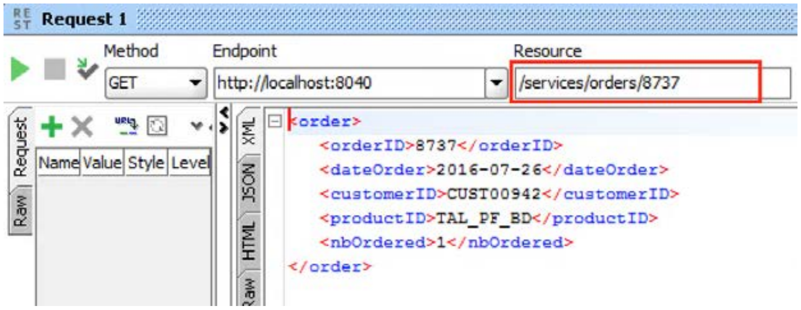


New REST Project

New REST Project
Creates a new REST Project in this workspace

URI: http://localhost:8040/services/orders

5. A new Request window opens.
   The REST service you published and deployed can return the details for any order recorded in the MySQL database. To access data for an order, the service takes one parameter. This parameter is passed via the URL. At the end of the service resource, type any order ID that was returned by the SOAP service in the previous section.



Request 1

Method: GET
Endpoint: http://localhost:8040
Resource: /services/orders/8737

```
<order>
    <orderID>8737</orderID>
    <dateOrder>2016-07-26</dateOrder>
    <customerID>CUST00942</customerID>
    <productID>TAL_PF_BD</productID>
    <nbOrdered>1</nbOrdered>
</order>
```

   If you don't remember an order ID, run the SOAP service placeOrdersV2 or search through the database with the MySQL Workbench client.
   After you add a valid order ID at the end of the resource (for example, */services/orders/8737* as in the screenshot), press the **Play** button.

Challenge

The ESB_Training_Project also has a route called readActiveMQData. This route catches any message sent to the ActiveMQ queue trainingQueue.

Publish and deploy it. Once deployed, it should consume any order message stored in the queue.

Next step

You have almost completed this lesson. Continue to the Wrap-up section for a review of the concepts we covered.

Wrap-up

This lesson covered the basic knowledge required to deploy ESB features manually or from TAC.

You learned that if necessary, you can deploy exported packages by dropping them in the runtime deploy folder. You also learned how to publish and deploy services via Nexus and TAC. Finally, you confirmed that your services and route were successfully deployed using SoapUI and the ActiveMQ administration console.

The next chapter shows how to monitor your services and test them using tools other than SoapUI.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let’s continue >**.

LESSON 4

Service Locator and Service Activity Monitoring

This chapter discusses the following.

Lab Overview

Monitoring Talend ESB

The Service Activity Monitoring (SAM) module allows you to log and monitor service requests and responses. Typical use cases are usage statistics and fault monitoring. This module has two parts: an agent (sam-agent) and a monitoring server (sam-server). The agent creates events from requests and replies on the service consumer and the provider side. To not disturb the normal messaging flow, the events are collected locally and periodically sent to the monitoring server. The monitoring server receives events from the agent, filters them (optional), and stores them in a database.

The Service Locator (SL) module provides service consumers with a mechanism to register, and also discover service endpoints on the runtime, thereby keeping consumers from knowing the physical location of the endpoint. Talend ESB uses Apache Zookeeper as its service locator server.

Note: While SL and SAM are in both Talend Open Studio for ESB and commercial Studio, the associated web-based user interfaces are part of Talend Administration Center, so they are only available in the commercial solution. These modules are licensed by Talend and require a license file to be installed.

Talend LogServer, with which you already familiar, is also available for all ESB modules with specific dashboards.

This chapter shows you how to use the SL and SAM modules on your SOAP and REST services.

Objectives

After completing this lesson, you will be able to:

Start the SAM and SL modules

Access the SAM and SL web UIs from TAC

Connect a runtime instance to Talend LogServer

Next Steps

First we will explore service activity with SAM.

Service Activity Monitoring

Overview

The Service Activity Monitoring (SAM) module allows administrators to keep track of all requests received and answered by web services.

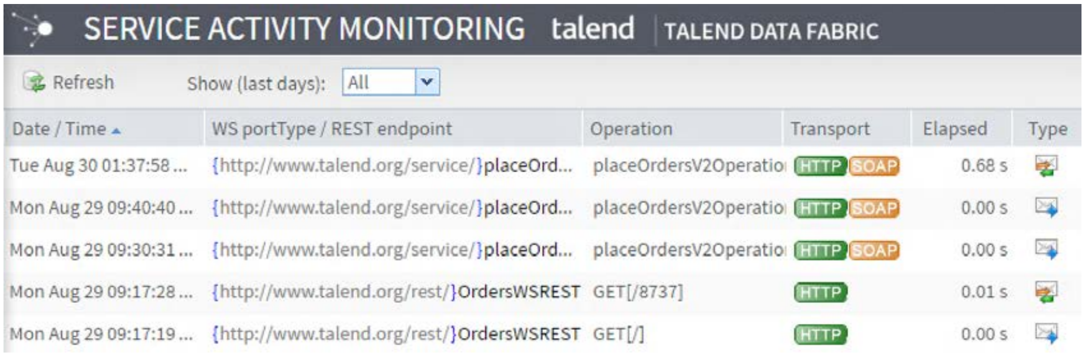
You can activate the SAM server feature in any runtime instance. You did this in the runtime-dedicated lab when you entered the command *tesb:start-sam*. Once the feature is activated, the runtime instance intercepts all incoming and outgoing messages for the services registered with SAM. When the runtime instance catches a message, it writes it into the SAM database.

By default, the services you deployed in the previous labs are configured to log into SAM.

This lab teaches you how to inspect the SAM logs in TAC.

Using SAM

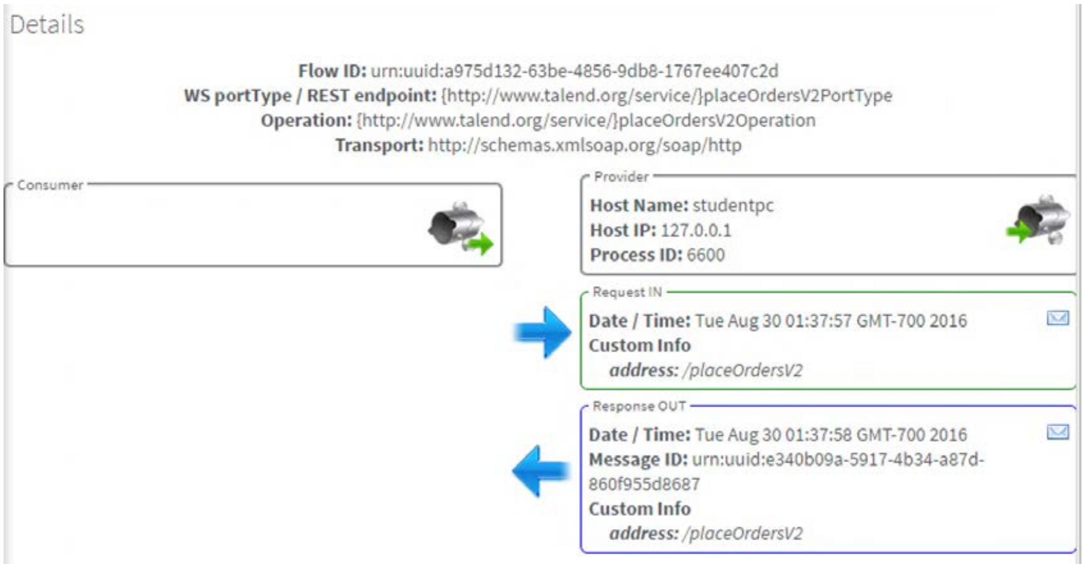
1. On the TAC menu, under **ESB Infrastructure**, click **Service Activity Monitoring**. This opens the Service Activity Monitoring web UI:



Date / Time ▲	WS portType / REST endpoint	Operation	Transport	Elapsed	Type
Tue Aug 30 01:37:58 ...	{http://www.talend.org/service/}placeOrd...	placeOrdersV2Operatio	HTTP SOAP	0.68 s	
Mon Aug 29 09:40:40 ...	{http://www.talend.org/service/}placeOrd...	placeOrdersV2Operatio	HTTP SOAP	0.00 s	
Mon Aug 29 09:30:31 ...	{http://www.talend.org/service/}placeOrd...	placeOrdersV2Operatio	HTTP SOAP	0.00 s	
Mon Aug 29 09:17:28 ...	{http://www.talend.org/rest/}OrdersWSREST	GET[/8737]	HTTP	0.01 s	
Mon Aug 29 09:17:19 ...	{http://www.talend.org/rest/}OrdersWSREST	GET[/]	HTTP	0.00 s	

Operations and requests sent to your web services are displayed here.

2. Click one of the requests to reveal more details about the service activity. The details appear below the list:



Details

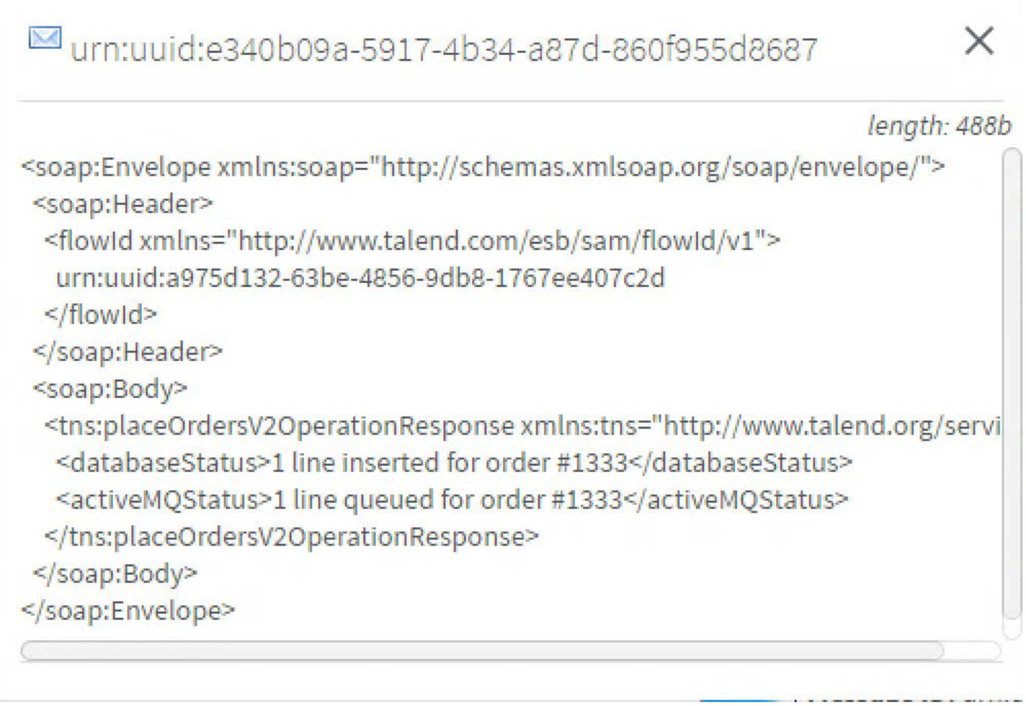
Flow ID: urn:uuid:a975d132-63be-4856-9db8-1767ee407c2d
WS portType / REST endpoint: {http://www.talend.org/service/}placeOrdersV2PortType
Operation: {http://www.talend.org/service/}placeOrdersV2Operation
Transport: http://schemas.xmlsoap.org/soap/http

Host Name: studentpc
Host IP: 127.0.0.1
Process ID: 6600

Date / Time: Tue Aug 30 01:37:57 GMT-700 2016
Custom Info
address: */placeOrdersV2*

Date / Time: Tue Aug 30 01:37:58 GMT-700 2016
Message ID: urn:uuid:e340b09a-5917-4b34-a87d-860f955d8687
Custom Info
address: */placeOrdersV2*

3. To see additional details, click the envelope in the upper right-corner of **Request OUT**:



urn:uuid:e340b09a-5917-4b34-a87d-860f955d8687

length: 488b

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
  <flowId xmlns="http://www.talend.com/esb/sam/flowId/v1">
   urn:uuid:a975d132-63be-4856-9db8-1767ee407c2d
  </flowId>
 </soap:Header>
 <soap:Body>
  <tns:placeOrdersV2OperationResponse xmlns:tns="http://www.talend.org/servi
   <databaseStatus>1 line inserted for order #1333</databaseStatus>
   <activeMQStatus>1 line queued for order #1333</activeMQStatus>
  </tns:placeOrdersV2OperationResponse>
 </soap:Body>
</soap:Envelope>
```

As expected, you can read the full response message.

4. In SoapUI, send a request to the service of your choice.
5. Back in TAC, click **Refresh** and confirm that you can see your latest service call.

You will now use the Service Locator module.

Service Locator

Overview

The Service Locator (SL) module allows administrators to keep track of services and know which are live and which are down.

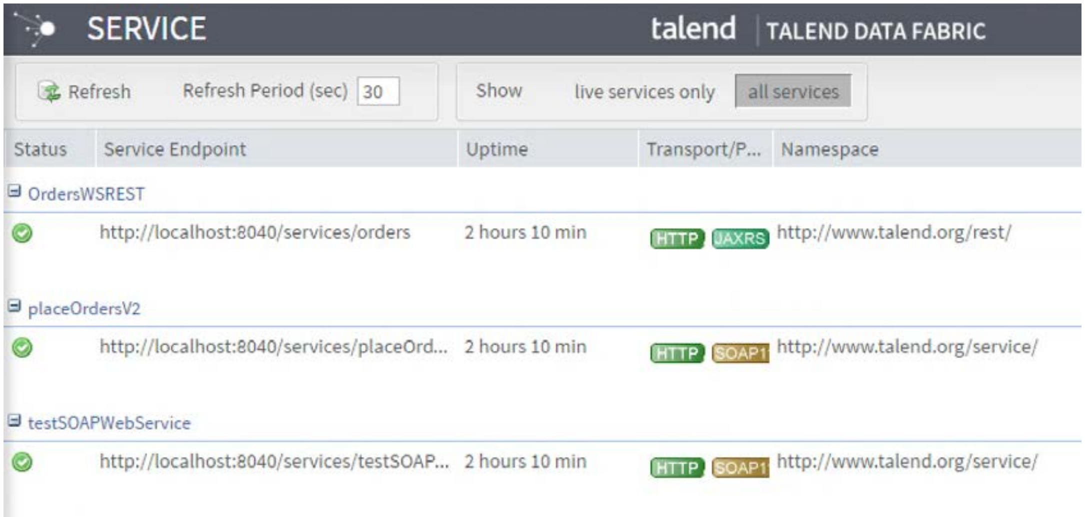
Any runtime instance can activate the SL feature. You did so in the runtime-dedicated lab when you entered the command *tesb:start-locator*. You can also run SL as a stand-alone program that manages the services of several runtimes.

By default, the services you deployed in the previous labs are configured to register with the locator.

This lab teaches you how to check the status of your services on the TAC Service Locator page.

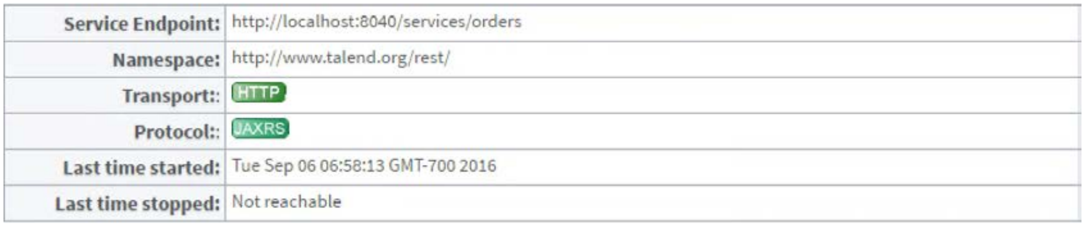
Using SL

1. In **TAC**, under **ESB Infrastructure**, click **Service Locator.**
2. The Service Locator page shows all registered services.



Status	Service Endpoint	Uptime	Transport/P...	Namespace
⊟ OrdersWSREST				
✓	http://localhost:8040/services/orders	2 hours 10 min	HTTP JAXRS	http://www.talend.org/rest/
⊟ placeOrdersV2				
✓	http://localhost:8040/services/placeOrd...	2 hours 10 min	HTTP SOAP1	http://www.talend.org/service/
⊟ testSOAPWebService				
✓	http://localhost:8040/services/testSOAP...	2 hours 10 min	HTTP SOAP1	http://www.talend.org/service/

3. To see additional details, click the **orders** REST service:

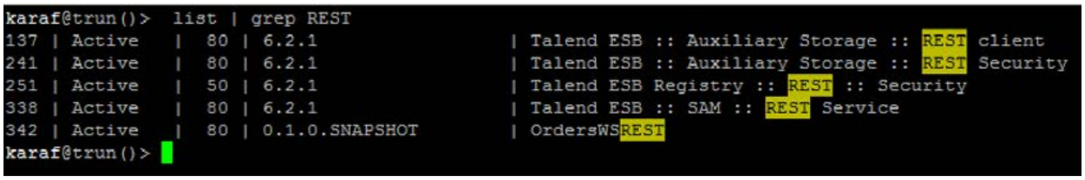


Service Endpoint:	http://localhost:8040/services/orders
Namespace:	http://www.talend.org/rest/
Transport::	HTTP
Protocol::	JAXRS
Last time started:	Tue Sep 06 06:58:13 GMT-700 2016
Last time stopped:	Not reachable

Stopping a service

The Service Locator page helps you identify services that have crashed. You can simulate a service fail by stopping a service in the runtime console:

1. From the **PuTTy** client, open a new connection to the runtime instance on *localhost*, port *8101*. Enter *karaf* as the user-name and password.
2. To get a list of all available REST services, enter:
   > list | grep REST

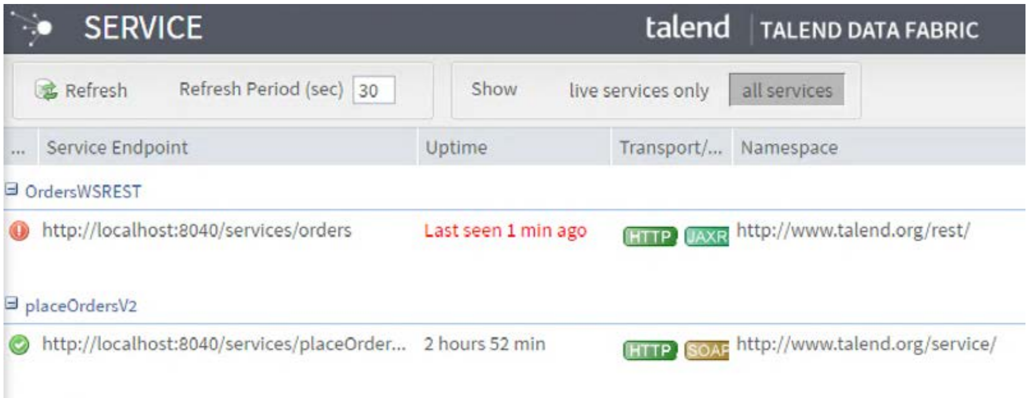


```
karaf@trun()>  list | grep REST
137 | Active    |  80 | 6.2.1                    | Talend ESB :: Auxiliary Storage :: REST client
241 | Active    |  80 | 6.2.1                    | Talend ESB :: Auxiliary Storage :: REST Security
251 | Active    |  50 | 6.2.1                    | Talend ESB Registry :: REST :: Security
338 | Active    |  80 | 6.2.1                    | Talend ESB :: SAM :: REST Service
342 | Active    |  80 | 0.1.0.SNAPSHOT           | OrdersWSREST
karaf@trun()>
```

Locate the REST service (its complete name is OrdersWSREST).

3. To stop the **OrdersWSREST** service, run
> stop OrdersWSREST

4. Back in TAC, on the **Service Locator** page, look for the **orders** REST service. It should be flagged as failed with a red exclamation mark.



...	Service Endpoint	Uptime	Transport/...	Namespace
⊟	OrdersWSREST			
!	http://localhost:8040/services/orders	Last seen 1 min ago	HTTP JAXR	http://www.talend.org/rest/
⊟	placeOrdersV2			
✓	http://localhost:8040/services/placeOrder...	2 hours 52 min	HTTP SOAP	http://www.talend.org/service/

You can now move on to the last monitoring module: Talend LogServer for ESB.

Talend LogServer

Overview

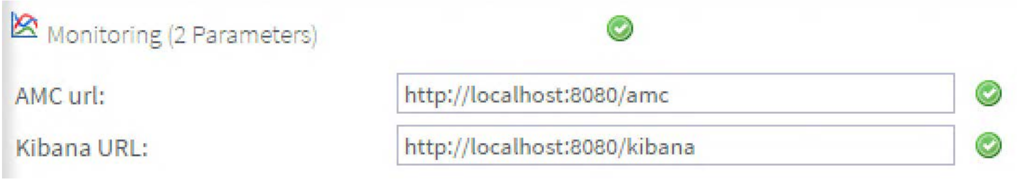
You already know about Talend LogServer and its dashboards from your experience with the Data Integration tools. Talend LogServer can also collect logs from your runtime instances. It even has an ESB dashboard in TAC.

Configuring Talend Runtime

1. From the **PuTTy** client, open a new connection to the runtime instance on *localhost*, port *8101*. Enter *karaf* as the username and password.
2. To activate the logging system, run the following command:
   >*tesb:start-el-default*
   This command installs and starts all necessary libraries to send runtime logs to Talend LogServer.
3. Restart the **Talend Runtime** service.

Configuring TAC

1. Log in to TAC as *administrator@company.com*
2. On the left menu, under **Settings**, click **Configuration**.
3. On the **Configuration** page, make sure that the **Monitoring** settings are configured with the host name *localhost*:



Monitoring (2 Parameters)

AMC url: http://localhost:8080/amc

Kibana URL: http://localhost:8080/kibana

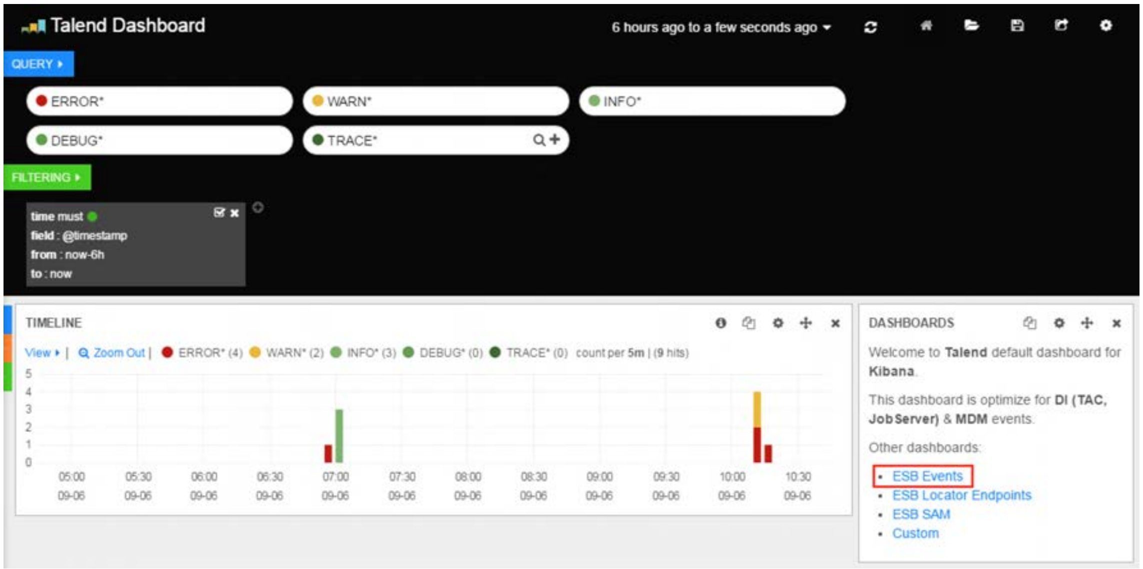
4. Log out of TAC.

Configuring Talend LogServer

1. In a file explorer window, navigate to **C:\Talend\6.2.0\logserv\elasticsearch-1.5.2\config**.
2. Right-click the file **elasticsearch.yml** and edit it with Notepad++.
3. Scroll down to the end of the file, in the **Security** settings. You must modify two parameters. The first is **http.cors.enabled**. This is set by default to false. When set to true, it allows an external application to send queries to Elasticsearch. Uncomment the line and set the parameter to *http.cors.enabled: true.*
4. The second parameter is **http.cors.allow-origin**, indicating which URL is allowed to run queries on Elasticsearch. The query agent we are using is Kibana, and it is embedded in TAC; the URL from which all queries sent to Elasticsearch originate is http://localhost:8080.
   Set the parameter as *http.cors.allow-origin: "http://localhost:8080"*
5. Save and close the file.
6. In the **Windows Services** window, restart **Talend Logserver 6.2.1**.

Exploring ESB logs in TAC

1. Now that the system is ready, before looking at the logs, send your services a few queries in **SoapUI**.
2. Log in to TAC as user *operator@company.com*
3. In the menu, under **Monitoring**, click **Logging**.
4. The default dashboard opens. However, this is not the ESB dashboard. A frame on the right side provides you with a link to the other dashboards. Click the **ESB Events** dashboard.



time must ●
field : @timestamp
from : now-6h
to : now

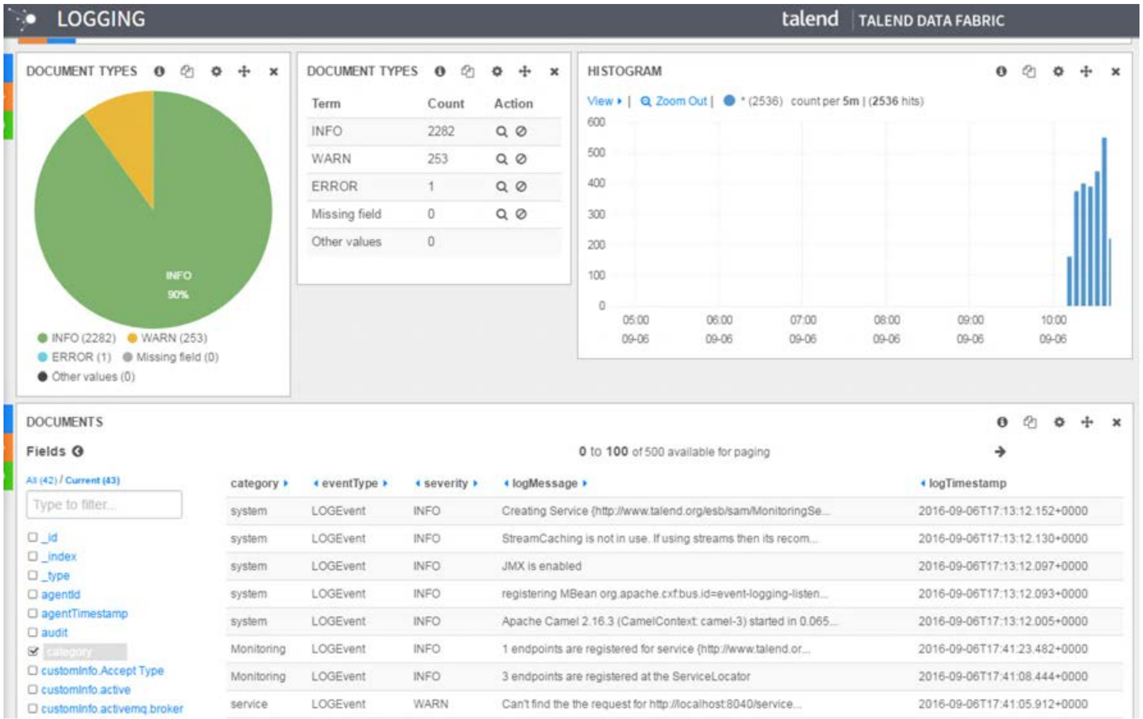
Welcome to **Talend** default dashboard for **Kibana**.

This dashboard is optimize for **DI (TAC, Job Server)** & **MDM** events.

Other dashboards:

- ESB Events
- ESB Locator Endpoints
- ESB SAM
- Custom

5. This dashboard filters data so you see only ESB runtime logs.



DOCUMENT TYPES

Term	Count	Action
INFO	2282	
WARN	253	
ERROR	1	
Missing field	0	
Other values	0	

□ _id
□ _index
□ _type
□ agentId
□ agentTimestamp
□ audit
☑ category
□ customInfo.Accept Type
□ customInfo.active
□ customInfo.activemq.broker

0 to 100 of 500 available for paging

category	eventType	severity	logMessage	logTimestamp
system	LOGEvent	INFO	Creating Service {http://www.talend.org/esb/sam/MonitoringSe...	2016-09-06T17:13:12.152+0000
system	LOGEvent	INFO	StreamCaching is not in use. If using streams then its recom...	2016-09-06T17:13:12.130+0000
system	LOGEvent	INFO	JMX is enabled	2016-09-06T17:13:12.097+0000
system	LOGEvent	INFO	registering MBean org.apache.cxf:bus.id=event-logging-listen...	2016-09-06T17:13:12.093+0000
system	LOGEvent	INFO	Apache Camel 2.16.3 (CamelContext: camel-3) started in 0.065...	2016-09-06T17:13:12.005+0000
Monitoring	LOGEvent	INFO	1 endpoints are registered for service {http://www.talend.or...	2016-09-06T17:41:23.482+0000
Monitoring	LOGEvent	INFO	3 endpoints are registered at the ServiceLocator	2016-09-06T17:41:08.444+0000
service	LOGEvent	WARN	Can't find the the request for http://localhost:8040/service...	2016-09-06T17:41:05.912+0000

Challenge

With your knowledge of Talend LogServer dashboards, use the Logging page and its search tools to find the latest service logs.

Next step

You have almost completed this lesson. Continue to the Wrap-up section for a review of the concepts we covered.

Wrap-up

In this lesson, you learned how to start and use the Service Activity Monitoring and Service Locator features. You accessed their web UIs in TAC to monitor the activity of your services.

You also learned how to start, configure, and use the Event Logging feature in Talend Runtime.

You can now access the logs using the Logging web UI in TAC. This UI has product-oriented preconfigured dashboards that classify logs related to Data Integration, ESB, or MDM events and make them easy to explore.

Further reading

For more information about the advanced capabilities of Talend ESB, refer to the *Talend Administration Center—User Guide*.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let’s continue >**.

This page intentionally left blank to ensure new chapters
start on right (odd number) pages.