

AYI ACADEMY: TALEND

Material práctico



Módulo 11: SOAP Web Services

Objetivos

Luego de completar esta lección podrá:

- Crear un archivo WSDL con Talend Studio
- Desarrollar un web services de operaciones SOAP
- Desarrollar un Job que permita consumir un webservices SOAP
- Acceder a la base de datos con un web services

Ejercicio

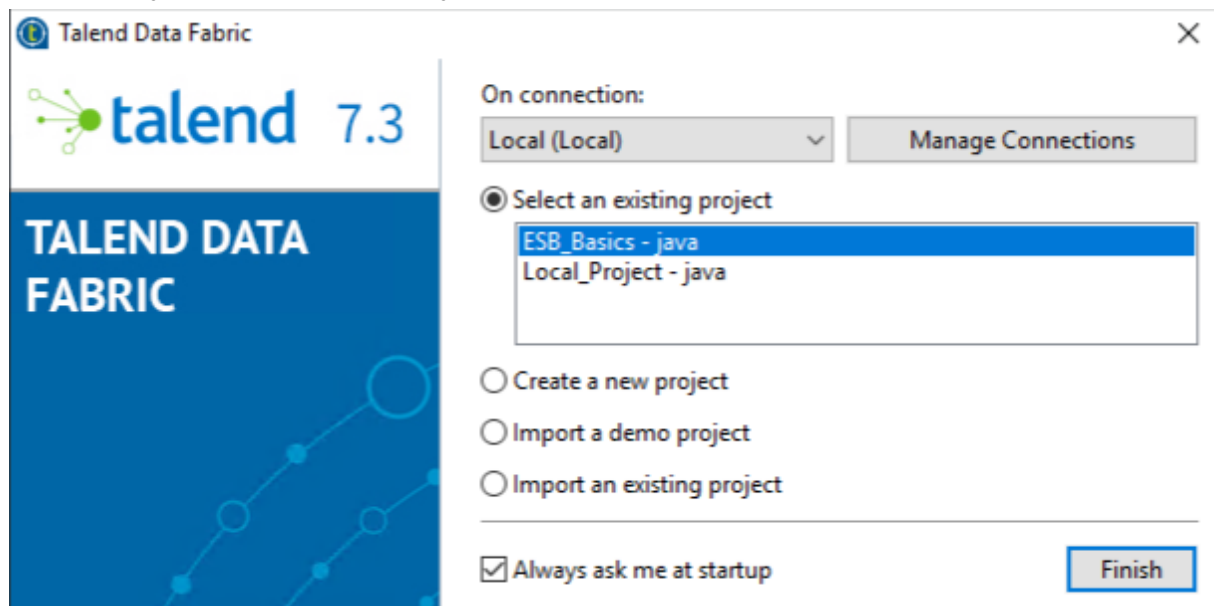
Los web services SOAP usan archivos WSDL. Un archivo WSDL provee un contrato entre el proveedor del servicio y el consumidor del servicio. Identifica las operaciones del servicio y los schemas de consulta y respuesta.

Este ejercicio muestra cómo crear un web services SOAP y cómo consumirlo. Diseña un web service SOAP en Talend Studio comenzando por el diseño del archivo WSDL. Una GUI específica lo ayuda a diseñar los objetos WSDL necesarios (por ejemplo, los service, binding, port type). Luego, cada operación del servicio puede ser vinculada a un DI Job. Diseñar las operaciones del web services—que es, la parte del servicio que procesa un request (invocación)—es simple: las operaciones son DI Jobs con componentes específicos de input y un output.

Le muestra cómo crear servicios web simples a más complejos y cómo consumirlos.

Abriendo el proyecto

Inicie su Talend Studio (para esta práctica la version 7.1)
Cree el proyecto ESB_Basics. Si ya existe, abralo.



Setting up the project

Asegúrese de que la perspectiva **Integration** esté seleccionada.



Tarea

Este ejercicio lo introducirá al proceso de crear un web services SOAP simple.

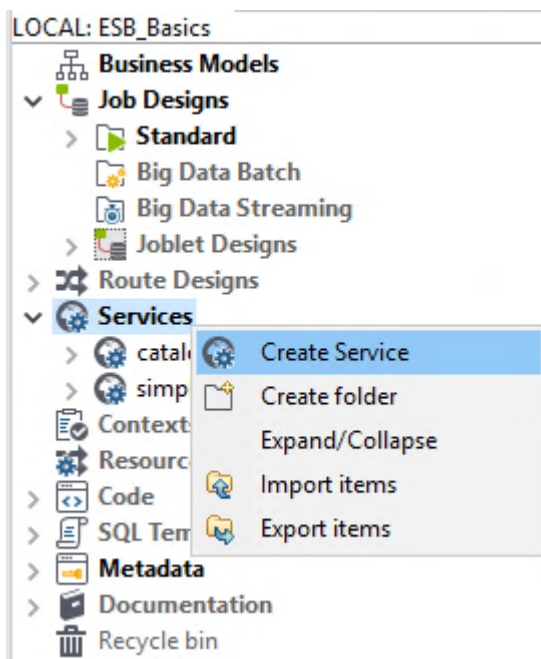
comenzaremos creando un nuevo archivo de contrato para el servicio, llamado archivo WSDL. En este archivo, definiremos el nombre del servicio, el nombre de sus operaciones y los elementos que esperamos para sus request y response (entrada y salida).

Luego asignaremos un DI Job a una operación.

Este primer web services es muy simple: tiene un string como parámetro request y response ese mismo string como mensaje de response.

Creando el servicio y el archivo WSDL

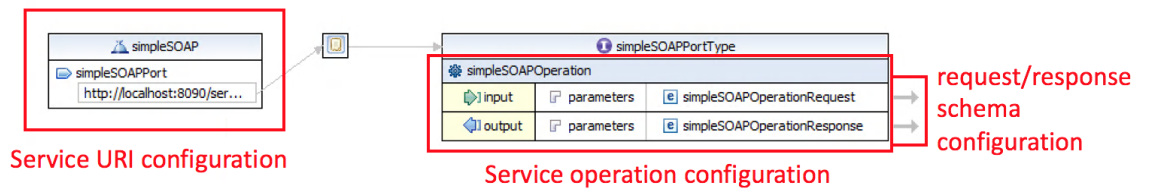
En el **Repository**, haga click derecho en **Services**. En el menú contextual, elija **Create Service**.



En la ventana del wizard **Services**, nombre su nuevo servicio como **simpleSOAP** y haga click en **Next**.

Tenes que elegir: usar un archivo WSDL existente o crear uno nuevo. Como no tenes un archivo, selecciona la opción **Create new WSDL** y haga click en **Finish**.

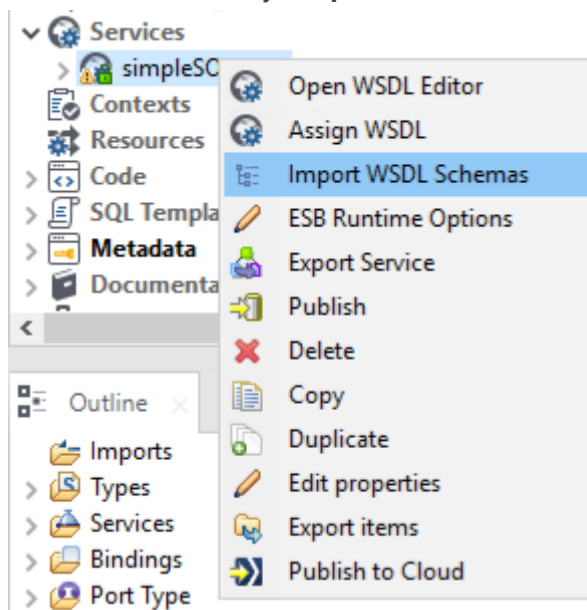
Se abre el editor de WSDL, mostrando varios elementos de configuración:



Puede editar cada elemento haciendo doble click sobre ellos. Por ahora, deja la configuración como es por default.

Para abrir el editor de schema, haga click en la flecha que está junto a **simpleSOAPOperationRequest**. Esta interfaz le permite editar, agregar y eliminar elementos en los schema de request y response. Como puede ver, el schema de request es muy simple. Está esperando un elemento String en un parámetro llamado "in". Cierre la ventana del editor de schemas. Abra el schema de response y observe que la respuesta esperada es un elemento String único denominado "out". Creará schemas más complejos más adelante en este capítulo. Guarde y cierre el editor WSDL.

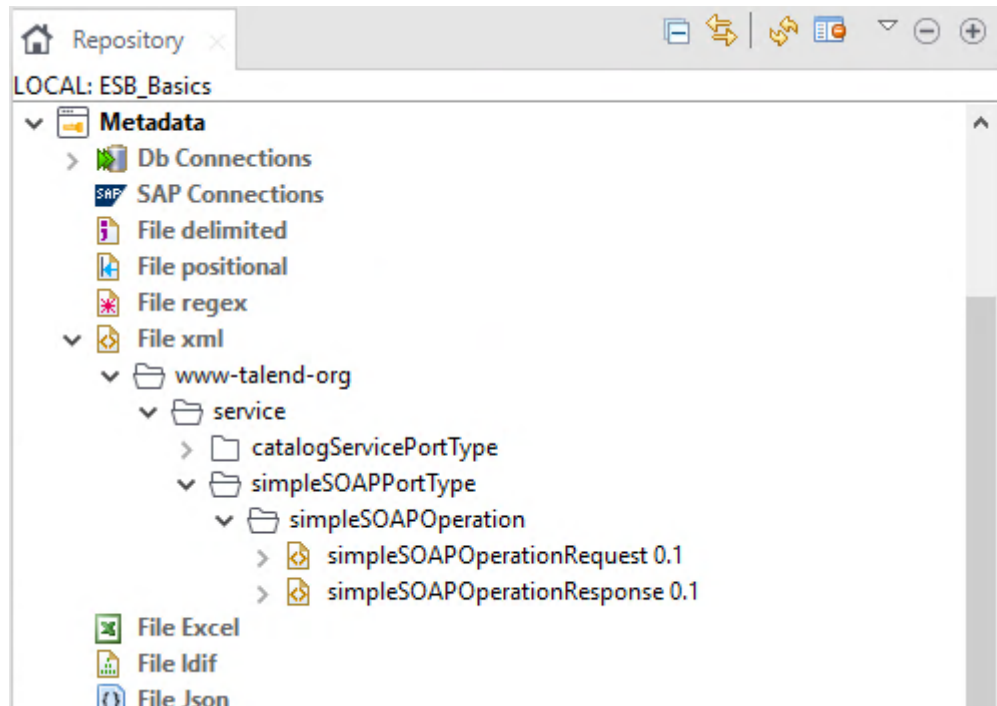
En el **Repository**, haga click derecho en su servicio **simpleSOAP**, y, en el menú contextual, elija **Import WSDL Schemas**.



Esta operación extrae los schemas de request y response de las operaciones del servicio y los almacena como metadatos XML en la sección Metadata del repositorio.

En el **Repository**, navegue por **Metadata > File XML > www-talend-org > service > simpleSOAPPortType > SimpleSOAPOperation**. Inspeccion los

schemas de **Request and Response**.



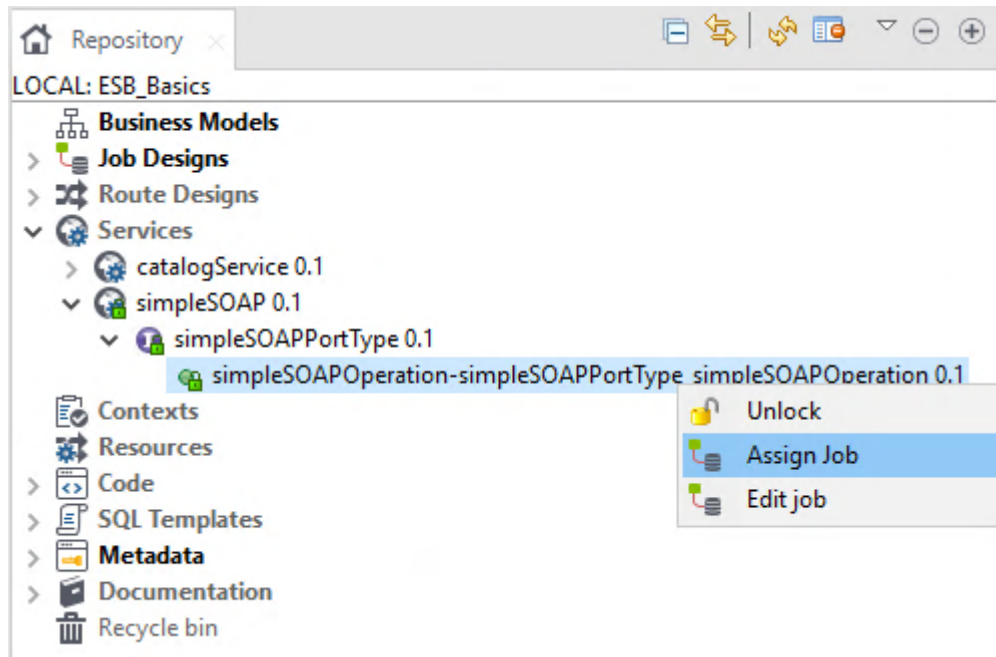
Asignando DI Job a una operación

Hasta ahora, ha creado el contrato WSDL llamado archivo WSDL y ha exportado su schema de request y response como metadata que puede usar fácilmente en DI Jobs.

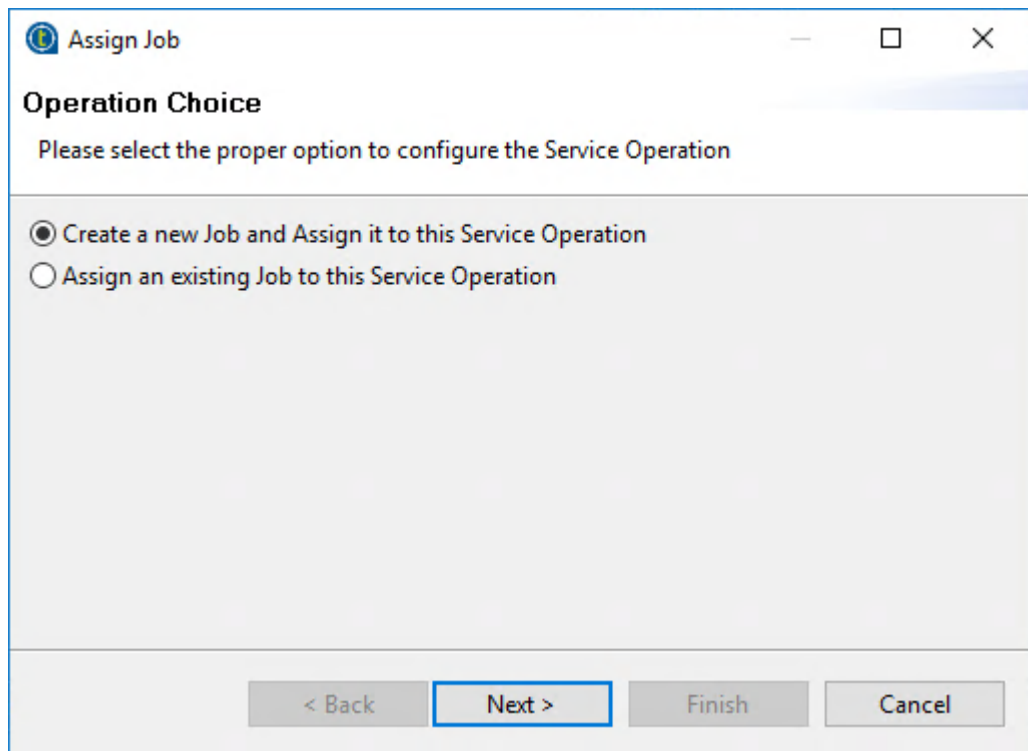
Ahora debe diseñar el Job que realmente haga algo cuando se llame al servicio web.

En el **Repository**, navegue por **Services > simpleSOAP 0.1 > simpleSOAPPortType 0.1** y haga click derecho en la operación

simpleSOAPOperation 0.1. En el menú, seleccione **Assign Job**.



En la ventana del wizard de **Assign Job**, elija **Create a new Job and Assign it to this Service Operation**, luego haga click en **Next**. Nota: en el caso de un servicio de múltiples operaciones, asigne un Trabajo a cada operación.



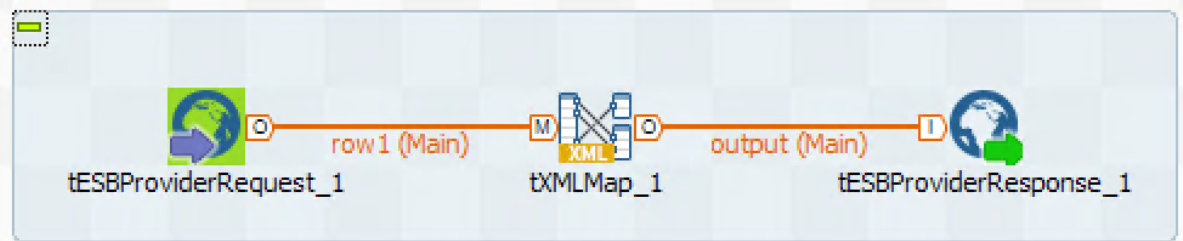
En la ventana del nuevo Job, deje el nombre por default y haga click en **Finish**. Esto crea y abre un nuevo Job, which que esta ligado a la operación de tu servicio. Se ejecuta cuando se llama a la operación del webservices. Cada nuevo Job es asignado a una operación del servicio y es creado con dos componentes por default: tESBProviderRequest y tESBProviderResponse.

- tESBProviderRequest es el encargado de recibir la solicitud (request) desde el webservices cliente y transmitirlo dentro del Job.

- tESBProviderResponse es el encargado de enviar el mensaje des respuesta desde el Job y enviarlo otra vez al web services cliente como respuesta.

En el nuevo Job, entre los componentes **tESBProviderRequest** y **tESBProviderResponse**, agregue un componente **tXMLMap**.

Dibuje un **Main** row desde **tESBProviderRequest_1** hacia **tXMLMap_1**, y un segundo main row, llamado **output**, desde **tXMLMap_1** hacia **tESBProviderResponse_1**. Cuando se le pregunte: **want to get the schema of the target component**, seleccione **Yes**.



Haga doble click en el componente **tXMLMap**.

Como se puede ver, entre los componentes **tESBProviderRequest** y **tESBProviderResponse** se puede ver el mismo schema: un único elemento Document-type llamado *payload*. El elemento payload contiene al mensaje SOAP request (en el componente tESBProviderRequest) y el mensaje SOAP response (en el componente tESBProviderResponse). Los mensajes SOAP son mensajes XML, que puedes usar:

- en un componente tXMLMap para extraer información del XML Document, procesarlo, y crear un nuevo XML document
- la XML metadata de los formatos de request y response, proveen el schema para extraer el request y construir el response.

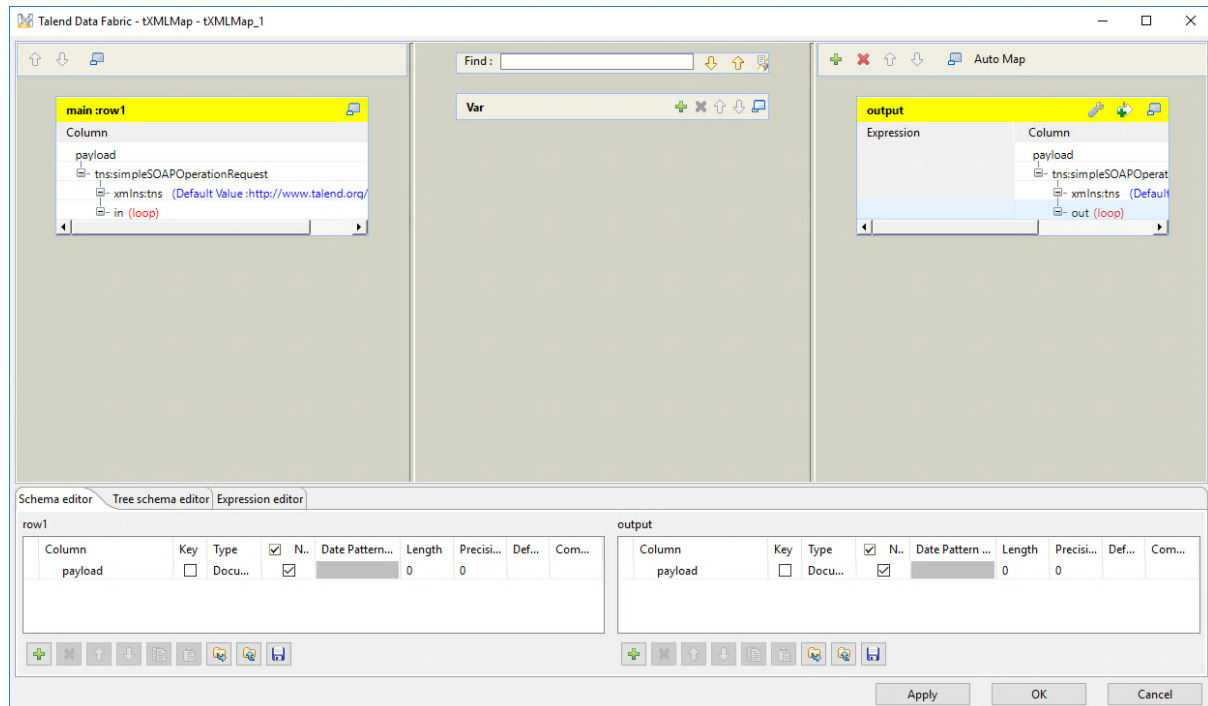
En la ventana del **tXMLMap window**, haga click derecho sobre el elemento **payload** que se encuentra en el flujo de entrada (**row1**). En el menu, elija **Import From Repository**.

Navegue por **Metadata > File XML > www-talend-org > service > simpleSOAPPortType > simpleSOAPOperation**, seleccione **simpleSOAPOperationRequest 0.1**, y haga click en **OK**. Esto importa el schema request de su operación dentro del elemento Document llamado payload.

Repita el proceso para el elemento de payload de salida: haga click derecho sobre el elemento de **payload** que se encuentra en el flujo de salida (**output**), y, en el menu, elija **Import From Repository**.

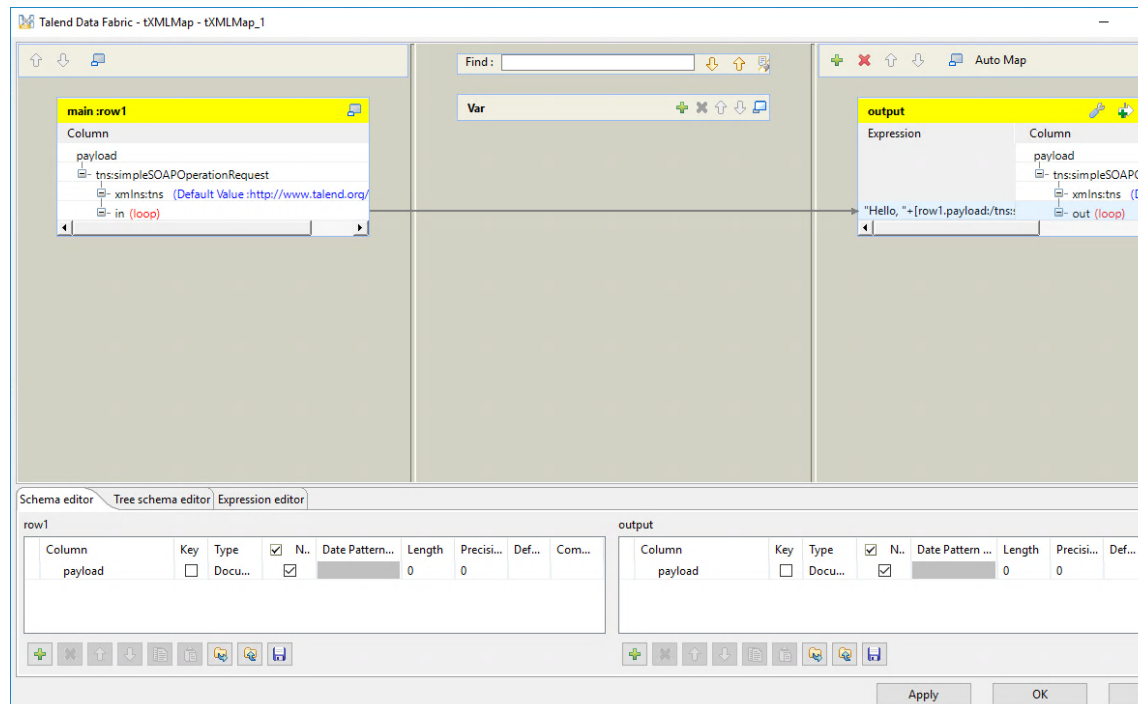
Navegue por **Metadata > File XML > www-talend-org > service > simpleSOAPPortType > simpleSOAPOperation**, seleccione **simpleSOAPOperationResponse 0.1**, y haga click en **OK**.

El editor de sur **tXMLMap** debería verse como la siguiente imagen:



En el campo **expression** de la columna **out** en el flujo **output**, ingrese:

- "Hello, "+[row1.payload:/tns:simpleSOAPOperationRequest/in]+"! I hope you're having fun ;)"



Cerrar el editor del tXMLMap editor, y haga click en **OK**.

Ejecutar el servicio

Tu simple servicio está listo: puede recibir un XML document que contiene un único string como parámetro y responde propagando el mensaje de respuesta que tb es un XML document. Estamos listos para ejecutar y testear el webwservice.


Save and run the Job.

Advanced settings	
Target Exec	
Memory Run	

```
Starting job simpleSOAPPartType_simpleSOAPOperation at 09:07
06/09/2016.

[statistics] connecting to socket on port 3472
[statistics] connected
Sep 06, 2016 9:07:24 AM
org.apache.cxf.wsdl.service.factory.ReflectionServiceFactoryBean
buildServiceFromWSDL
INFO: Creating Service {http://www.talend.org/service/}simpleSOAP
from WSDL:
C:/Talend/6.2.1/studio/workspace/ESB_BASICS/services/simpleSOAP_0.1.w
sdl
Sep 06, 2016 9:07:25 AM org.apache.cxf.endpoint.ServerImpl
initDestination
INFO: Setting the server's publish address to be
http://localhost:8090/services/simpleSOAP
2016-09-06 09:07:25.208:INFO:oejs.Server:jetty-8.1.14.v20131031
2016-09-06 09:07:25.301:INFO:oejs.AbstractConnector:Started
SelectChannelConnector@localhost:8090
web service [endpoint: http://localhost:8090/services/simpleSOAP]
published
```

Como puedes ver en la consola de log, el servicio esta expuesto en *http://localhost:8090/services/simpleSOAP*.

Para probar el servicio, deberá abrir la herramienta , the **Soap UI**. Esta herramienta simula un servicio cliente y ayuda a los desarrolladores a acceder y testear web services.

En **Soap UI**, crea un **New SOAP Project**.

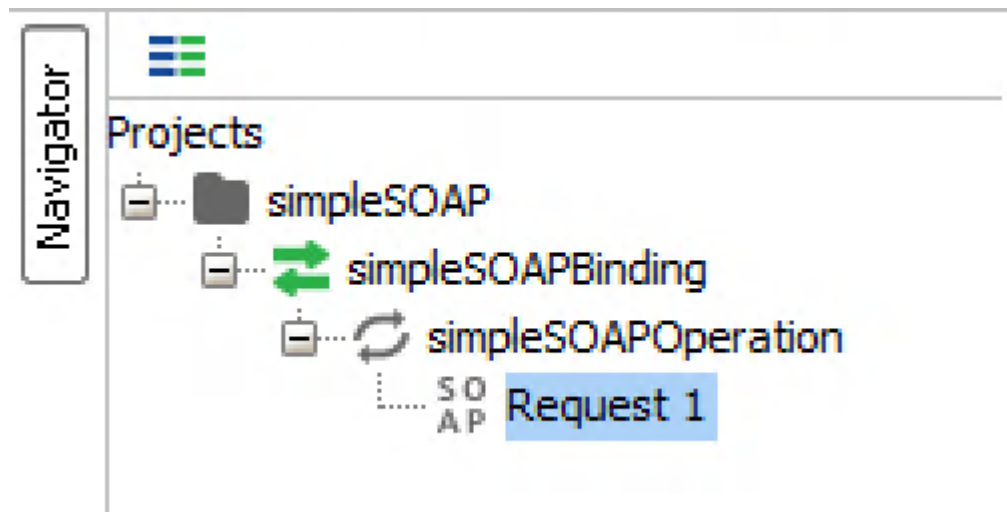
En la ventana de **New SOAP Project**, en la caja inicial **Initial WSDL** , ingrese:

- *http://localhost:8090/services/simpleSOAP?wsdl*

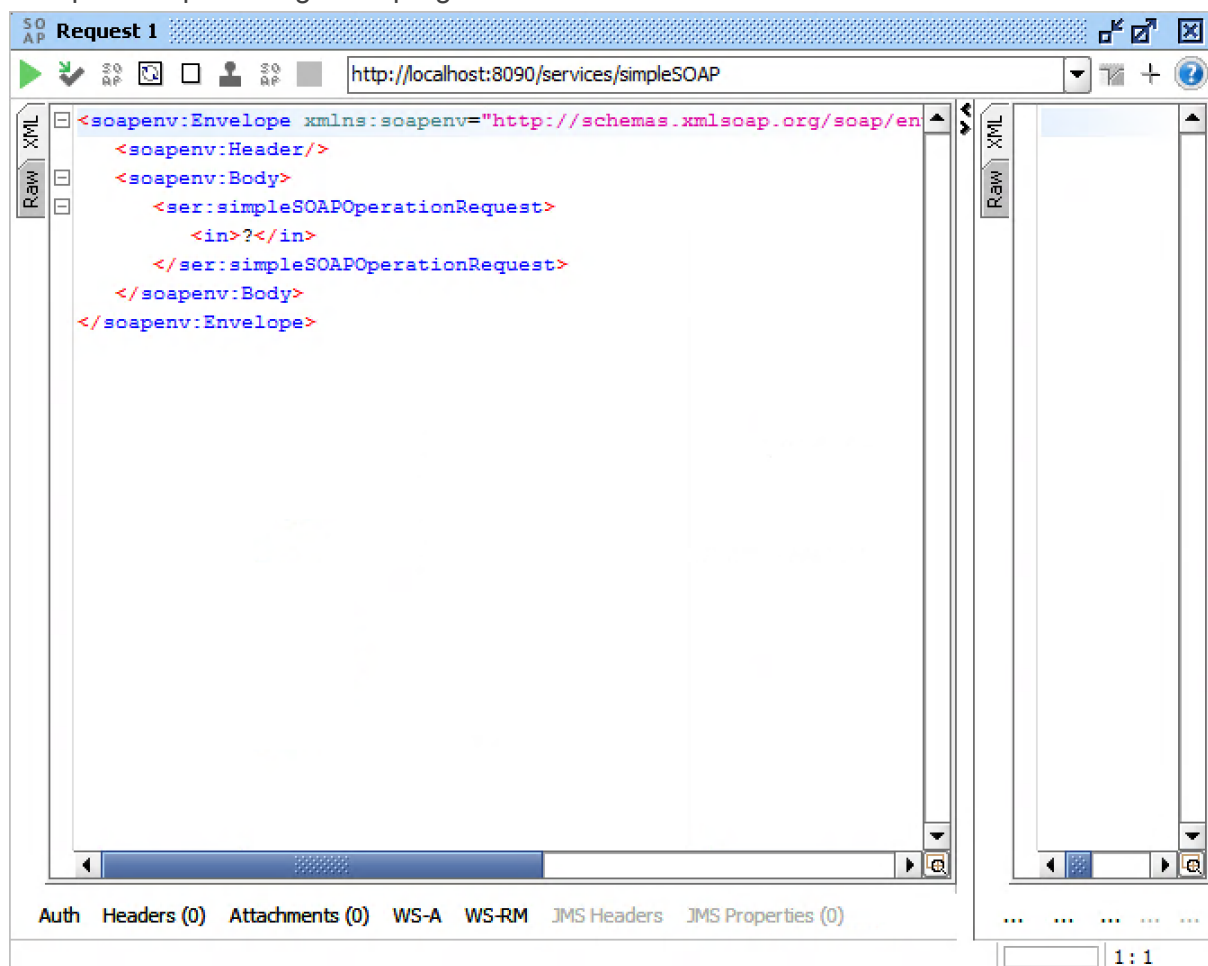
Haga click en **OK**.

Proporcionar la url del archivo WSDL permite al cliente SoapUI recuperar todos los detalles del servicio que está corriendo en el Talend Job. Podes encontrar la url del archivo WSDL agregando *?wsdl* al endpoint del servicio.

En el nuevo proyecto, navegue **simpleSOAP** > **simpleSOAPBinding** > **simpleSOAPOperation** > **Request 1**, y haga doble click en **Request 1**.



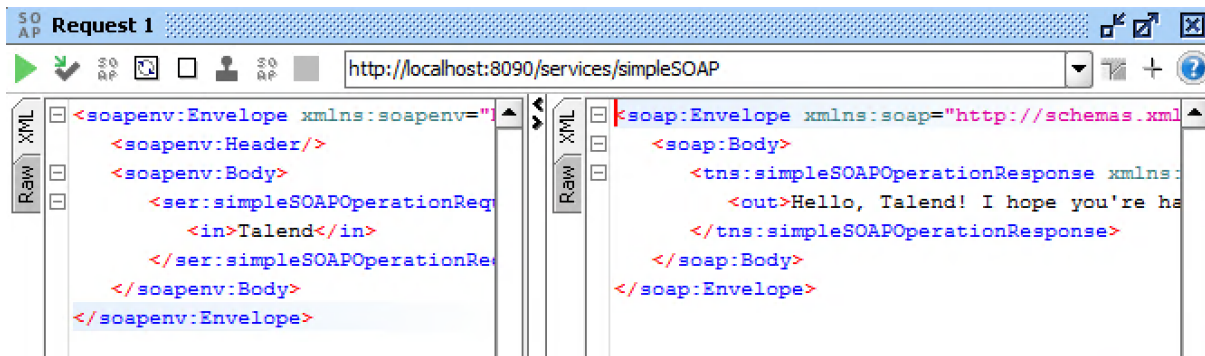
Esto abre una nueva ventana **Request** que contiene un mensaje request pre construido que puedes enviar a tu webservice. El valor de los parámetros es reemplazado por un signo de preguntas.



En la venta de **Request 1**, en el tag del parametro `in`, reemplace el signo de pregunta por su primer nombre.

Envíe la solicitud (request) al web service haciendo click en el botón **Play** verde.

En el panel de la derecha, el web services debería enviar la respuesta (response)



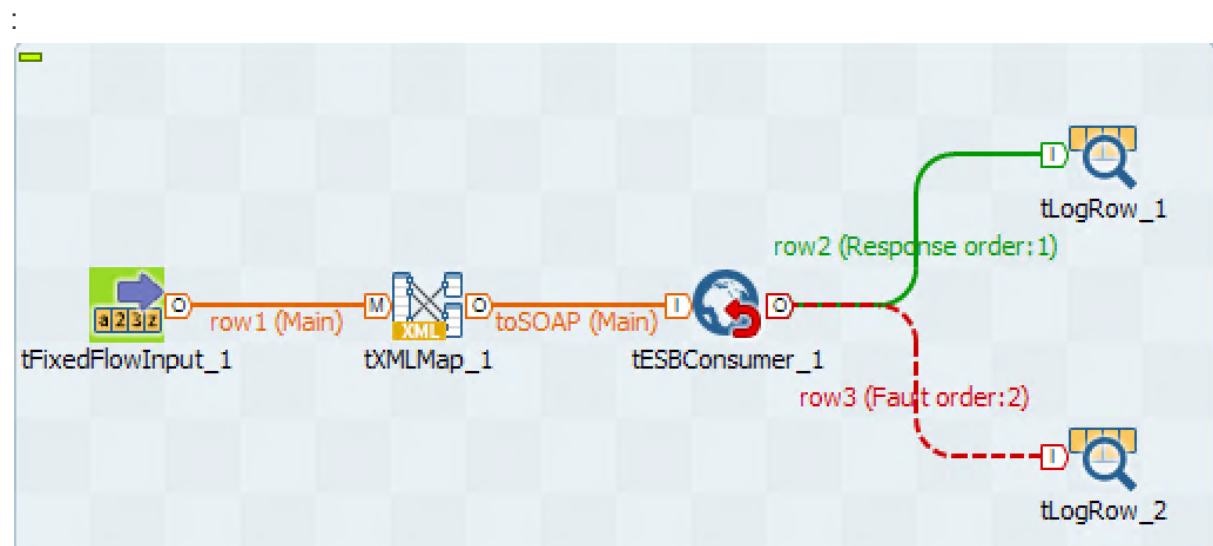
Deje el Job de su servicio corriendo. Continúe con el siguiente ejercicio donde crearemos un DI Job que consuma ese servicio.

Usando el componente tESBConsumer

Cree un nuevo Job y pónle por nombre *simpleSOAPConsumer*.

Agregue los siguientes componentes en su diseño:

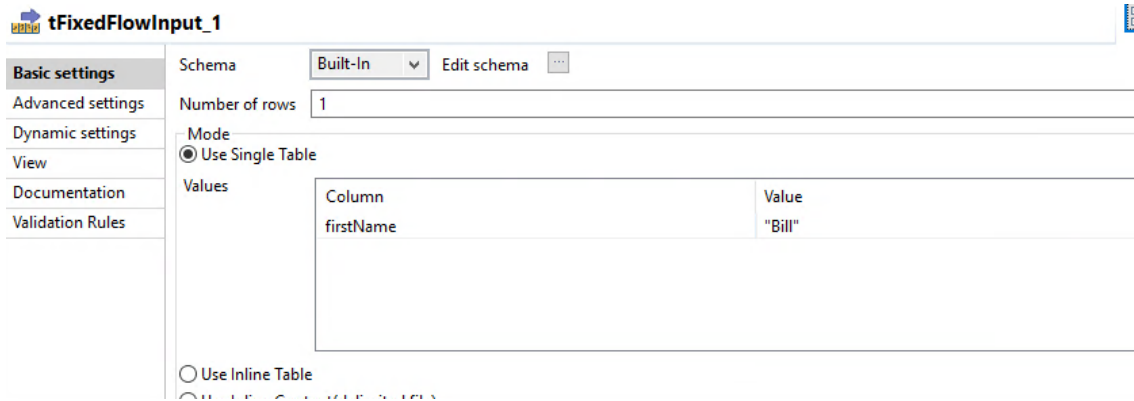
- tFixedFlowInput** (genera una fila de entrada de datos fijos)
- tXMLMap** (transforma el flujo de dato al formato XML SOAP request)
- tESBConsumer** (consume un web service)
- tLogRow** (muestra el flujo de datos en consola); agregue dos de este componente. Unalos como se ven en la siguiente imagen.



Note que hay dos flujos de datos de salida diferentes para el componente **tESBConsumer**: **Response**, usado para llevar la respuesta del web service, y **Fault**, en donde podras recibir mensajes de error del web service.

Configure el componente **tFixedFlowInput**:

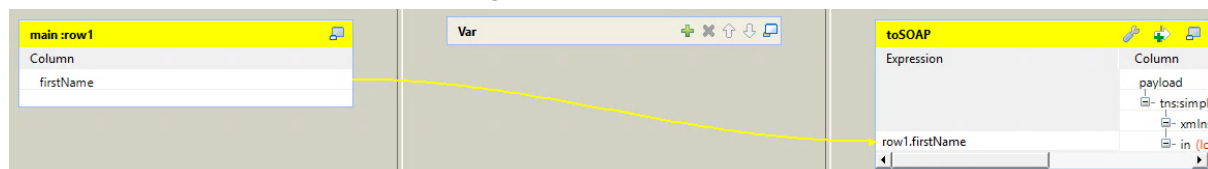
- Haga click en el botón de **Edit schema** y cree una columna llamada *firstName* de tipo *String* y haga click en **OK**. Si le pregunta si quiere propagar los cambios seleccione **Yes**.
- En **Mode**, seleccione **Use Single Table**.
- Para setear la columna **firstName**, en el área de **Values**, en la columna **Value**, ingrese *"Bill"* (o algún otro nombre).



Doble click en el componente **tXMLMap**. Este componente es usado para transformar la columna **firstName** en un mensaje XML que tu web services pueda entender.

- En el flujo **output**, click derecho sobre **payload**.
- En el menú, elija **Import From Repository**.
- Navegue por **Metadata > File xml > www-talend-org > service > simpleSOAPPortType > SimpleSOAPOperation**, y seleccione **simpleSOAPOperationRequest 0.1**.
- Haga click en **OK**.

Para cerrar el editor de tXMLMap, haga click en **OK**.

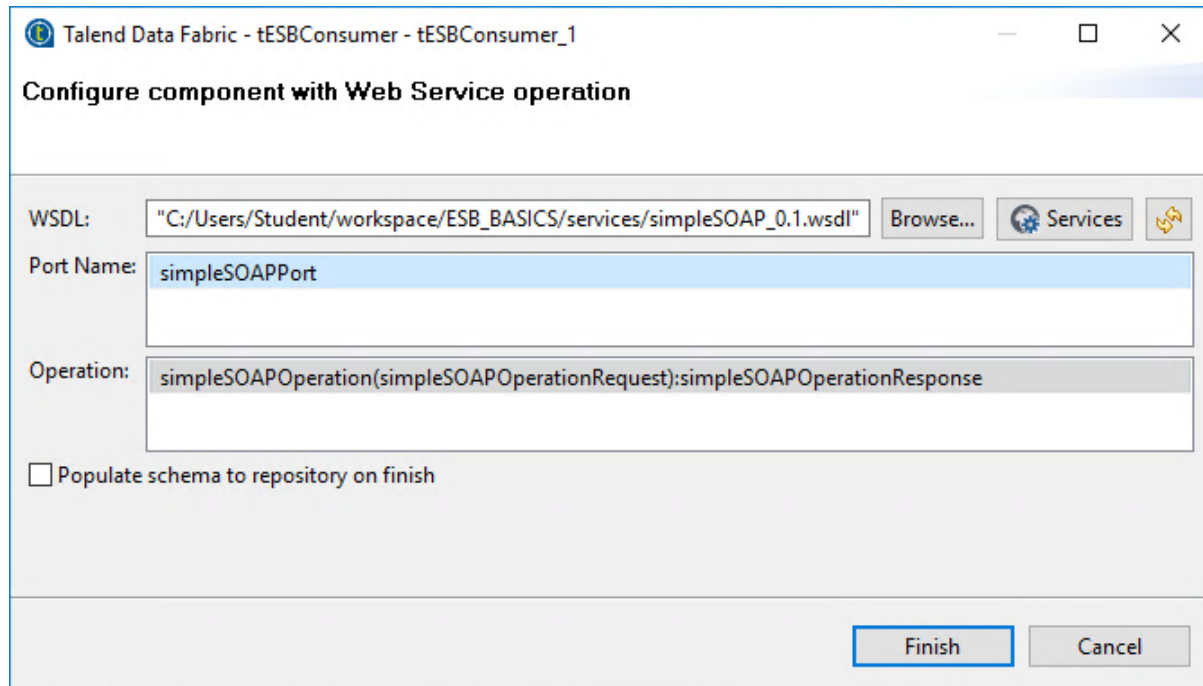


Doble click en el componente **tESBConsumer**. Esto abre el wizard **tESBConsumer**. Debe proporcionar el archivo WSDL para que el componente recupere la información del contrato de servicio. Ingresa la URL en la caja de **WSDL**, o, si el servicio que quieres acceder y consumir ha sido creado con Talend Studio, click en el botón **Services**.

En la ventana **Repository Content**, seleccione el servicio **simpleSOAP** y haga click en **OK**.

Solo hay un puerto y una operación para su servicio, por lo que no necesita configurar ningún otro parámetro. Si su servicio tuviera más de un puerto u operación, deberá seleccionar el puerto y la operación a la que desea llamar.

Para cerrar el wizard **tESBConsumer**, haga click en **Finish**.



Talend Data Fabric - tESBConsumer - tESBConsumer_1

Configure component with Web Service operation

WSDL: "C:/Users/Student/workspace/ESB_BASICS/services/simpleSOAP_0.1.wsdl" Browse... Services

Port Name: simpleSOAPPort

Operation: simpleSOAPOperation(simpleSOAPOperationRequest):simpleSOAPOperationResponse

☐ Populate schema to repository on finish

Finish Cancel

Guarda los cambios y ejecuta el JOB. Esto envia el firstName que configuraste al web service y muestra en la consola la respuesta.

```
Starting job simpleSOAPconsumer at 00:23 07/09/2016.
[statistics] connecting to socket on port 3981
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<tns:simpleSOAPOperationResponse
xmlns:tns="http://www.talend.org/service/"><out>Hello, Bill! I hope
you're having fun ;)</out></tns:simpleSOAPOperationResponse>
[statistics] disconnected
Job simpleSOAPconsumer ended at 00:23 07/09/2016. [exit code=0]
```

No olvides detener la ejecución del Job

simpleSOAPPortType_simpleSOAPOperation Job una vez que ya no lo uses.

Tarea

Este ejercicio muestra como contruir un web service SOAP un poco más complejo con Talend Studio.

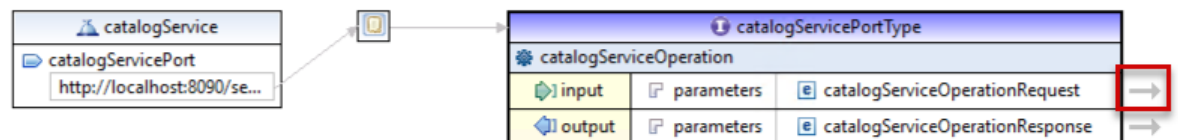
Este web service SOAP nos permitirá recuperar información acerca de productos en una base de datos. enviaremos un parametro product ID y recuperaremos en un mensaje XML el detalle de la información de ese producto desde la base de datos.

Creando un servicio SOAP

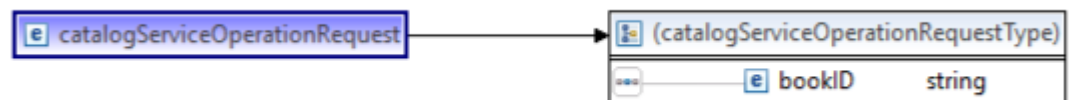
En el **Repository**, click derecho en **Services**, click en **Create Service**, y nombrelo como *catalogService*. Haga click en **Next**.

Elija crear un nuevo archivo WSDL para este servicio y luego haga click en **Finish**.

En el editor del WSDL, para editar el formato del request, haga click en la flecha a la derecha del parámetro input.



En la nueva ventana, haga doble click en el parametro por default **in** y renombrelo por *bookID*.



Guarde, luego cierre la ventana y vuelva al editor WSDL. Para editar el formato de la respuesta, haga click en la flecha que aparece a la derecha del parámetro output.

La respuesta envía todas las columnas de información sobre el libro que la base de datos puede devolver.

Haga doble click en el elemento **out** y renombrelo como *title*

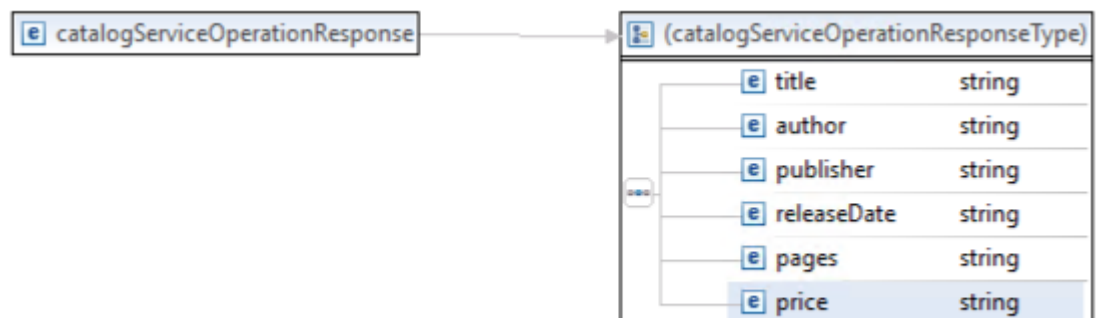
Para agregar columnas en este schema, haga click derecho en el elemento **title** y elija **Insert Element**. Elija **After** para insertar un nuevo elemento.

Renombre el nuevo elemento como *author* y deje su tipo como **string**.

Edite el modelo e inserte y renombre los siguientes elementos:

- *publisher*, string
- *releaseDate*, string
- *pages*, string
- *price*, string

Cuando finalice, guarde y cierre su schema.



En el editor del WSDL, guarde el servicio.

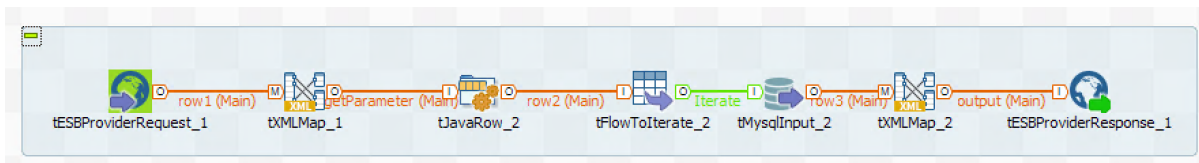
Para guardar el schema de su request y response como XML metadata, en el **Repository**, navegue en **Services**, click derecho en el elemento **catalogService** y elija **Import WSDL Schemas**.

Creando el JOb que se asigna a la operación del servicio

En el **Repository** expanda **Services**, expanda **catalogService** luego navegue hasta la operación **catalogServiceOperation**, click derecho, y elija **Assign Job**.

En el wizard, cree un nuevo Job y deje el nombre por default.

El Job detrás de este servicio se verá como en la siguiente imagen:



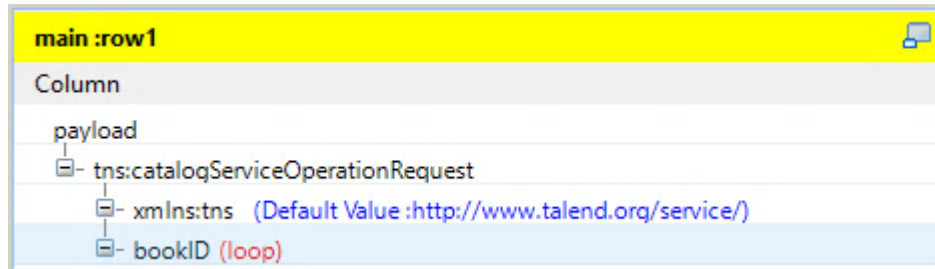
Miremos los componentes:

- **tESBProviderRequest** recibe la solicitud y la envía al flujo Main.
- **tXMLMap** lee el request XML y extrae el parametro bookID.
- **tJavaRow** almacena el valor de bookID en una variable de contexto.
- **tFlowToIterate** corta el flujo Main y nos permite conectar con el componente tDBInput.
- **tDBInput** ejecuta una query en la base de datos: recupera información acerca del libro que corresponde al ID que guardamos en la variable de contexto.
- **tXMLMap** toma la información recuperada de la base de datos y le da el formato al mensaje XML de respuesta.
- **tESBProviderResponse** envía el XML de respuesta al servicio web cliente.

Agregue un componente **tXMLMap** y unalo con el componente **tESBProviderRequest** usando un **Main** row.

Doble click en el componente **tXMLMap**. En la ventana del **tXMLMap**, click derecho en el elemento **payload**, y elija **Import From Repository**. Según lo que hizo en el primer ejercicio de este capítulo, importe el schema del request de la metadata de su servicio. La configuración de su tXMLMap debería

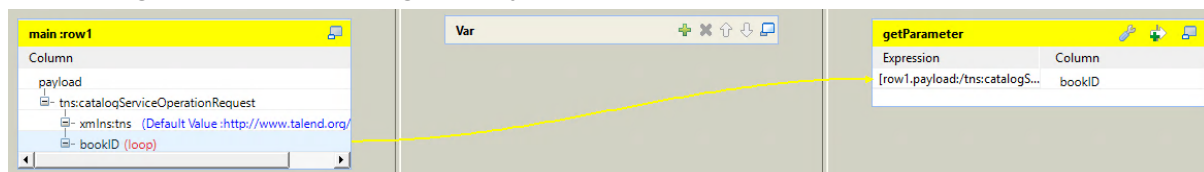
aparecer como la imagen a continuación.



Todavía en el editor de **tXMLMap**, haga click en **+** en el panel derecho para agregar una salida output de nombre *getParameter*.

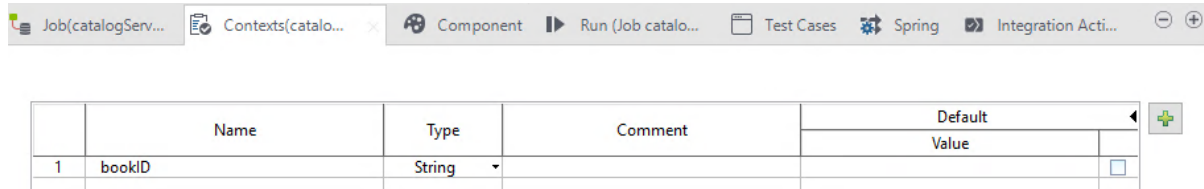
En el **Schema editor**, agregue una única columna de nombre **bookID** y tipo **String**.

Map desde el origen **bookID** en la columna **bookID** del output **getParameter** output, haga click en **OK** para guardar y cerrar el editor de **tXMLMap**.



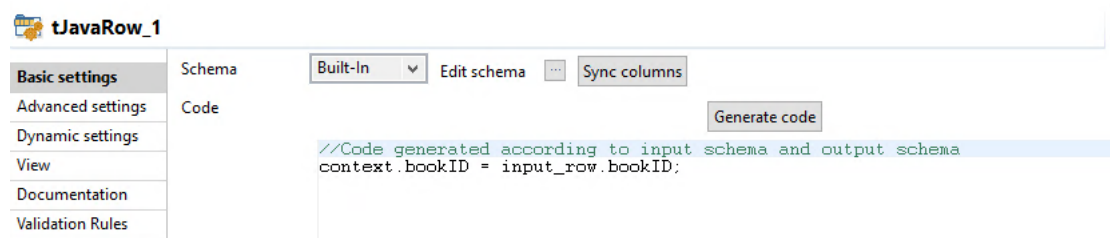
Agregue un componente **tJavaRow** a la derecha del componente **tXMLMap**. Click derecho en el componente **tXMLMap**, seleccione **Row > getParameter** output, y unalo al componente **tJavaRow**.

En la pestaña **Context**, agregue una nueva variable de contexto de tipo **String** y nómbrala *bookID*.



En la vista **Component** del componente **tJavaRow**, escriba la siguiente línea de código:

- `context.bookID = input_row.bookID;`



A la derecha agregue un componente **tFlowTolterate**. Dibuje un **Main row** desde el componente **tJavaRow** uniendo los dos componentes.

En el **Repository**, navegue por **Metadata > Db Connections >**

trainingDatabase 0.1 > Table Schemas. arrastre el schema **books** a la derecha del componente **tFlowTolterate**. Cuando le pregunte que componente usar, seleccione **tDBInput(MySQL)**.

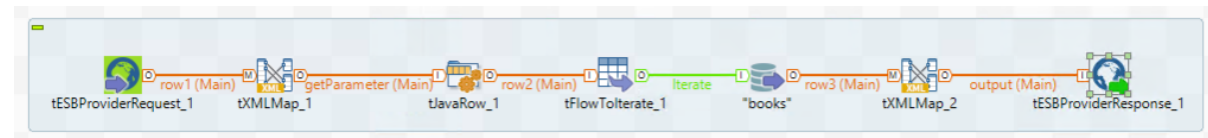
Desde el componente **tFlowTolterate**, dibuje un flujo **Iterate** para unirlo con el componente **tDBInput**.

En la vista **Component** del componente **tDBInput**, actualiza la query de manera que solo seleccione informacion de bookID que recibimos en el request. Puedes hacerlo agregando una condición WHERE a la query.

Actualiza la query de la siguiente manera:

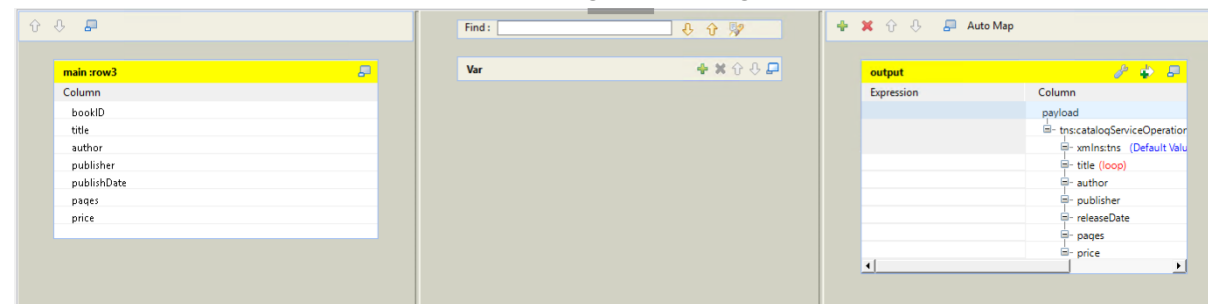
```
"SELECT `books`.`bookID`, `books`.`title`, `books`.`author`, `books`.`publisher`,
`books`.`publishDate`, `books`.`pages`, `books`.`price` FROM `books` where
`books`.`bookID` = '"+context.bookID+'"
```

agrega a la derecha un componente **tXMLMap**. Unelo con el componente **tDBInput** de manera que este quede como input, y el componente **tESBProviderResponse** como output. Nombre el output como *output*, y, cuando le pregunten si “you want to get the schema of the target component”, haga click en **Yes**. Su Job se verá como en la siguiente imagen.



Abra el componente **tXMLMap_2**. En el editor de **tXMLMap**, su flujo de entrada muestra las columnas recuperadas de la query SQL, y su flujo de salida muestra el elemento payload.

Haga click derecho en el elemento **payload** en el **output**, y elija **Import From Repository**, e importe de la su schema de respuesta de **catalogService**. Su editor de tXMLMap debería verse como la siguiente imagen:



Map each column from the **input** flow into its column in the **output** schema (or to save time, choose **Auto Map**). Be sure to map **publishDate** to **releaseDate**. To save and close your **tXMLMap**, click **OK**.

Ejecute y testee el servicio

Guarde y ejecute el Job.

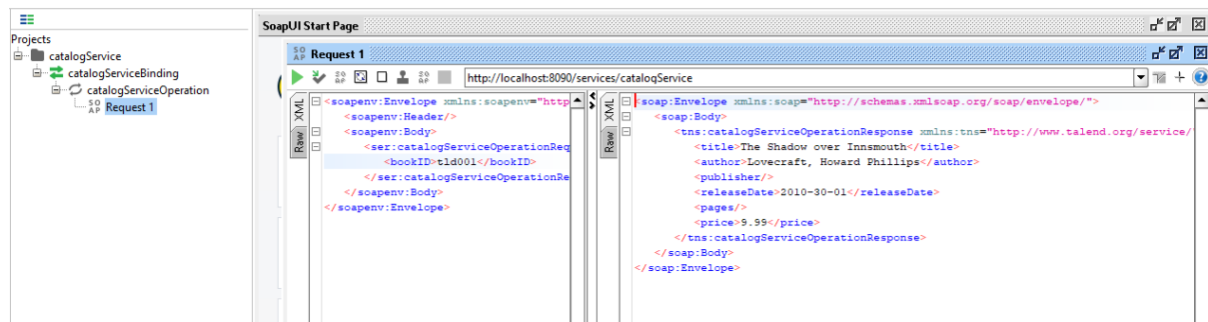
En **Soap UI**, cree un nuevo proyecto SOAP y nombrelo *catalogService*.

Proporcione al proyecto SOAP la url del WSDL,

http://localhost:8090/services/catalogService?wsdl

En la ventana **Request**, ingrese el valor del parámetro **bookID** como *tlid001* y haga click en el botón **Play** para enviar la solicitud al servicio.

La respuesta del servicio se puede ver en el panel derecho de la ventana Request.



No olvide de hacer click en Kill una vez que ya no use el servicio..

Resumen

In this module, you learned how to:

- Crear un archivo WSDL con Talend Studio
- Desarrollar un web service SOAP
- Desarrollar un Job que consume al web service SOAP
- Acceder a la base de datos a través de un web service SOAP