



SOFIA2: CONEXIÓN DESDE TALEND

Enero 2017

Versión 1



indra

1 INDICE

1	INDICE	2
1.1	REQUISITOS	3
1.2	OBJETIVOS Y ALCANCE DEL PRESENTE DOCUMENTO	3
2	INSTALACIÓN DE LOS COMPONENTES EN TALEND	4
2.1	DESCARGAR COMPONENTES DE GITHUB.....	4
2.2	CONFIGURAR TALEND PARA USAR COMPONENTES DE USUARIO	5
3	CONFIGURACIÓN PREVIA EN SOFIA2	6
3.1	DARSE DE ALTA EN LA INSTANCIA SOFIA2 INCLOUDLAB.....	6
3.2	ACCEDER A LA CONSOLA WEB DE SOFIA2 CLOUDLAB.....	7
3.3	BÚSQUEDA Y SUSCRIPCIÓN A ONTOLOGÍAS	7
3.4	CREACIÓN DE UN KP.....	9
4	CÓMO DESARROLLAR UN JOB TALEND PARA SOFIA2	11
4.1	JOB TALEND DE CONSULTA A SOFIA2	12
4.1.1	Configuración del componente tSofia2Input.....	13
4.1.2	Conversión de formato JSON a filas	14
4.1.3	Visualización de la salida	15
4.2	JOB TALEND DE INSERCIÓN EN SOFIA2	16
4.2.1	Generación de filas aleatorias.....	17
4.2.2	Paso a formato JSON.....	18
4.2.3	Configuración de tSofia2Output	20
4.3	JOB TALEND DE SUSCRIPCIÓN A UNA ONTOLOGÍA DE SOFIA2	21
4.3.1	Conexión reutilizable.....	22
4.3.2	Generación de filas aleatorias.....	23
4.3.3	Suscripción a la consulta	23
4.3.4	Lectura de objeto JSON	24

INTRODUCCIÓN

1.1 Requisitos

Antes de seguir esta guía se recomienda leer la guía **SOFIA2-Conceptos SOFIA2.doc** así como unos conocimientos mínimos del uso de Talend.

1.2 Objetivos y alcance del presente documento

Esta guía pretende introducir los componentes que permiten conectar la herramienta de ETL Talend con la plataforma SOFIA2.

Para eso se parte de:

- **SOFIA2 On Cloud:** es una versión de la Plataforma pública pensada para hacer pruebas de integración en la Plataforma. Es la instancia del SIB sobre la leeremos y escribiremos los datos guardados en la ontologías.
- La aplicación **Talend Data Integration** que nos permite desarrollar de forma sencilla procesos de extracción transformación y carga (ETL) de datos.
- Los **componentes de SOFIA2 para Talend** que permiten a Talend hacer consultas e insertar datos en las ontologías de SOFIA2.

2 INSTALACIÓN DE LOS COMPONENTES EN TALEND

2.1 Descargar componentes de Github

Para descargar los componentes debemos ir al repositorio de Github de Sofia2 y ahí nos descargaremos el fichero con la última versión: <https://github.com/Sofia2/TalendComponents>

4 commits 1 branch 0 releases 1 contributor

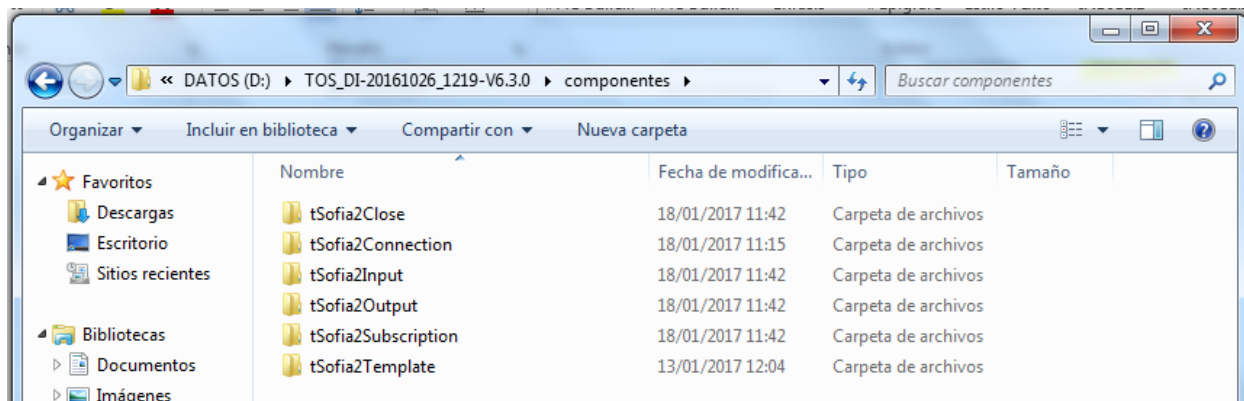
Branch: master New pull request Find file Clone or download

IgnacioMovellan Iconos modificados

Nombre	Descripción	Fecha de modificación
tSofia2Close	Iconos modificados	5 days ago
tSofia2Connection	Iconos modificados	5 days ago
tSofia2Input	Iconos modificados	5 days ago
tSofia2Output	Iconos modificados	5 days ago
tSofia2Subscription	Iconos modificados	5 days ago
tSofia2Template	Added standalone-api JAR	12 days ago
README.md	Initial commit	12 days ago

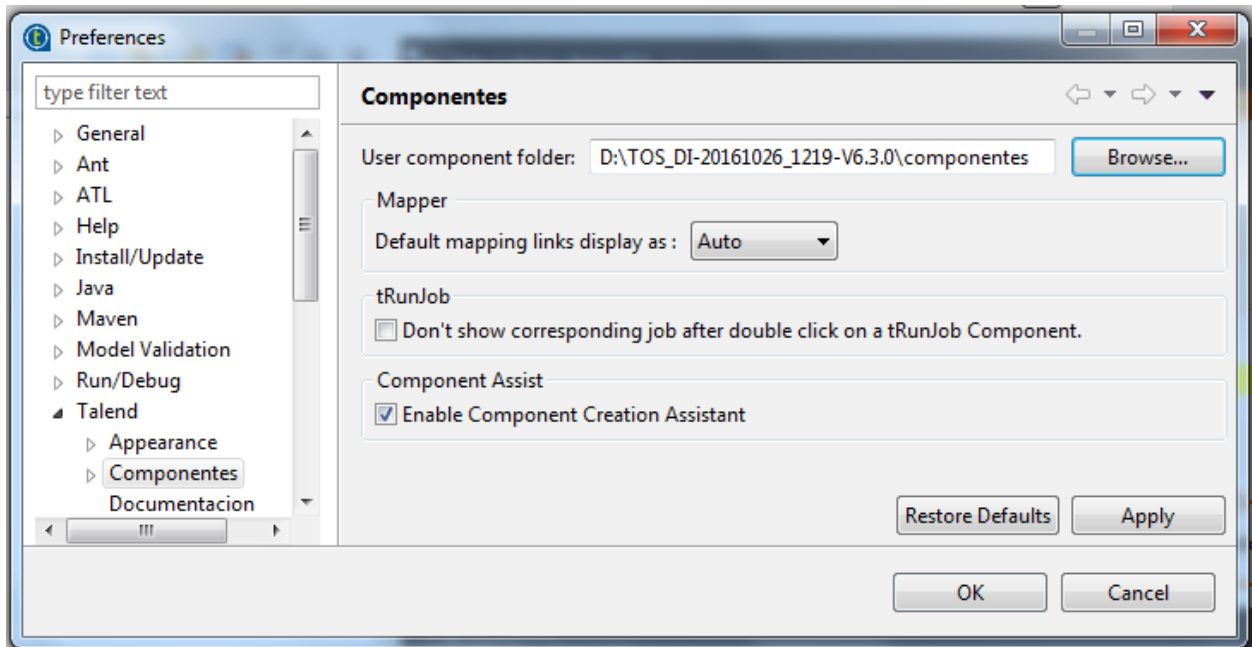
Clone with HTTPS
Use Git or checkout with SVN using the web URL.
<https://github.com/Sofia2/TalendComponents>
Open in Desktop Download ZIP

Una vez descargado el fichero lo descomprimos y copiamos las carpetas que hay dentro del directorio *TalendComponents-master* en una nueva carpeta que llamaremos *componentes* dentro de la instalación de nuestro Talend (en nuestro caso *D:\TOS_DI-20161026_1219-V6.3.0\componentes*).

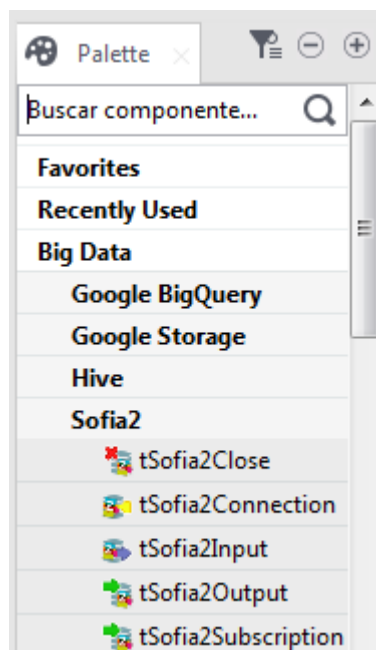


2.2 Configurar Talend para usar componentes de usuario

Para que Talend reconozca estos nuevos componentes debemos abrir la aplicación y ajustar las preferencias (**Ventana->Preferences** en el menú superior). En el menú que aparece elegiremos **Talend->Componentes** en el menú lateral y en el campo **User Component Folder** introducimos la ruta de la carpeta que hemos creado.



Ahora dándole a OK debería cargar nuestros componentes. En el futuro si queremos añadir más componentes bastaría con copiarlos en la misma carpeta y reiniciar Talend. Para comprobar que se han cargado correctamente, abrimos un nuevo Job de Talend y en la paleta nos tienen que aparecer los nuevos componentes dentro de la categoría **Big Data->Sofia2**.



3 CONFIGURACIÓN PREVIA EN SOFIA2

3.1 Darse de alta en la instancia Sofia2 inCloudLab

En la sección Productos de la Web de Sofia2 (<http://sofia2.com/sofia2incloud.html>) se puede acceder a **Sofia2 CloudLab**:

VERSIONES SOFIA2

-**Sofia2 CloudLab**: este entorno de experimentación permite comenzar a desarrollar con una cuenta gratuita teniendo acceso a todas las funcionalidades de Sofia2 y accediendo a multitud de datos disponibilizados. [Puedes probarlo aquí](#).

También se puede acceder vía: <http://sofia2.com/console/gestionusuarios/registerUser>

En esta URL podemos registrarnos:



Una vez que lo hagamos se nos asignará el rol **USUARIO**, este rol permite consumir información pública de la plataforma, pero no volcar información en esta.

Con este rol puedo crear Jobs que consuman (consulten, se suscriban) a las ontologías que otros usuarios hayan definido como públicas.

NOTA: Más tarde veremos como solicitar el rol **COLABORADOR**, que permite volcar información en la plataforma.

A la hora de crear el usuario es importante seguir las reglas que se indican, entre ellas que la password debe contener letras, números y letras mayúsculas y minúsculas.

3.2 Acceder a la Consola Web de Sofia2 CloudLab

Una vez que tengo mi usuario de Sofia2 CloudLab desde la página principal de Sofia2 (<https://sofia2.com/console>) puedo hacer Login:

Mensajes procesados por Sofia2 CloudLab - 2 4 3 . 4 4 1 . 0 3 2



Login

HOME PRODUCTO CASOS DE USO DESARROLLO NOSOTROS DEMOSTRADORES CONTACTO

O bien a través de este link: <http://sofia2.com/console/login>

Nos logaremos con el usuario creado en el punto anterior, la pantalla principal tiene este aspecto:

3.3 Búsqueda y Suscripción a Ontologías

Una vez dentro de la consola web el próximo paso es buscar **ontologías públicas** que pueda consumir desde mi APP y suscribirme a ellas para poder usarlas.

Una **Ontología** representa una Entidad viva en la Plataforma:

- Un usuario **COLABORADOR** puede crear Ontologías

- El **propietario de una Ontología** puede hacer **CRUDS** (INSERTAR/ ACTUALIZAR/ BORRAR/ CONSULTAR) sobre la Ontología
- El propietario de una Ontología **puede dar permisos a otros usuarios** para bien consultar, bien crear/modificar/borrar.
- El propietario de una Ontología puede hacer **PÚBLICA** una Ontología, en cuyo caso cualquier usuario de la Plataforma puede consultar esa Ontología.

Para buscar y suscribirme a las Ontologías pulsaremos en la opción **Suscribirse a una Ontología**. Y pondremos un criterio de búsqueda, buscaremos primero **talendDemo**:

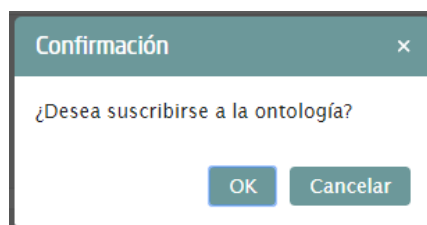
ONTOLOGÍAS / SUSCRIPCIÓN A ONTOLOGÍAS					
/ SUSCRIPCIÓN A ONTOLOGÍAS DE GRUPO					
▼ Filtrar Búsqueda					
LISTADO DE ONTOLOGÍAS					
Identificación	Valoración	Propietario	Activa	Pública	Opciones
talendDemo	★★★★★	colaborador_ignacio	✓	✓	👁
Showing 1 to 1 of 1 entries (filtered from 317 total entries)					
First Previous 1 Next Last					

Una vez encontrada seleccionaré el icono de **Ver**



En la pantalla de detalle puedo ver si una instancia está **Activa**, su **esquema** y un **ejemplo de Instancia de esa Ontología**.

Seleccionaré **Suscribirse**, al final de la página. La **suscripción** me permitirá usar esta Ontología (consultarla) desde mi APP.



Ahora en la pantalla principal aparecerá nuestra ontología en el Paso 1.

3.4 Creación de un KP

Tras suscribirme a las Ontologías debo crear un KP a través de la Consola Web.

Un KP representa una aplicación Sofia2, o de una forma más precisa **representa una conexión de una aplicación Sofia2 hacia la Plataforma.**

El primero paso es **Crear un KP**, para eso selecciono la opción Crear KP del paso 2 de la pantalla principal:

BIENVENIDO A LA CONSOLA WEB SOFIA2

Pasos para empezar a trabajar con la plataforma Paso a Colaborador

Paso 1

✓ Suscripción a Ontologías:

talendDemo ▼

Suscribirse a una Ontología

Paso 2

✗ Creación de Kp de la ontología: - No existen kps asociados a la ontología seleccionada -

Crear Kp

Cuando creo un KP debo darle un nombre, descripción y seleccionar las ontologías que se manejarán desde ese KP y pulsar el botón **Crear** al final de la página:

THINKPS SOFIA2 / NUEVO THINKP

ThinkKP

Mis Tokens

Mis Instancias

Identificación

TalendKP

Descripción

KP para la conexión desde Talend a la ontología

Ontología

Ontologías de Grupo

talendDemo-

Al crearlo me mostrará esta información:

THINKPS SOFIA2 / CONSULTAR THINKP

ThinkKP

Mis Tokens

Mis Instancias

Identificación

TalendKP

Usuario

usuario_ignacio

Descripción

KP para la conexión desde Talend a la ontología

Meta-Información

Ontologías

talendDemo

Cancelar

Crear

Modificar

Eliminar

En esta pestaña podemos ver el identificador del KP y si vamos a la pestaña **Mis Tokens** vemos el token de conexión que necesitaremos más adelante.

THINKPS SOFIA2 / CONSULTAR THINKP

ThinkKP

Mis Tokens

Mis Instancias

ThinkKP	Propietario	Token	Ult. Conexión	Activo
TalendKP	usuario_ignacio			

Showing 1 to 1 of 1 entries

First




Previous

1

Next

Last

Cancelar




Tras esto podré ver el KP creado en Mis ThinkKPs, desde esta ventana puedo verlo , editarlo  o borrarlo .

THINKPS SOFIA2 / MIS THINKKPS

Filtrar Búsqueda

Nuevo ThinkKP

LISTADO DE THINKKPS

Identificación	Descripción	Ontologías	Usuario	Opciones
TalendKP	KP para la conexión desde Talend a la ontología	talendDemo	usuario_ignacio	  

4 CÓMO DESARROLLAR UN JOB TALEND PARA SOFIA2

En el punto anterior hemos descargado los componentes de Sofia2 y hemos configurado Talend. En este punto explicaremos los pasos necesarios para desarrollar un Job sobre la instancia de Sofia2 publicada en <http://sofia2.com>.

Esta instancia de Sofia2: **Sofia2 CloudLab** ofrece un entorno de experimentación que permite comenzar a desarrollar con la plataforma con una cuenta gratuita teniendo acceso a todas las funcionalidades de Sofia2 y accediendo a multitud de datos publicados.

El proceso para desarrollar un Job con Sofia2 consiste básicamente en:

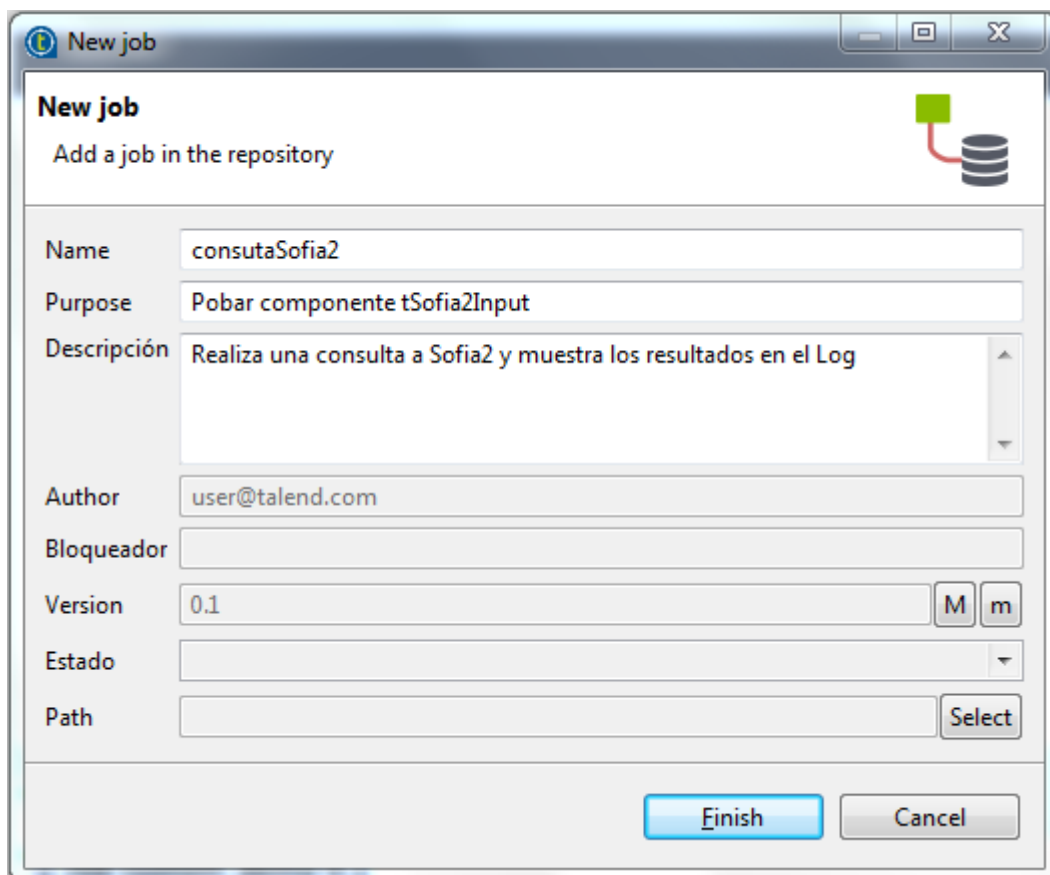
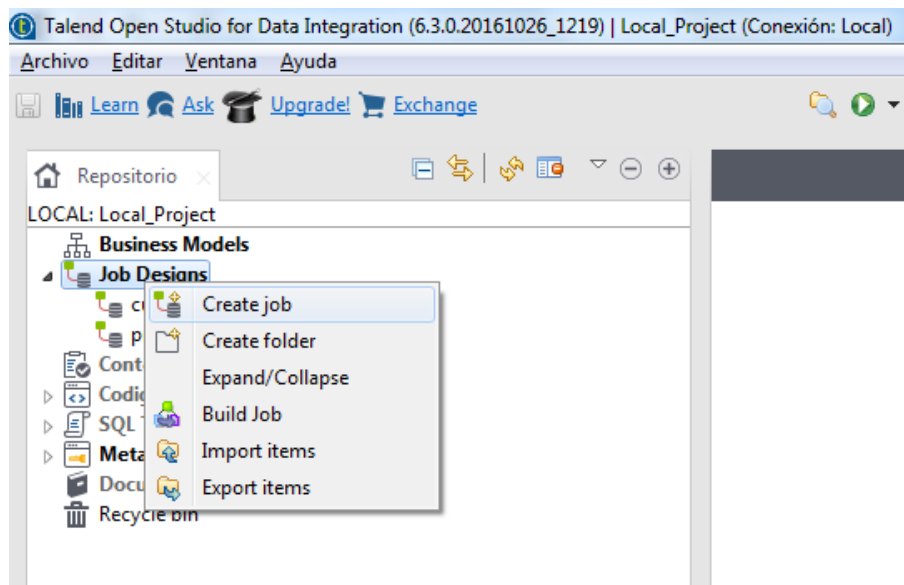
- Crear en Sofia2 un KP para conectarse a una ontología pública
 - Darse de alta en la instancia Sofia2 CloudLab
 - Acceder a la Consola Web de Sofia2 CloudLab
 - Búsqueda y suscripción a ontologías
 - Creación de APP (KP y Token)
 - Consultar información de estas ontologías desde Consola acceso BDTR
- Desarrollo de Job de consulta sobre las ontologías en Sofia2
- Desarrollo de Job que inserta en Sofia2
- Desarrollo de Job que se suscribe a cambios en ontologías o eventos

4.1 Job Talend de consulta a Sofia2

Hasta el momento:

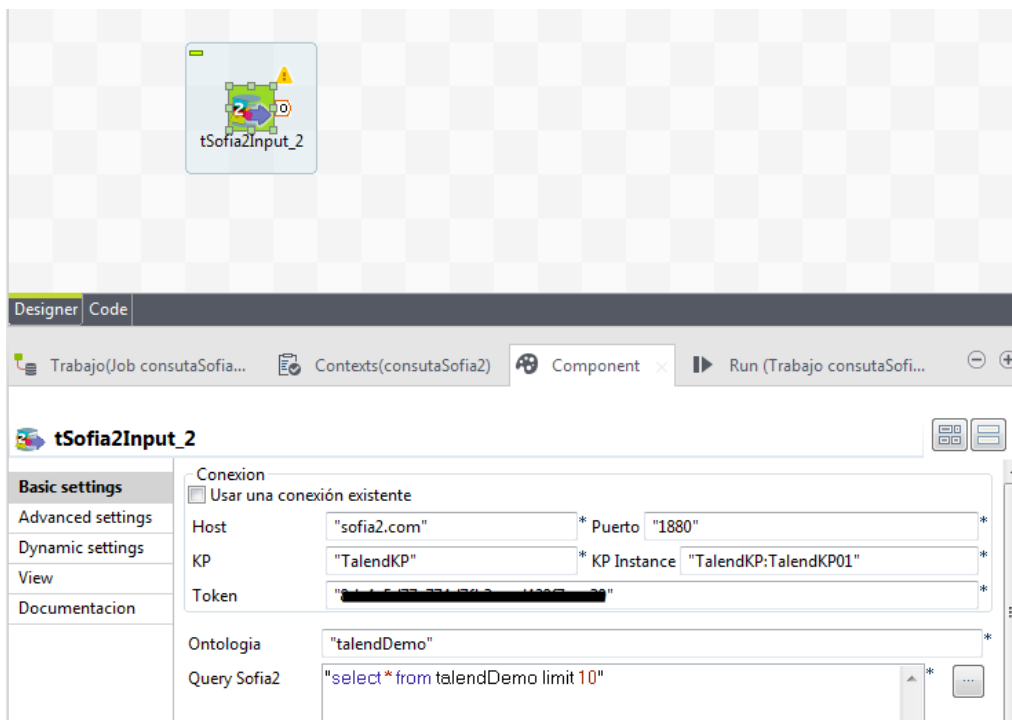
- Nos hemos suscrito a una ontología pública lo que me permite consultarla
- Hemos creado un KP con un Token

En este paso vamos a crear un Job de Talend que haga una consulta a esta ontología. Para ello empezamos creando un job vacío en Talend.



4.1.1 Configuración del componente tSofia2Input

A continuación iremos a la paleta (a la derecha de la pantalla) y en la categoría **Big Data->Sofia2** elegimos el componente **tSofia2Input** y pinchamos en algún lugar del job para dejar una instancia del componente. A continuación hacemos doble click en el componente de nuestro job y en la parte inferior configuramos el componente:



En host y puerto podemos dejar los valores por defecto (*sofia2.com* y *1880* respectivamente).

En KP, Ontología y Token ponemos los que hemos creado y en la consulta ponemos:

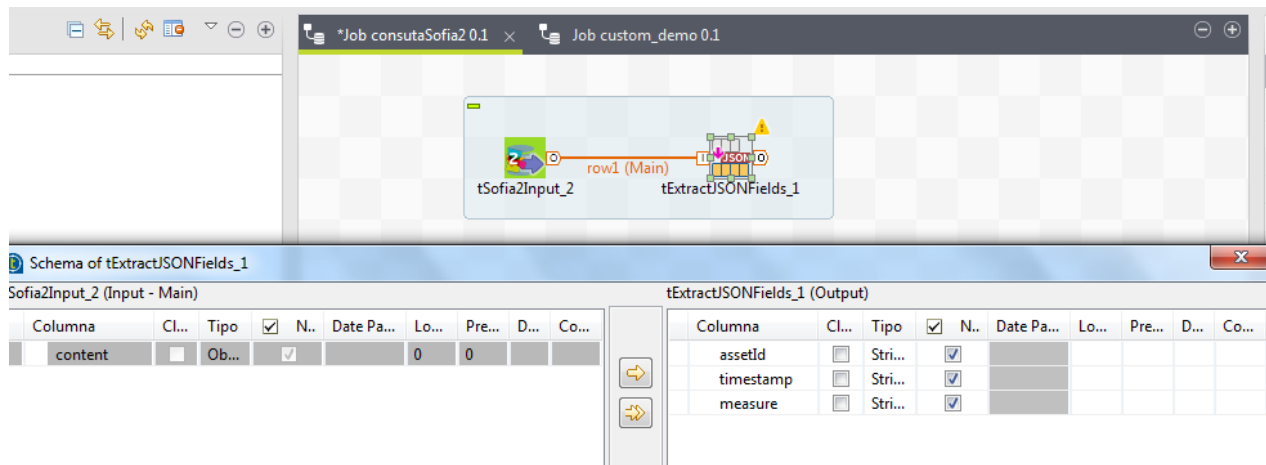
```
"select * from talendDemo limit 10"
```

Es importante limitar el resultado de la consulta, sobre todo si trabajamos con ontologías que tienen un gran volumen de datos.

Si vemos el esquema de este componente solo tiene una columna *content* de salida, aquí nos devolverá el resultado de la consulta como un único array JSON con todos los datos obtenidos.

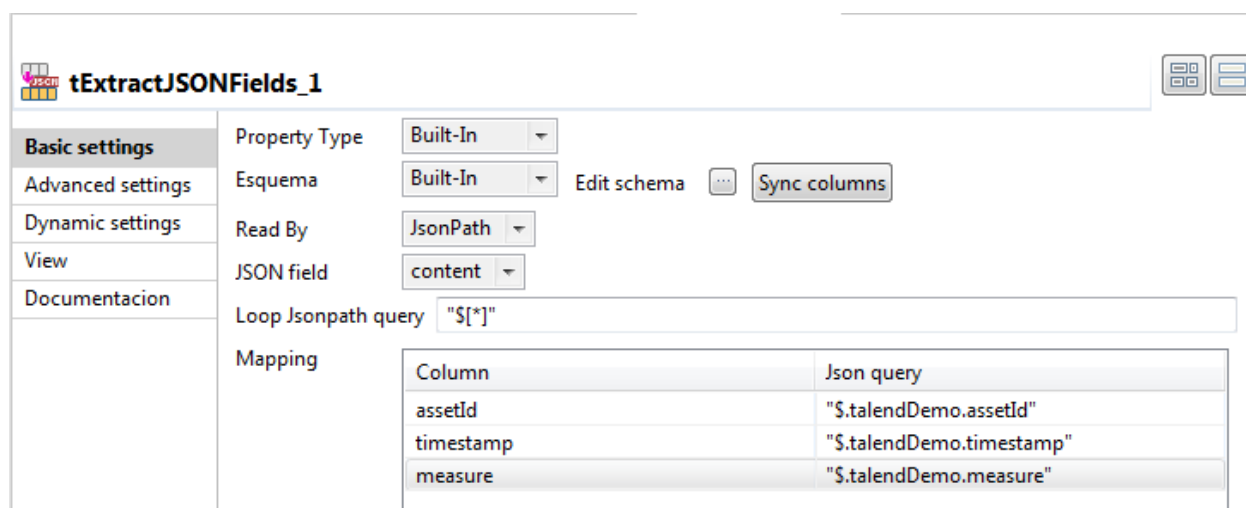
4.1.2 Conversión de formato JSON a filas

Para extraer los datos del array JSON devuelto por tSofia2Input, debemos poner un componente **tExtractJSONFields**. Para usarlo debemos editar el esquema de este componente y añadir en la salida las columnas que queremos obtener, en nuestro caso *assetId*, *timestamp* y *measure*.



The screenshot shows the Talend Studio interface. At the top, there are two job windows: "Job consultaSofia2 0.1" and "Job custom_demo 0.1". Below them, a workflow diagram shows a component labeled "row1 (Main)" connected to "tSofia2Input_2" and "tExtractJSONFields_1". Below the diagram, the "Schema of tExtractJSONFields_1" window is open, showing the "tExtractJSONFields_1 (Output)" schema. The schema table has columns: Columna, Cl..., Tipo, N., Date Pa..., Lo..., Pre..., D..., Co... The table contains three rows: "assetId", "timestamp", and "measure".

Ahora debemos decirle al componente como debe rellenar estos campos. En las propiedades elegimos lo siguiente:

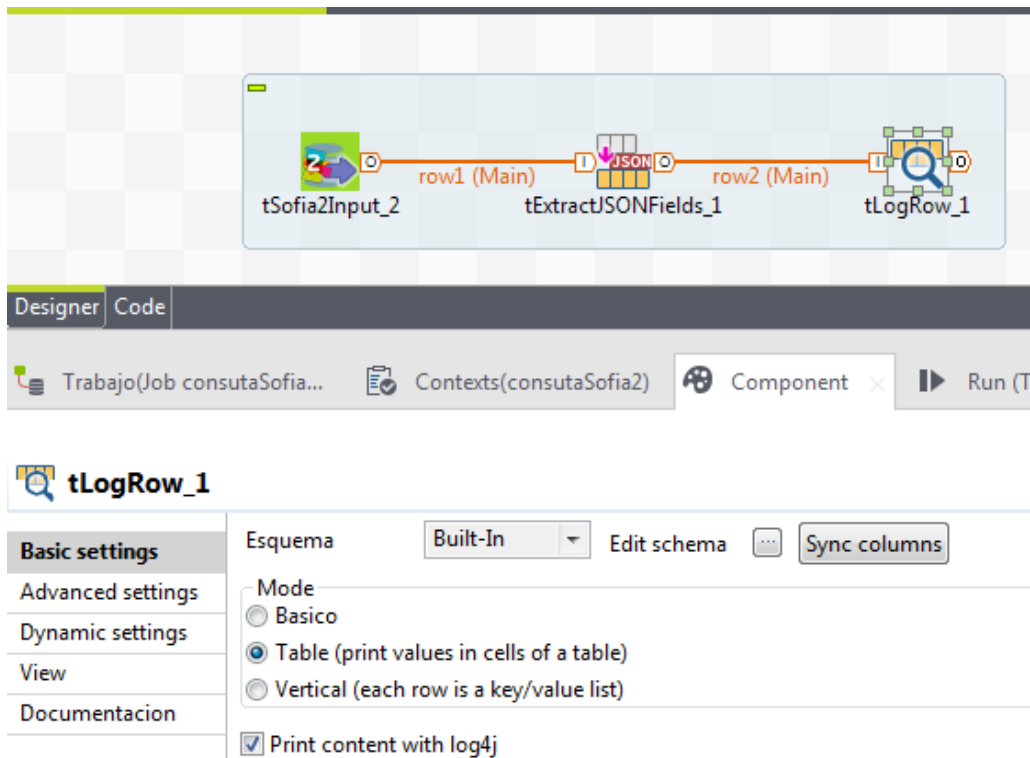


The screenshot shows the properties window for the "tExtractJSONFields_1" component. The "Basic settings" tab is selected. The properties are: "Property Type" is Built-In, "Esquema" is Built-In, "Read By" is JsonPath, "JSON field" is content, and "Loop Jsonpath query" is "\$[*]". There is a "Sync columns" button. The "Mapping" table shows columns: Column, Json query, with rows: assetId, timestamp, and measure.

Loop Jsonpath query indica que debe iterar en cada elemento del array inicial ("[\$*]"). Y para cada elemento de esta query mapeamos los campos que hemos definido. Para más información sobre como mapear los campos de un objeto JSON se puede acudir a la documentación de la librería Jsonpath.

4.1.3 Visualización de la salida

Para comprobar que se ha leído correctamente, añadimos al final un paso tLogRow que configuramos en modo tabla:



tLogRow_1

Basic settings

Advanced settings

Dynamic settings

View

Documentacion

Esquema: Built-In

Mode:

- ☐ Basico
- ☒ Table (print values in cells of a table)
- ☐ Vertical (each row is a key/value list)

☒ Print content with log4j

Ahora ya podemos ejecutar el Job y nos debería dar un resultado como el siguiente:

```

-----+-----+-----+
|                                     tLogRow_1                                     |
|-----+-----+-----+
| assetId      | timestamp                                     | measure |
|-----+-----+-----+
| Concord      | {"$date":"2017-01-09T16:36:55.000Z"} | 8        |
| Dover        | {"$date":"2017-01-09T16:36:55.000Z"} | 1.3      |
| Salt Lake City | {"$date":"2017-01-09T16:36:55.000Z"} | 5.7      |
| Indianapolis  | {"$date":"2017-01-09T16:57:26.000Z"} | 5.8      |
| Lansing       | {"$date":"2017-01-09T16:57:26.000Z"} | 7.4      |
| Annapolis     | {"$date":"2017-01-09T16:57:26.000Z"} | 5        |
| Juneau        | {"$date":"2017-01-10T08:19:20.000Z"} | 9.3      |
| Olympia       | {"$date":"2017-01-10T08:19:20.000Z"} | 7.6      |
| Carson City   | {"$date":"2017-01-10T08:19:20.000Z"} | 9.9      |
| Baton Rouge   | {"$date":"2017-01-10T08:22:48.000Z"} | 1.8      |
|-----+-----+-----+

```

4.2 Job Talend de inserción en Sofia2

En este paso vamos a crear un Job de Talend que inserte valores aleatorios en la ontología que a la que nos hemos suscrito. Para ello primero debemos tener permisos de colaborador en Sofia2, esto se consigue pulsando el botón **Paso a Colaborador** que hay en la parte superior derecha de la pantalla de inicio. Al pulsar nos pedirá un motivo para nuestro cambio de rol y un administrador estudiará nuestra petición.

Sofia2

WIZARD { REST } usuario_ignacio [ROL_USUARIO] Idioma

BIENVENIDO A LA CONSOLA WEB SOFIA2

Pasos para empezar a trabajar con la plataforma **Paso a Colaborador**

Paso 1	Paso 2	Paso 3
Suscripción a Ontologías:	Creación de Kp de la ontología:	Tokens del Kp:

Una vez que tengamos permisos de colaborador creamos un nuevo Job:

New job
Add a job in the repository

Name: insertSofia2

Purpose: Probar el componente tSofia2Output

Descripción: Un simulador que escribe valores aleatorios en una ontología de prueba.

Author: user@talend.com

Bloqueador:

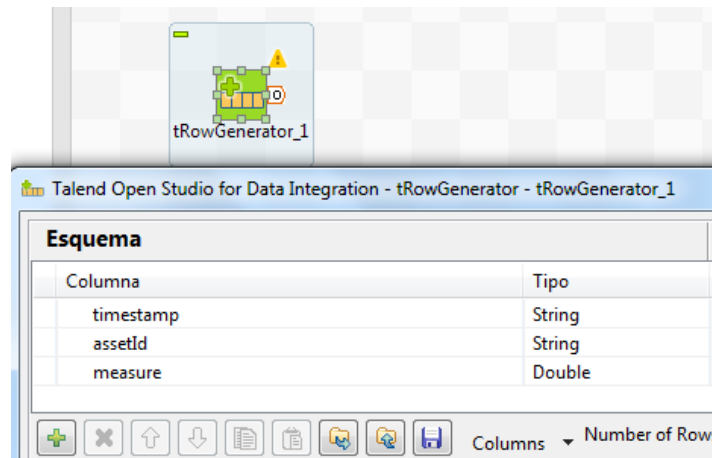
Version: 0.1

Estado:


Path:

4.2.1 Generación de filas aleatorias

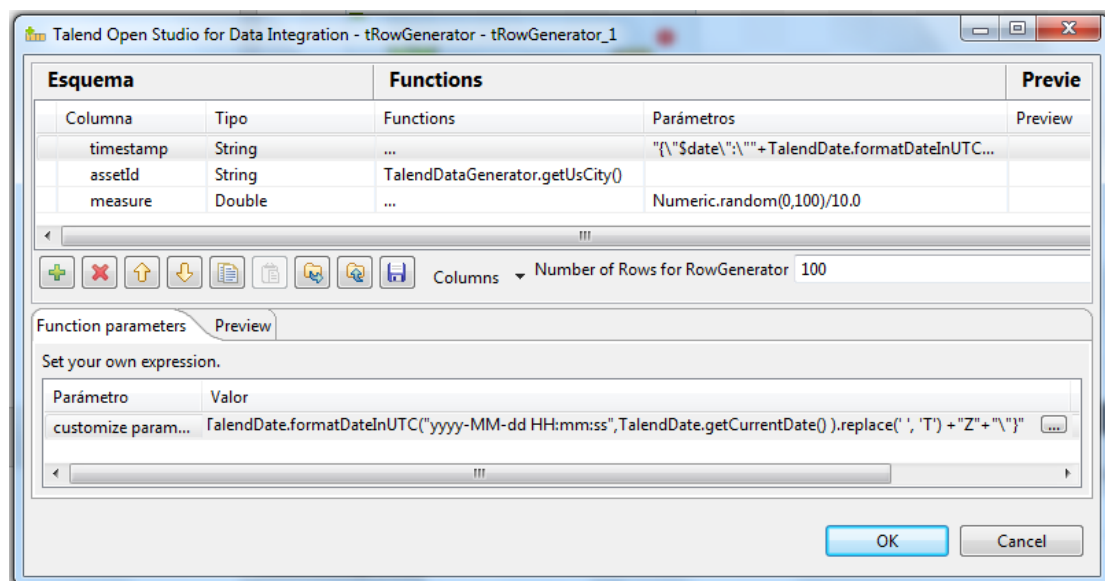
Para tener datos de prueba que insertar en la ontología usamos el componente tRowGenerator y abrimos su editor para añadir 3 columnas en el esquema:



Para generar estas columnas lo hacemos de la siguiente manera:

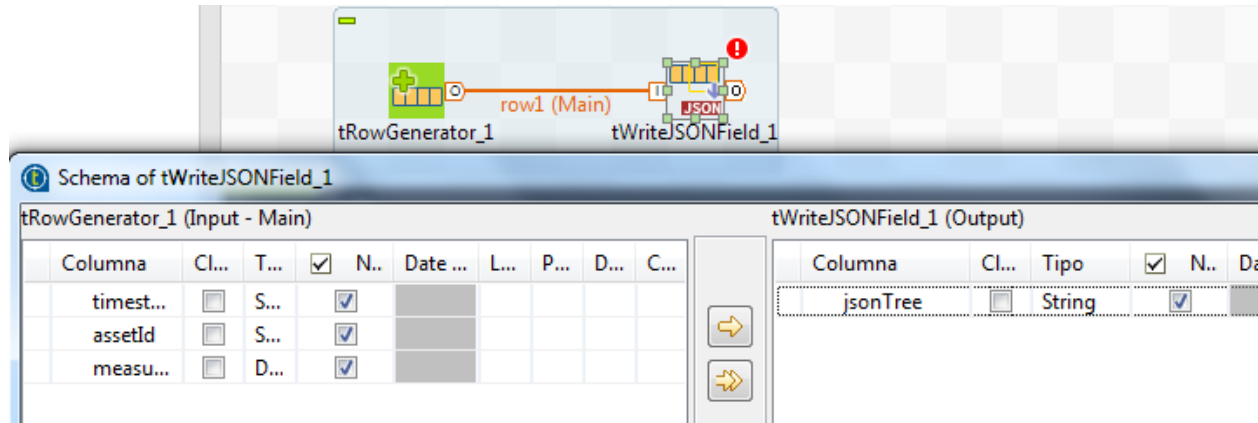
- **timestamp:** en la columna función elegimos los puntos suspensivos (la primera opción del desplegable). En function parameters (en la parte inferior) abrimos el editor () de la columna valor y escribimos lo siguiente:
`"{"$date\":" + TalendDate.formatDateInUTC("yyyy-MM-dd HH:mm:ss", TalendDate.getCurrentDate()).replace(' ', 'T') + "Z"+"\"}"`
- **measure:** en la columna función elegimos los puntos suspensivos (la primera opción del desplegable). En function parameters (en la parte inferior) escribimos lo siguiente en la columna valor:
`Numeric.random(0,100)/10.0`
- **assetId:** En la columna función elegimos la opción:
`TalendDataGenerator.getUsCity()`

Nos quedará algo como esto:



4.2.2 Paso a formato JSON

Para insertar en Sofia2 debemos convertir las filas que hemos generado a un objeto JSON. Para ello usaremos el componente `tWriteJSONFile` y en su esquema definiremos un campo **jsonTree** de tipo *String*.



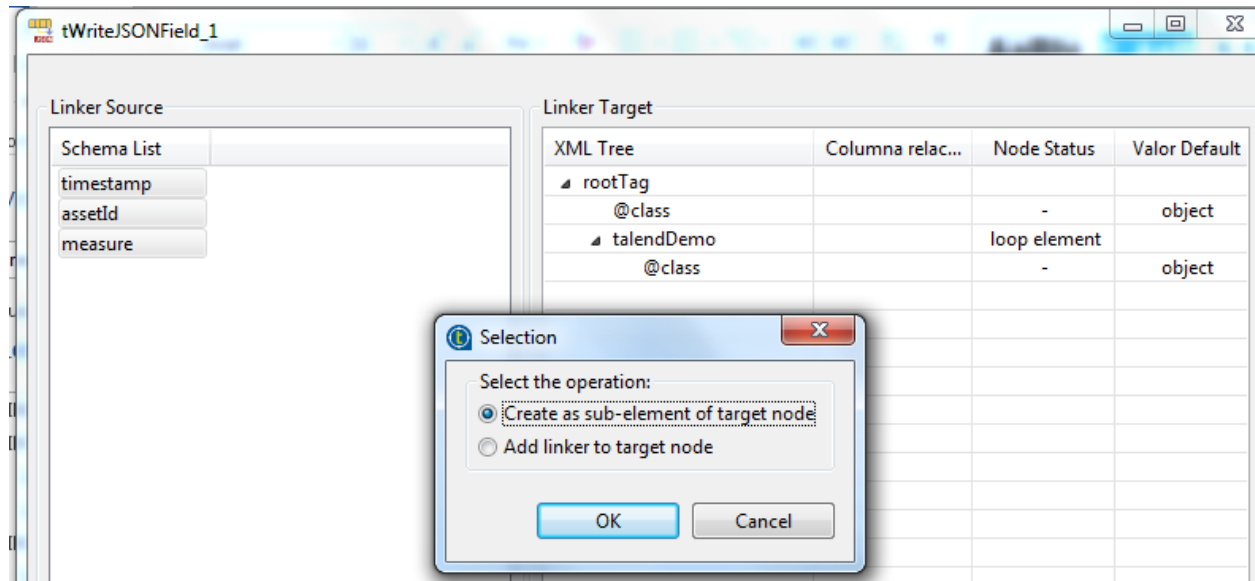
A continuación hacemos doble clic en el componente para configurar nuestro objeto JSON. En la pantalla que nos aparece debemos añadir el atributo **class** a la raíz (botón derecho en *rootTag* y seleccionar *Add Attribute*) y pinchando en el nuevo elemento *@class* hacemos doble clic y seleccionamos *Set A Fix Value* y escribimos **object**.

Linker Target			
XML Tree	Columna relac...	Node Status	Valor Default
▲ rootTag			
@class		-	object

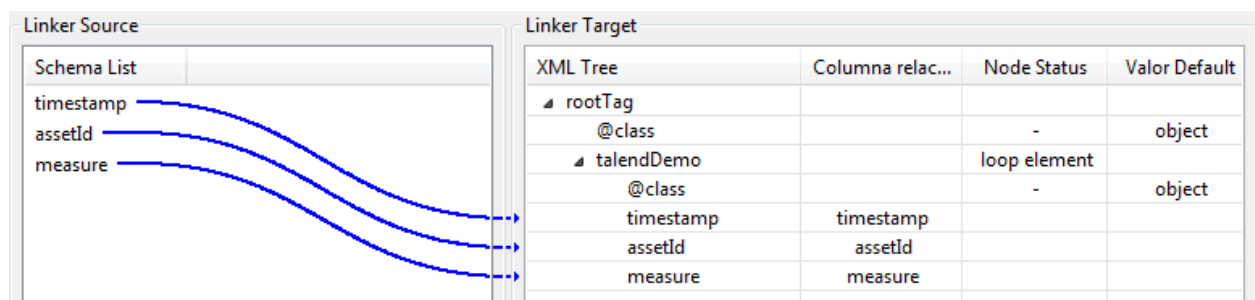
El siguiente paso es añadir el objeto que contiene la ontología. Para ello pulsamos el botón derecho sobre el *rootTag* y elegimos *Add Sub-Element* y le damos el nombre **talendDemo**. A este sub elemento le añadimos también el atributo **class** con el valor **object**. Para terminar de configurar el objeto, lo marcamos como Loop Element (botón derecho sobre *talendDemo* y seleccionamos *Establecer como Elemento Bucle*).

Linker Target			
XML Tree	Columna relac...	Node Status	Valor Default
▲ rootTag			
@class		-	object
▲ talendDemo		loop element	
@class		-	object

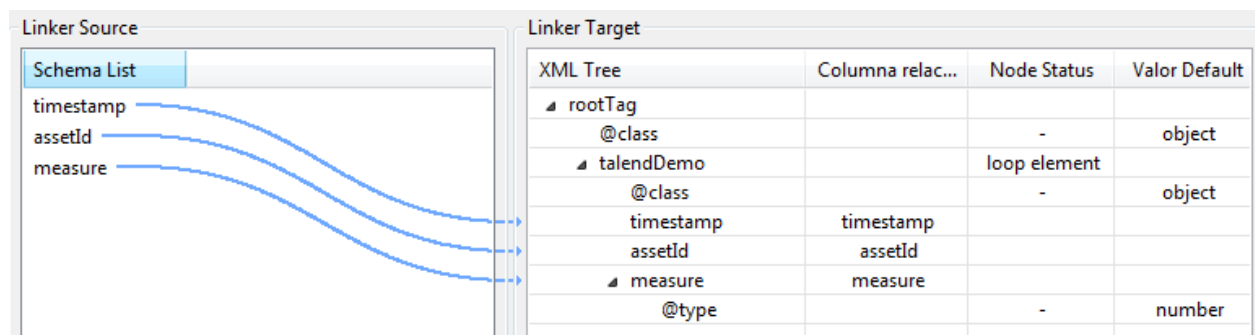
Ahora ya podemos arrastrar los 3 elementos que aparecen a la izquierda y los soltamos encima del objeto *talendDemo*. Al soltarlos nos preguntará que queremos hacer con esos elementos y elegiremos **Create as sub-element of target node**.



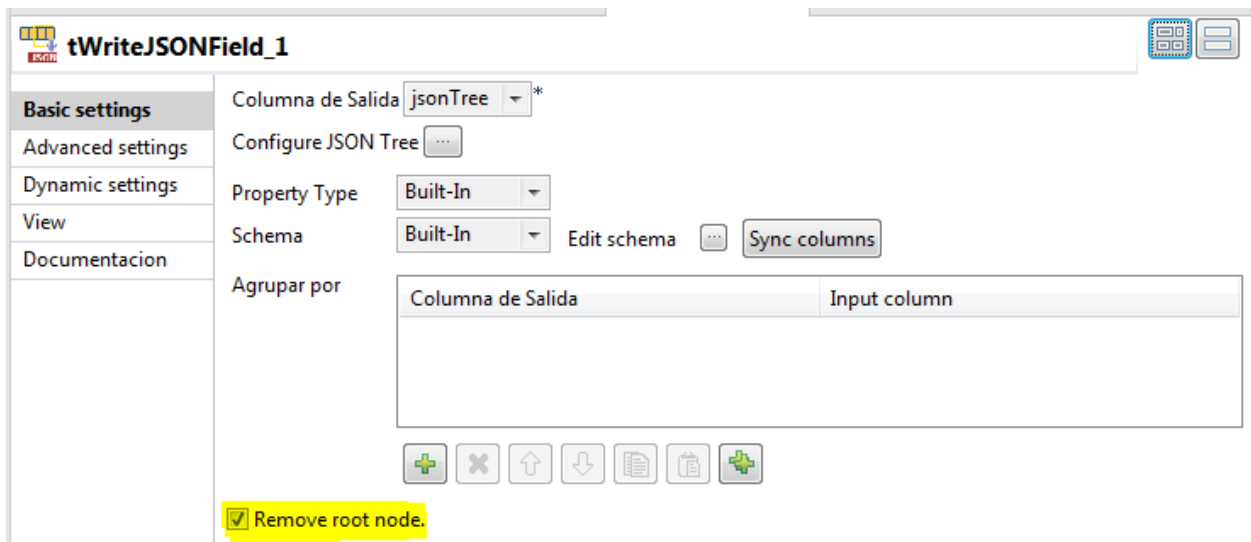
Ahora nuestro objeto JSON queda de la siguiente manera:



Por defecto este componente marca todos los campos como String, pero nuestro campo *measure* es un doble. Entonces tenemos que añadir un atributo a *measure* con nombre **type** y valor **number**. El resultado final es el siguiente:

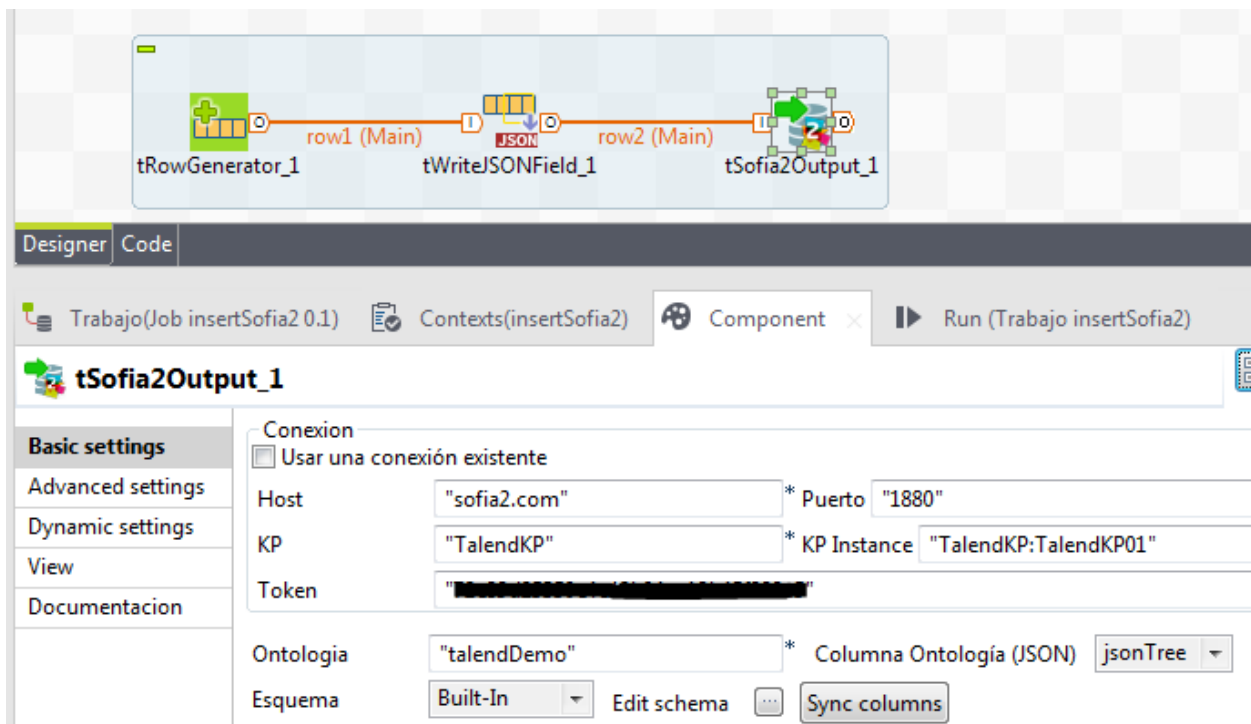


Por último en la configuración del componente, seleccionando el componente y eligiendo la pestaña *Component* de la ventana inferior, marcamos eliminar el nodo raíz:



4.2.3 Configuración de tSofia2Output

Ahora que ya tenemos convertida nuestra fila de datos a un objeto JSON ya podemos insertarla en Sofia2. Para ello cogemos el objeto tSofia2Ouput de la paleta, lo conectamos a la salida del tWriteJSONField y lo configuramos de la siguiente manera:



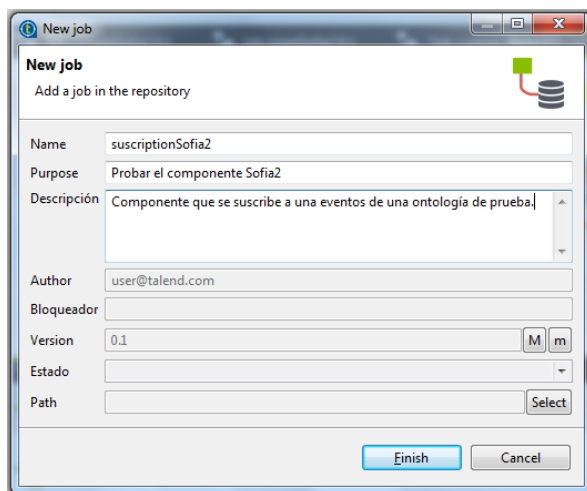
Con esto podemos ejecutar el job y debería insertar 100 líneas aleatorias en la ontología. Podemos consultar estas líneas con un job de consulta como el del punto anterior.

4.3 Job Talend de suscripción a una ontología de Sofia2

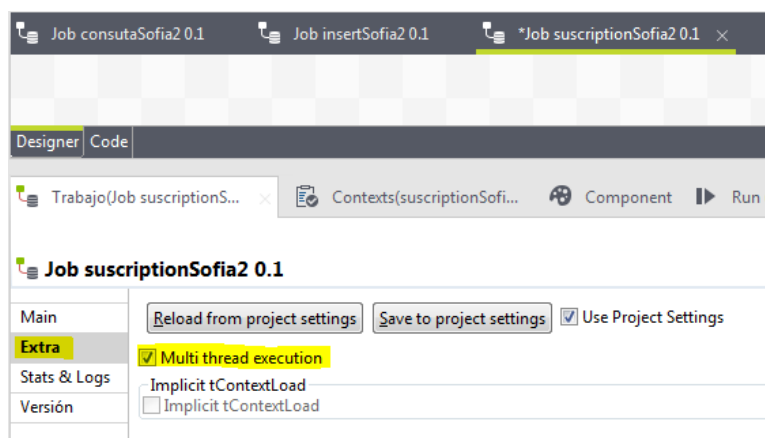
Por último vamos a crear un Job que se suscriba a una ontología, usando el componente **tSofia2Subscription**. Con este componente podemos suscribirnos de 3 maneras distintas:

- **Ontología:** Genera una línea de datos por cada instancia de una ontología que se inserte en Sofia2
- **Query Sofia2:** Genera una línea de datos por cada evento insertado en una ontología que cumpla la consulta dada.
- **CEP:** Genera una línea de datos por cada Evento que se genere usando el motor de eventos de Sofia2.

En este ejemplo nos suscribiremos a todos los eventos que tengan un valor de *measure* superior a 5 usando el modo Query a Sofia2. Vamos a empezar creando el job:



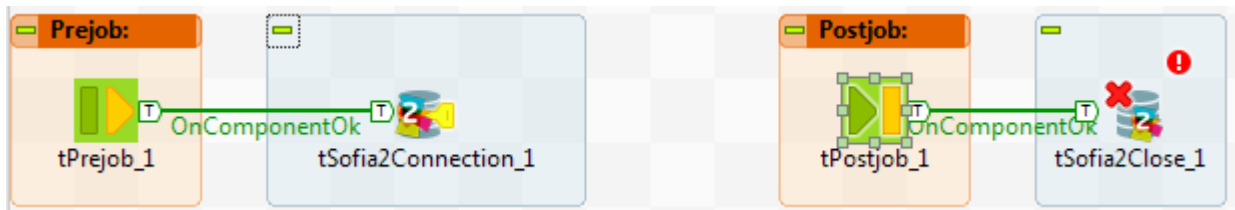
Para poder probar la suscripción necesitamos que se estén insertando datos mientras nuestro Job está en ejecución. Para simular esto, haremos que éste sea multihilo y ejecute en paralelo un simulador que inserte (una copia del job del punto anterior) con nuestra suscripción. Para ello en la pestaña Trabajo de la parte inferior marcamos dentro de **Extra, Multi thread execution**.



4.3.1 Conexión reutilizable

Como vamos a tener dos componentes que se conectarán a Sofia2 con el mismo KP y Token, vamos a aprovechar a crear una conexión compartida usando los componentes **tSofia2Connection** y **tSofia2Close** que abren y cierran dicha conexión.

Para ello vamos a arrastrar estos componentes junto a tPreJob y tPostJob desde la paleta al área de diseño y los conectaremos de la siguiente manera:



A continuación configuramos la conexión dentro del componente tSofia2Connection:

tSofia2Connection_1	
Basic settings	Host "sofia2.com" * Puerto "1880" *
Advanced settings	KP "TalendKP" *
Dynamic settings	KP Instance "TalendKP:TalendKP01" *
View	Token "XXXXXXXXXXXXXXXXXXXX" *
Documentacion	

Y dentro de tSofia2Close le indicamos que debe cerrar la conexión abierta por el componente anterior:

tSofia2Close_1	
Basic settings	Conexion tSofia2Connection_1 *
Advanced settings	

Al conectarlos estos componentes al tPreJob y al tPostJob nos aseguramos de que la conexión se crea antes de que arranque cualquier otro hilo y que se cierre al terminar el Job aunque este termine con error.

4.3.2 Generación de filas aleatorias

A continuación montamos el simulador, para esto podemos copiar los componentes del Job anterior:



Pero modificaremos el componente de salida para que use la conexión que hemos definido anteriormente. Para ello marcamos usar conexión existente:

tSofia2Output_1

Basic settings

Advanced settings

Dynamic settings

View

Documentacion

Conexion

☒ Usar una conexión existente Conexion tSofia2Connection_1 *

Ontologia "talendDemo" * Columna Ontologia (JSON) jsonTree

Esquema Built-In Edit schema Sync columns

4.3.3 Suscripción a la consulta

Ahora crearemos el subjob encargado de la suscripción. Para ello seleccionaremos el componente tSofia2Subscription, lo pondremos en el área de diseño y lo configuraremos de la siguiente manera:

tSofia2Subscription_1

Basic settings

Advanced settings

Dynamic settings

View

Documentacion

Conexion

☒ Usar una conexión existente Conexion tSofia2Connection_1 *

Tipo de suscripción Query Sofia2 Tiempo de ejecución (s) 10 ☐ Ejecución continua

Ontologia "talendDemo"

Query Sofia2 "select * from talendDemo where talendDemo.measure > 5"

Es importante ver que hemos desmarcado la opción **Ejecución continua** y hemos dado un **tiempo de ejecución** de 10 segundos. Esto hace que el proceso pare después del tiempo indicado. De lo contrario estaría ejecutando continuamente hasta que el usuario lo pare manualmente. Otra forma de hacer que el proceso pare es usando el componente **tDie** que puede conectarse a un filtro para tener una parada en función de los datos recibidos.

4.3.4 Lectura de objeto JSON

De forma similar a como lo hacía el componente tSofia2Input, éste nos devuelve un objeto JSON, pero en lugar de tener un array con todos los resultados tenemos una fila con un objeto por cada resultado recibido.

Para leer este objeto JSON podemos copiar el componente que hicimos para el job de consulta, pero modificando la consulta del bucle para que sea "\$" en lugar de "\$[*]".

Column	Json query
assetId	\$.talendDemo.assetId
timestamp	\$.talendDemo.timestamp
measure	\$.talendDemo.measure

Finalmente añadimos un componente de tLogRow a continuación para visualizar los resultados:

Column	Json query
assetId	\$.talendDemo.assetId
timestamp	\$.talendDemo.timestamp
measure	\$.talendDemo.measure