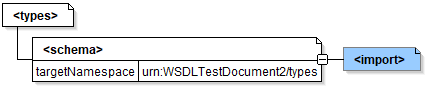
Añadir los datos correspondientes al “WSDL”

Primero vamos a decirle al sistema que queremos importar un “**namespace**”. Podemos hacerlo de dos formas.

1. La primera es mediante la vista “Schema”:
   1. Del menú de la derecha (**Components**) seleccionamos el correspondiente a **import** [] y lo arrastramos hacia el “**Schema**”



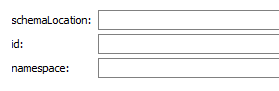
* + 1. Quedaría de la siguiente forma:



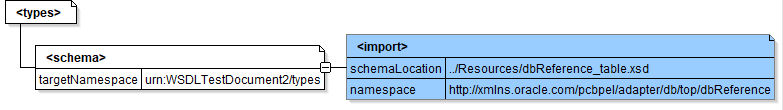
* + 1. Antes de continuar, veamos como va quedando el código. Ahora que insertamos un componente **import**:



* + 1. Ahora vamos a los “**properties**” de **import** para agregar los datos correspondientes:



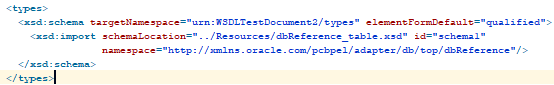
* + - 1. **SchemaLocation**: Ubicación del “**schema”** o archivo “**XSD**” dentro del proyecto: “schemaLocation="../Resources/dbReference\_table.xsd"”
      2. **id**: Identificador que va a tomar.
      3. **Namespace:** En este caso, vamos a realizar un “**WSDL**” que sirva de intermediario para el “**XSD**” existente, por lo que el namespace debe ser el mismo:
         1. "http://xmlns.oracle.com/pcbpel/adapter/db/top/dbReference"
    1. Una vez completados los datos, el resultado obtenido en el “**Schema**” es el siguiente:



* + 1. Si vamos a la pestaña de código, se verá de la siguiente forma:

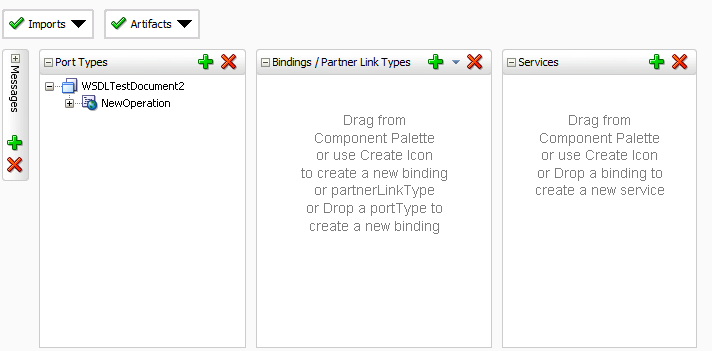


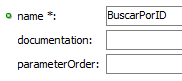
1. Mediante “**Source**”:
   1. En este caso es fácil darse cuenta que hacer. Si cada nodo que pongamos en “Schema” representa un elemento “**XML**” en “**source**”, lo mismo pasa a la inversa.
   2. Por lo que podemos ir directo al código y poner la etiqueta de importación correspondiente:



* 1. Esto quiere decir que cualquier cambio que hagamos en cualquiera de las tres vistas, se verá reflejado en el archivo “**source**” y lo mismo pasa de la forma inversa.
  2. Además, deberíamos añadir a la cabecera del “**source**” el siguiente valor, copiado exactamente como está en el xsd de referencia:
     1. “xmlns:inp1="http://xmlns.oracle.com/pcbpel/adapter/db/top/dbReference”
     2. Eso va dentro de la etiqueta [**definitions**]
     3. Esto es para decirle el prefijo de los elementos a referenciar.

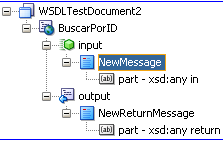
Ahora vamos a modificar desde la vista de diseño los datos correspondientes a los mensajes de entrada y salida que debe manejar el “**WSDL**”.

1. Primero nos dirigimos a la vista “**Design**”:
   1. 
2. Aquí vemos 3 secciones:
   1. **Port Types**: Aquí declaramos la estructura del mensaje de entrada y salida y las posibles operaciones que se pueden realizar.
   2. **Bindings**: Este va a crear un link de acceso al puerto.
   3. **Services**: Por defecto jDevelop se va a encargar de llenar este campo con los datos correspondientes, pero se puede modificar a gusto.
3. Vamos a cambiar el nombre de la operación para poner algo más entendible.
   1. Damos clic sobre “NewOperation” y esto nos mostrará la ventana de property:



* 1. En este caso solo nos interesa cambiar el nombre. Le debemos poner un nombre descriptivo y entendible.

1. En la sección de “**Port Types**” desplegamos hasta tener el árbol totalmente abierto:



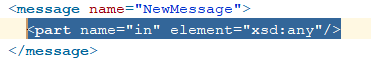
1. Vemos la estructura un momento:
   1. Tenemos dos ramas principales [**input**] y [**output**] las cuales van a almacenar los datos correspondientes a los mensajes de entrada y salida.
   2. Cada uno tiene dentro creado por defecto un mensaje. Como sus nombres indican, son los mensajes a utilizar para recibir y devolver datos.
   3. Como último nivel, tenemos dos archivos [**part**], estos se encargan de decir el tipo de dato de entrada y salida.
2. Como hicimos con la operación, vamos a cambiar los nombres de los mensajes por algo más descriptivo:
   1. Input:



* 1. Output:

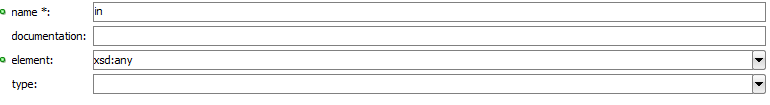


\* Puede que te de error si cambias el nombre de los mensajes y no te permita seguir modificando hacia abajo en la jerarquía. Para ello, podemos ir al código y Volver a poner el nombre del mensaje antiguo (Se debe cambiar en los dos lugares). Nombramos primeros las variables [**part**] y después volvemos renombrar los mensajes:

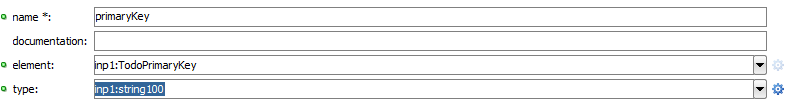




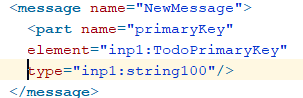
1. Primero vamos a modificar el [**part**] del input. Damos clic sobre [] para acceder a la pestaña de properties:



* 1. **name**: Nombre que recibirá el parámetro de entrada. Pude ser el que desees siempre que sea descriptivo o siga con las normas del proyecto.
  2. **documentation:** Sería una breve descripción del campo.
  3. **element:** aquí hay que tener en cuenta que, si no tenemos un “**namespace**” declarado, entonces no saldrá la opción de [**select - ]**
  4. **type**: lo mismo que “**element**”, es el tipo de dato que va a tener la variable que utilicemos; si no tenemos “**namespace**” no podremos usar el select.



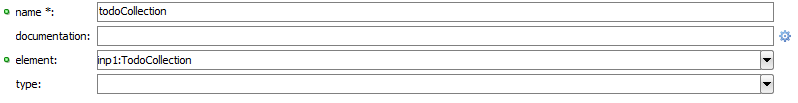
* 1. Así quedaría con todos los datos llenos. Los valores de “**element**” y “**type**” son tomados del XDS creado.
  2. En código se vería de la siguiente forma:



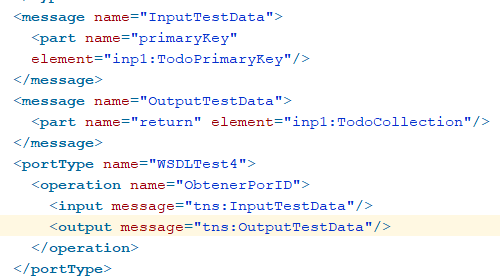
1. Ahora hacemos el mismo procedimiento para los datos correspondientes al output:

\* Recuerda que, si haces algún cambio en el diseño y no te deja seguir trabajando, puedes ir al “**source**” y revertirlo con [**ctrl + z**]

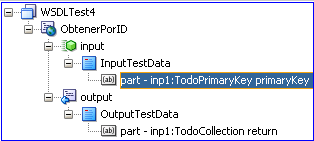
\* También puede darse el caso que si le pones “**type**” al [**part**] siga dando error. Le quitamos el “**type**” y ya no hay problema.



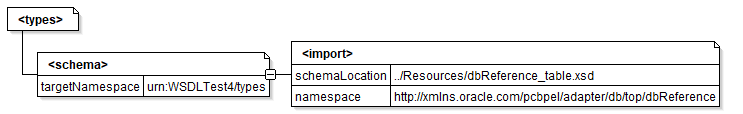
1. Después de todas las modificaciones a los [**part**], [**messages**] y [**operation**] quedaría así:
   1. **Source**:



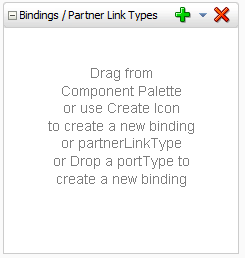
* 1. **Design**:



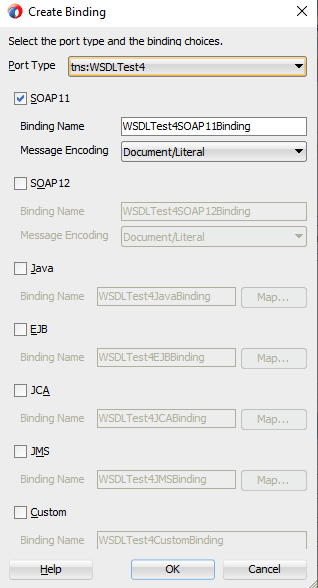
* 1. **Schema**:



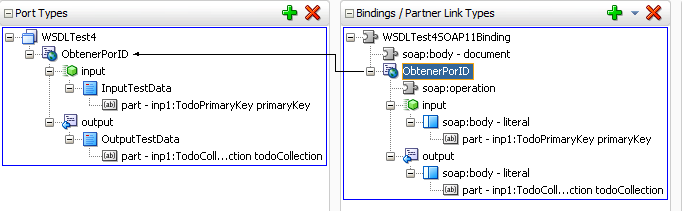
1. Para terminar la creación del WSDL, vamos a “**Design**” y nos concentramos en la sección de “**Bindings**”:



* 1. Presionamos el [+ -> ] para acceder a la ventana de creación de “Bingings”.



* 1. Dentro tenemos varias opciones, pero las que nos interesan son:
     1. **Port Type**: a que puerto se le va a realizar la vinculación. Podemos seleccionar cualquiera de los que hayamos creados dentro del WSDL.
     2. Las otras opciones corresponden al tipo de vinculación que queramos hacer. Esto depende de cómo estén realizando el proyecto. Para este caso se va a seleccionar SOAP 11.
        1. **Binding name**: Nombre para identificar el componente.
        2. **Message Encode**: Tipo de formato a utilizar para manejar los mensajes. Por defecto dejamos el []
  2. Presionamos ok y con esto terminamos la creación del “**Binding**”.



* 1. En la imagen se puede observar como el realiza las vinculaciones entre el “**Port**” y el “**Binding**”.
  2. Para probar ver **tutorial 3 - 1.3**