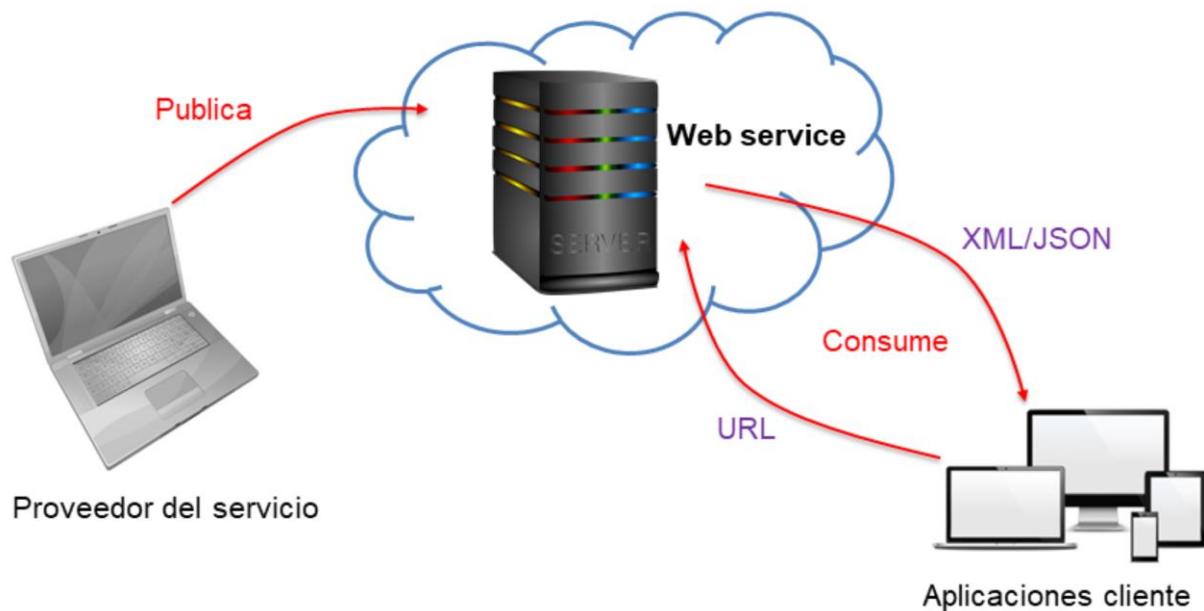


# Conceptos básicos de Web Services

Introducción

*GeneXus® 16*

A continuación veremos qué son los web services y cómo podemos usar estos servicios en una aplicación GeneXus.



Los Web Services son programas que brindan funcionalidades útiles a otros programas y son ubicados en servidores web para que puedan ser localizados e invocados a través de una red, generalmente Internet.

El proveedor de servicios "publica" un Web Service en un servidor y las aplicaciones cliente "consumen" el Web Service publicado.

Para acceder al servicio, la aplicación cliente utiliza su localización (URL) para invocarlo y eventualmente le envía los parámetros requeridos.

A cambio recibe la información devuelta, generalmente como una estructura en formato XML o JSON.



Los Web Services pueden ser desarrollados siguiendo distintos estándares, los más comunes en la industria son SOAP y REST. Cada estándar define cómo debe publicarse la información de las funciones disponibles en el web service.

Los web services SOAP utilizan una definición escrita en WSDL (Web Services Description Language), mientras que los servicios REST utilizan el estándar OpenAPI.

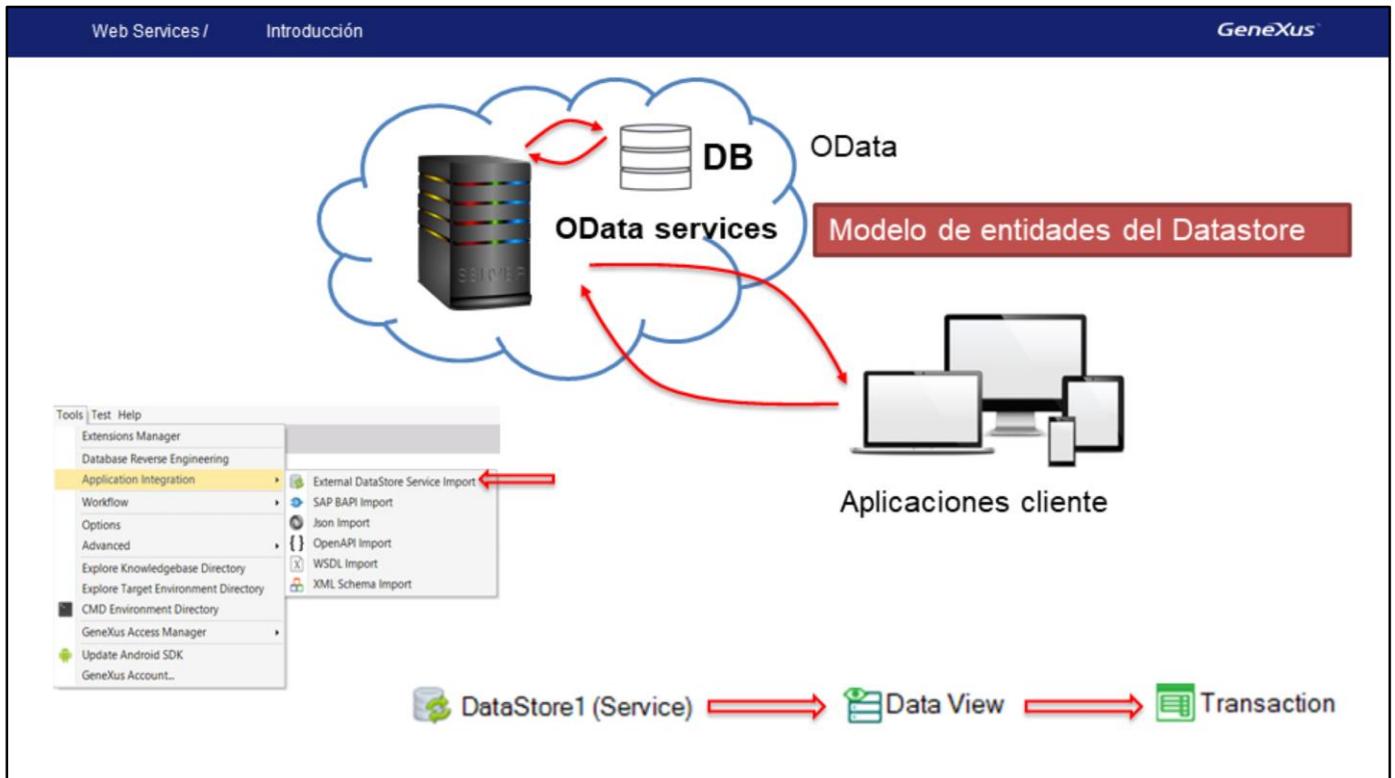
Para acceder a un Web Service publicado, debemos conocer su localización (URL) e importar su definición para poder tener acceso a las funciones disponibles en el servicio web.

GeneXus permite consumir Web Services que hayan sido desarrollados en cualquier herramienta de programación o plataforma, con protocolos SOAP o REST.

Para integrar un Web service a una aplicación GeneXus, vamos a Tools/Application Integration, elegimos WSDL Import si el web service sigue el protocolo SOAP y OpenAPI Import si el webservice tiene arquitectura REST.

Esto disparará un wizard que variará dependiendo del tipo de servicio web elegido.

Al finalizar, si el webservice era SOAP, GeneXus creará automáticamente un Objeto Externo asociado al Web Service y los tipos de datos estructurados necesarios para manejar sus datos. Si en cambio era REST, se crearán automáticamente una serie de objetos GeneXus, (que por lo general incluiremos en un módulo, por ejemplo WebServices) y que nos permitirán ejecutar el servicio directamente a través de esos objetos en nuestra KB. El wizard dejará en una carpeta API los programas a invocar y en una carpeta Model los SDTs para manejar los datos.



Un tipo especial de servicios web son los llamados OData services. Estos servicios utilizan el protocolo Open Data Protocol (Odata) diseñado para proveer operaciones de insertar, modificar o eliminar registros de una base de datos, a través de un sitio web.

Así como en un webservice Soap o Rest había que importar primero su definición, para consumir un servicio Odata primero debemos importar el modelo con las entidades de la base de datos incluidas en el servicio.

Para hacerlo, vamos a Tools/Application Integration y elegimos External DataStore Service Import. Como el proceso implica crear un DataStore del tipo Servicio para asociarlo el datastore externo, sólo podemos usar esta funcionalidad en GeneXus Full.

Luego de ejecutado el wizard, se crearán tantos objeto transacción como entidades estaban en el modelo y podremos trabajar con ellas como con cualquier otra transacción de nuestra aplicación.



Aplicación GeneXus

## Objetos:

- Procedure
- Business component
- Data Provider

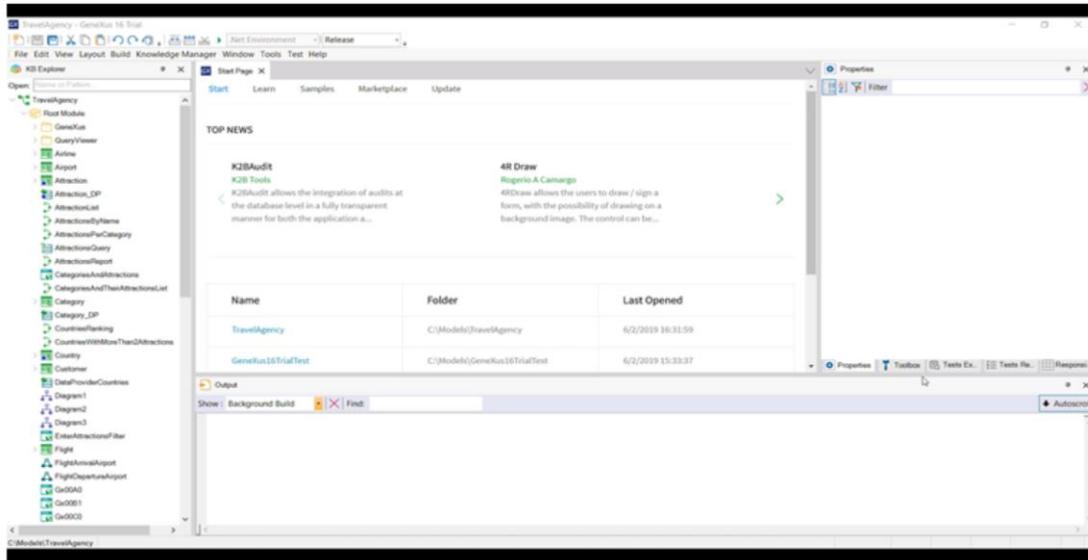


Properties	
Filter	
Procedure: MyWebService	
Name	MyWebService
Description	My Web Service
Module/Folder	Root Module
Main program	False
Call protocol	Internal
Execute in new LUW	False
Qualified Name	MyWebService
Object Visibility	Public
Interoperability	
Expose as Web Service	True
Web Service Protocol	
SOAP Protocol	True
REST Protocol	True
Generate OpenAPI interface	Use Environment property value

GeneXus también permite crear web services y publicarlos en un servidor web.

Los objetos GeneXus que pueden ser expuestos como un web service son los procedimientos, los business components y los data providers.

## Ejemplo de uso de un web service REST



[ DEMO: <https://youtu.be/bvmt0Gjcxpw> ]

Como ejemplo de lo visto, crearemos un web service REST que inserte una aerolínea nueva en la base de datos y lo invocaremos desde nuestra aplicación.

Para hacerlo, antes que nada abrimos la transacción Airline y ponemos su propiedad Business Component en True. Salvamos.

Luego vamos a crear un objeto procedimiento que reciba por parámetro los datos de una aerolínea e inserte un registro nuevo en la tabla Airline.

Llamamos al procedimiento CreateNewAirlineWS y definimos una regla Parm con 2 variables de entrada: &AirlineName y &AirlineDiscountPercentage. Definimos estas dos variables usando Add Variable y luego en la sección Variables definimos una variable más, Airline del tipo business component Airline.

En el source escribimos el código para insertar una aerolínea con los datos recibidos por parámetro. No necesitamos darle valor al AirlineId ya que es autonumerado.

Ahora vamos a las propiedades del objeto procedimiento y asignamos **Expose as Web Service** en True, **SOAP Protocol** en False, **REST Protocol** lo dejamos en True y **Generate OpenAPI interface** en Yes. Esto último permitirá que se genere la definición de nuestro webservice en el estándar OpenAPI.

Damos botón derecho sobre el procedimiento y elegimos Build With This Only, con lo que el servicio se publicará en nuestro servidor web, en este caso en nuestra máquina local. En la ventana de Ouput vemos un mensaje que nos confirma que se generó la documentación de la API Rest del webservice, con la definición del mismo.

Antes de importar el webservice, vamos a crear un módulo nuevo, al que llamamos WebServices, así queda todo guardado en ese módulo.

Para importar el webservice, vamos a Tools/Application Integration y elegimos OpenAPI Import. En el File Path / URL escribimos: C:\Model\TravelAgency\CSharpModel\Web\default.yaml que es donde quedó generada la documentación de la API Rest y elegimos como destino el módulo WebServices. Vemos que se importó todo correctamente y si abrimos el módulo WebServices, en el folder Api está el procedimiento

importado. También vemos que en el folder Model hay un SDT llamado CreateNewAirlineWSInput, que si lo abrimos vemos que los parámetros que precisamos pasarle al servicio están disponibles aquí.

Para probar que el web service funcione correctamente, creamos un web panel llamado CreateNewAirlineUsingWS. Luego arrastramos un botón al form y al evento lo llamamos Create airline. Ahora vamos a las variables y definimos una variable &CreateNewAirlineWSInput que automáticamente queda definida del tipo del SDT.

Para poder tener un feedback del resultado de la ejecución del webservice, creamos también una variable &IsSuccess y otra &HttpMessage, vemos que los tipos se asignan por defecto.

Damos doble clic sobre el botón y en el evento cargamos los miembros del SDT, luego invocamos al webservice pasando el SDT como parámetro y por último escribimos el siguiente código para dar mensajes por pantalla

Presionamos F5... Vemos las aerolíneas que tenemos ingresadas... Y ejecutamos el webpanel que creamos recién.

Presionamos el botón y vemos que el servicio nos informa que la aerolínea se creó correctamente.

Para verificar vamos a la transacción Airline... Y vemos que efectivamente se agregó la aerolínea que queríamos.

## Ejemplo de web service SOAP

<https://training.genexus.com/genexus/curso-genexus-15-analista?es#web-services-gx15>

Aquí vimos un de utilización de un webservice tipo REST. Si desea ver cómo usar un servicio SOAP, puede consultar el siguiente video:

<https://training.genexus.com/genexus/curso-genexus-15-analista?es#web-services-gx15>

## Más información sobre Web Services

WSDL: <https://wiki.genexus.com/commwiki/servlet/wiki?6181>  
OpenAPI: <https://wiki.genexus.com/commwiki/servlet/wiki?31864>  
OData: <https://wiki.genexus.com/commwiki/servlet/wiki?40713>

Por más información sobre webservices en GeneXus, siga los siguientes links del wiki:

WSDL: <https://wiki.genexus.com/commwiki/servlet/wiki?6181>  
OpenAPI: <https://wiki.genexus.com/commwiki/servlet/wiki?31864>  
OData: <https://wiki.genexus.com/commwiki/servlet/wiki?40713>

# GeneXus™

**The power of doing.**

Videos

[training.genexus.com](http://training.genexus.com)

Documentation

[wiki.genexus.com](http://wiki.genexus.com)

Certifications

[training.genexus.com/certifications](http://training.genexus.com/certifications)