

WSMG

Catálogo de productos de Tiendas

17 April 2022

Versión 1.3

Contenidos

WSMG	1
1. Implementación WSMG Integración del Web Service de Productos	2
1.1. Introducción	2
1.2. Resumen de Operación	2
2. WSDL del Web Service	2
2.1. WSDL para la API de WSMG	2
3. Métodos de referencia	3
3.1. GetCatalog(string cliente, string usuario, string password) como un ProductRequest ...	3
3.2. GetStockPriceCatalog(string cliente, string usuario, string password) como un ProductRequest	4
3.3. GetCategories(string cliente, string usuario, string password) como un CategoryRequest	4
3.4. GetBrands(string cliente, string usuario, string password) como un BrandRequest	5
3.5. GetErrors(string cliente, string usuario, string password) como un ErrorRequest	5
3.6. GetXMLContenido(string cliente, string usuario, string password) como un ContentRequest	6
4. Semántica de datos retornados por el servicio	7
4.1. Método GetCatalog	7
4.2. Método GetStockPriceCatalog	8
4.3. Método GetCategories	8
4.4. Método GetBrands	9
4.5. Método GetErrors	9
4.6. Método GetXMLContenido	10
4.7. Método GetStockPriceCatalogTime	10
5. Consumir servicio desde el lado del cliente	11
5.1. NET	11

1. Implementación WSMG

Integración del Web Service de Productos.

1.1. Introducción

El presente documento esboza el proceso en el cual ustedes realizan pedidos al Servicio Web.

1.2. Resumen de Operación

Al momento de llevar a cabo una petición, mediante una aplicación web (implementada según su tecnología) ustedes invocarán a nuestro Web Service, enviándoles de regreso un objeto complejo conteniendo dicha petición.

2. WSDL del Web Service

2.1. WSDL para la API de WSMG

El WSDL para la API de WSMG está disponible en:

<https://ecommerce.microglobal.com.ar/WSMG/WSMG.asmx?WSDL>

O bien para realizar pruebas en:

https://ecommerce.microglobal.com.ar/WSMG_back/WSMG.asmx?WSDL

El WSDL está disponible para que sea 'entendible' por las herramientas de desarrollo de su preferencia – no está diseñado para lectura humana directa. Cualquier set de herramientas debe ser capaz de consumir este (SOAP) servicio (Java, PHP, .NET, etc.).

Agregando una referencia de servicio (en .NET) genera un 'proxy', un set de Clases y Métodos (código en su aplicación) localmente que expone los objetos remotos – otras plataformas pueden tener algo equivalente. En PHP, por ejemplo, puede ser aún más sencillo:-

<https://solaajayi.wordpress.com/2013/12/04/beginners-guide-to-consuming-a-web-service-using-php-soapclient/>

Generalmente Usted no debería intentar escribir cualquier cosa que consuma o emita un XML – esta 'capa' debe ser gestionada por su set de herramientas – dejándole a Usted el tema de los 'objetos'.

3. Métodos de referencia

3.1. GetCatalog(string cliente, string usuario, string password) como un ProductRequest

Usted llamará a esta función para obtener el catálogo.

Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `ProductsRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public string originDate;  
public List<Product> listProducts = new List<Product>();
```

Asimismo, los componentes de `Product` son:

```
public string partNumber;  
public string descripcion;  
public double precio;  
public int stock;  
public float iva_pct;  
public string partNumber_ori;  
public string upc;  
public string codCategoria;  
public string categoria;  
public string codMarca;  
public string marca;  
public string bundle;  
public int peso;  
public int ancho;  
public int alto;  
public int profundidad;
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `listProducts` será no nulo conteniendo lo mencionado anteriormente.
- Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `listProducts` será nulo.

3.2. `GetStockPriceCatalog(string cliente, string usuario, string password)` como un `ProductRequest`

Usted llamará a esta función para obtener el catálogo de precios y stock.

Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `ProductsRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public List<Product> listProducts = new List<Product>();
```

Asimismo, los componentes de `Product` son:

```
public string partNumber;  
public double precio;  
public int stock;
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `listProducts` será no nulo conteniendo lo mencionado anteriormente.
- Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `listProducts` será nulo.

3.3. `GetCategories(string cliente, string usuario, string password)` como un `CategoryRequest`

Usted llamará a esta función para obtener las categorías correspondientes.

Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `CategoryRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public List<Category> listCategories = new List<Category>();
```

Asimismo, los componentes de `Category` son:

```
public string codCategoria;  
public double categoria;
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `listCategories` será no nulo conteniendo lo mencionado anteriormente.
- Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `listCategories` será nulo.

3.4. `GetBrands(string cliente, string usuario, string password)` como un `BrandRequest`

Usted llamará a esta función para obtener las marcas correspondientes. Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `BrandRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public List<Brand> listBrands = new List<Brand>();
```

Asimismo, los componentes de `Brand` son:

```
public string codMarca;  
public double marca;
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `listBrands` será no nulo conteniendo lo mencionado anteriormente.
- Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `listBrands` será nulo.

3.5. `GetErrors(string cliente, string usuario, string password)` como un `ErrorRequest`

Usted llamará a esta función para obtener los errores correspondientes. Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `ErrorRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public List<Error> listErrors = new List<Error>();
```

Asimismo, los componentes de `Error` son:

```
public int codError;  
public string error;
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `listErrors` será no nulo conteniendo lo mencionado anteriormente.

Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `listErrors` será nulo.

3.6. GetXMLContenido(string cliente, string usuario, string password) como un ContentRequest

Usted llamará a esta función para obtener los artículos y su correspondiente contenido. Deberá proveer el cliente provisto por MG, usuario no nulo caso de existir y la apropiada contraseña.

El valor de respuesta que obtendrá se corresponde a un objeto del tipo `ContentRequest`, cuyos componentes son:

```
public int result;  
public string message;  
public XmlDocument content = new XmlDocument();
```

Según el resultado de la operación:

- Si la operación es exitosa, `result` contendrá el valor entero "0", `message` cadena vacía y `content` será no nulo conteniendo lo mencionado anteriormente.

Si la operación no es exitosa, `result` contendrá un valor entero distinto de "0", `message` tendrá el mensaje apropiado para informar dicho error y `content` será nulo.

4. Semántica de datos retornados por el servicio

Adicional al archivo WSDL de Web Services en etapa de desarrollo, se complementa la información aclarando la semántica de los datos retornados por el servicio.

4.1. Método [GetCatalog](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'ProductsRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Fecha Origen	originDate	Char(10)	Fecha de origen en Catálogo de Productos.
Productos	listProducts	Colección <i>Product</i>	Lista de Objetos de Clase <i>Product</i>

Colección de Objeto **Product**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
PartNumber	partNumber	Char(50)	Partnumber Único en Microglobal
Descripción	descripcion	Char(200)	Descripción del producto
Precio	precio	Decimal (16,2)	Precio acordado con Tienda
Stock	stock	Entero	Cantidad disponible del producto
Iva	iva_pct	Decimal(5,2)	Porcentaje de alícuota de Iva (21,10.8 o 0)
PartNumber_Ori	partNumber_ori	Char(50)	PartNumber del Fabricante
UPC	upc	Char(13)	Código UPC del producto (vacío si no tiene)
Código Categoría	codCategoria	Char(2)	Código Categoría del producto
Categoría	categoria	Char(40)	Descripción de la Categoría
Código Marca	codMarca	Char(2)	Código Marca del producto
Marca	marca	Char(40)	Descripción de la Marca
Código de Bundle	bundle	Char(10)	Código de Bundle (vacío si no tiene)
Peso	Peso	Entero	Peso en gramos
Ancho	Ancho	Entero	Ancho en milímetros
Alto	Alto	Entero	Alto en milímetros
Profundidad	Profundidad	Entero	Profundidad en milímetros

4.2. [Método GetStockPriceCatalog](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'ProductsRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Productos	listProducts	Colección <i>Product</i>	Lista de Objetos de Clase <i>Product</i>

Colección de Objeto **Product**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
PartNumber	partNumber	Char(50)	Partnumber Único en Microglobal
Precio	precio	Decimal (16,2)	Precio acordado con Tienda
Stock	stock	Entero	Cantidad disponible del producto

4.3. [Método GetCategories](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'CategoryRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Categorías	listCategories	Colección <i>Category</i>	Lista de Objetos de Clase <i>Category</i>

Colección de Objeto **Category**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Categoría	codCategoria	Char(2)	Código Categoría
Categoría	categoria	Char(40)	Descripción de la Categoría

4.4. [Método GetBrands](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'BrandRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Marcas	listBrands	Colección <i>Brand</i>	Lista de Objetos de Clase <i>Brand</i>

Colección de Objeto **Brand**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Marca	codMarca	Char(2)	Código Marca
Marca	marca	Char(40)	Descripción de la Marca

4.5. [Método GetErrors](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'ErrorRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Errores	listErrors	Colección <i>Error</i>	Lista de Objetos de Clase <i>Error</i>

Colección de Objeto **Error**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Error	codError	Entero	Código Error
Error	error	Char(200)	Descripción del Error

4.6. [Método GetXMLContenido](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web

Parámetros de Salida: Se retornará un objeto complejo 'ContentRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Contenido	content	XML (DOM)	Objeto de Clase XmlDocument (XML DOM)

4.7. [Método GetStockPriceCatalogTime](#)

Parámetros de Entrada:

Atributo	Nombre en XML	Tipo de Dato	Comentario
Cliente	cliente	Char(7)	Código de Cliente en Microglobal
Usuario	usuario	Char(10)	En Blanco si es Administrador
Password	password	Char(30)	Contraseña de acceso a servicios Web
Fecha	Fecha	Char(30)	Fecha desde que se registran cambios (ejemplo formato 2022-04-26T23:36:11)

Parámetros de Salida: Se retornará un objeto complejo 'ProductsRequest' que contendrá

Atributo	Nombre en XML	Tipo de Dato	Comentario
Código Retorno	result	Entero	Tipo Resultado (0 si OK, sino Error)
Descripción	message	Char(200)	Descripción del Error / Mensaje
Productos	listProducts	Colección <i>Product</i>	Lista de Objetos de Clase <i>Product</i>

Colección de Objeto **Product**. Cada uno de ellos posee los siguientes atributos:

Atributo	Nombre en XML	Tipo de Dato	Comentario
PartNumber	partNumber	Char(50)	Partnumber Único en Microglobal
Precio	precio	Decimal (16,2)	Precio acordado con Tienda
Stock	stock	Entero	Cantidad disponible del producto

5. Consumir servicio desde el lado del cliente

Se adiciona además ejemplos de consumo de operaciones del servicio.

5.1. NET

Para probar el servicio, deberá crear un cliente y usarlo para llamar al servicio. Por un lado, puede hacerlo usando la herramienta svcutil.exe desde la línea de comandos con la siguiente sintaxis:

```
svcutil.exe  
https://ecommerce.microglobal.com.ar/WSMG/WSMG.asmx?WSDL
```

Esto generará un archivo de configuración y un archivo de código que contiene la clase de cliente. Agregue los dos archivos a su aplicación cliente y use la clase de cliente generada para llamar al Servicio. Por ejemplo:

C#

```
class Test  
{  
    static void Main()  
    {  
        WSMGSoapClient client = new WSMGSoapClient();  
  
        // Usar la variable variable 'client' para llamar a operaciones de servicio.  
        client.GetCatalog(cliente, usuario, password);  
  
        // Cierro el cliente.  
        client.Close();  
    }  
}
```

Visual Basic

```
Class Test  
    Shared Sub Main()  
        Dim client As WSMGSoapClient = New WSMGSoapClient()  
        ' Usar la variable variable 'client' para llamar a operaciones de servicio.  
        client.GetCatalog(cliente, usuario, password)  
  
        ' Cierro el cliente.  
        client.Close()  
    End Sub  
End Class
```

Por otro lado, si dispone de una aplicación web puede agregar una referencia de servicio para poder consumirlo:

Este es el proceso mediante el cual un cliente localiza un servicio web y obtiene su descripción del servicio.

El objetivo localizar el WSDL.

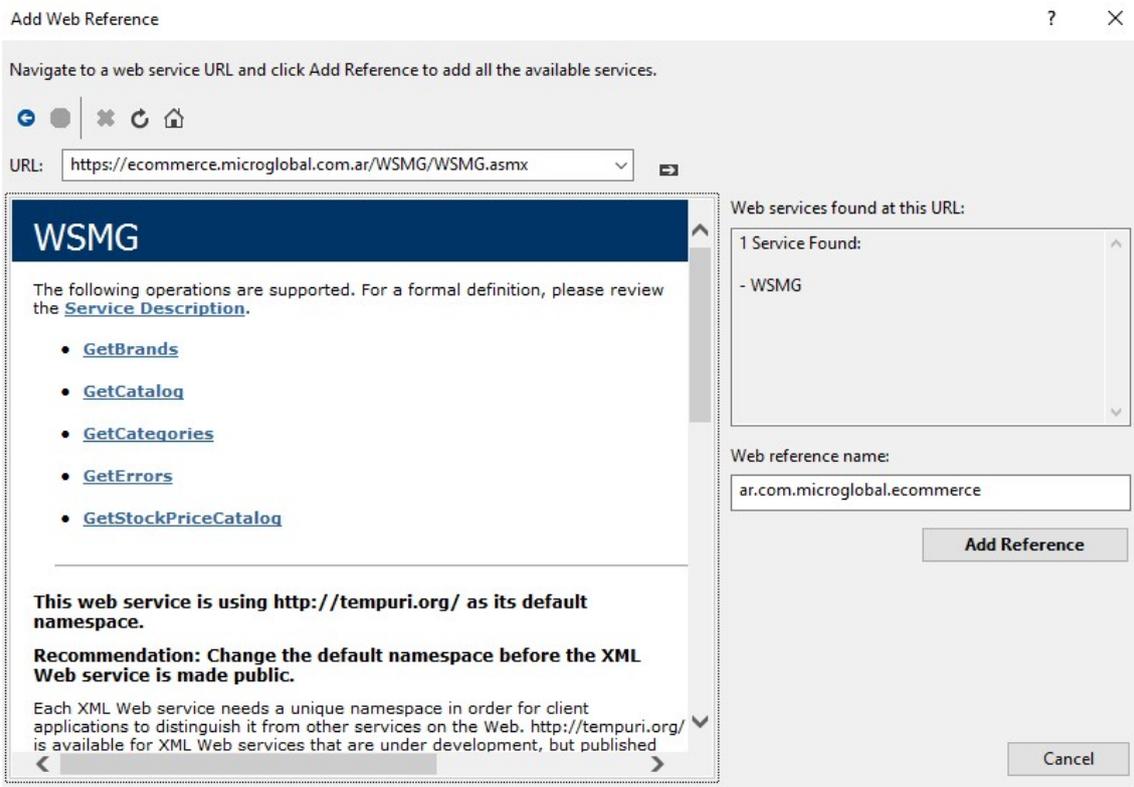
Al agregar una referencia web a su proyecto para el servicio web, se genera una clase proxy que interactúa con el servicio web y proporciona una representación local del servicio web.

En el menú Proyecto, haga clic en Agregar referencia de servicio.

A continuación, ingrese la URL de servicio, haga clic en IR, espere a la carga y luego aceptar.

Si dispone de tecnología .NET 4.0, o superior haga lo siguiente:

En el cuadro de diálogo Agregar referencia de servicio, haga clic en Avanzado y en Agregar Referencia Web. Luego, introduzca en URL el servicio, busque el servicio.



En el cuadro de diálogo Configuración de referencia de servicio, haga clic en Agregar referencia web. Obteniendo como resultado algo similar con lo que luego, podrá utilizar las operaciones de servicio con el ejemplo mencionado anteriormente.