

QeQ

Web Service

RESTful XML

Manual de Consumo

V2.6

Índice general

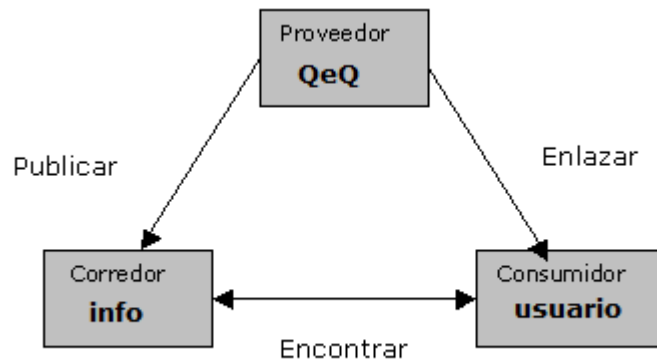
- 1.- Definiciones y documentación referente a Web Service
- 2.- Ejemplo de consumo utilizando Delphi
- 3.- Ejemplo de consumo utilizando Visual Studio (C#)
- 4.- La aplicación ejemplo

¿Qué es un web service?

Un servicio web (en inglés, Web service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

El modelo de web service de QeQ (QeQ WS).

El modelo de QeQ Web Service se basa en la arquitectura RESTful XML que describe a un consumidor y un proveedor relacionados para las operaciones de publicar, encontrar y enlazar.



Formato de entrega.

- **Al consumir el QEQ WS el resultado se presenta en una forma estándar de representación de datos.**

XML (Extensible Markup Language) es la opción para este requerimiento.

Ventajas y retos de QeQ WS.

Los web services apuntan a ser la piedra fundamental de la nueva generación de sistemas distribuidos. Éstos son algunos puntos para fundamentar esta afirmación:

- **Interoperabilidad:**
Cualquier web service puede interactuar con otro web service. Como los web services pueden ser implementados en cualquier lenguaje, los desarrolladores no necesitan cambiar sus ambientes de desarrollo para producir o consumir web services.
- **Ubicuidad:**
Los web services se comunican utilizando HTTP y XML. Por lo tanto cualquier dispositivo que soporte estas tecnologías pueden implementar o acceder web services. Muy pronto estarán presentes en teléfonos, autos e incluso máquinas expendedoras, las que avisarán a la central cuando el stock sea menor al indicado.
- **Encapsular reduce la complejidad**
Todos los componentes en un modelo de web services son web service. Lo importante es la interface que el servicio provee y no como esta implementado, por lo cual la complejidad se reduce.
- **Fácil de utilizar:**
El concepto detrás de los web services es fácil de entender, incluso existen toolkits de vendedores como IBM o Microsoft que permiten a los desarrolladores crear web services en forma rápida y fácil y Para su explotación se requiere de ser iniciado en la programación para ambientes web y aplicaciones locales.
- **Soporte de la Industria:**
Todos las empresas de software importantes soportan SOAP y REST, e incluso están impulsando el desarrollo de web services.

Lógica de conexión y consumo.

1. **Primero autenticarse**, cadena de consumo: **access**

`https://qeq.mx/datos/qws/access?var1=usuario&var2=contraseña`

Donde debe sustituir su usuario y contraseña

Nota: La contraseña se cifra en una cookie local, por lo que la invocación al mismo deberá permitir el uso de cookies y debe persistir durante toda la conexión de consulta.

IMPORTANTE: Para usar caracteres especiales (ñ, &, acentos, etc.) la cadena se debe codificar ([URL ENCODE](#)). **Ejemplo var2=Abc123#:**
`https://qeq.mx/datos/qws/access?var1=usuario&var2=Abc123%23`

2. **Buscar persona**, cadena de consumo: **pepsp**

`https://qeq.mx/datos/qws/pepsp?nombre=miguel&paterno=osorio&materno=chong`

Las variables que se pueden consultar son: nombre, paterno, materno, curp, rfc, fecha_nacimiento, issste, imss.

IMPORTANTE: Para usar caracteres especiales (ñ, &, acentos, espacios, etc.) la cadena se debe codificar ([URL ENCODE](#)).

Ejemplo nombre=maría José&paterno=peña:
`https://qeq.mx/datos/qws/pepsp?nombre=mar%C3%ADa%20jos%C3%A9&paterno=pe%C3%B1a`

3. **Buscar empresa**, cadena de consumo: **pepse**

`https://qeq.mx/datos/qws/pepse?razonsoc=abba&rfc=a`

Las variables que se pueden consultar son: razonsoc, rfc.

IMPORTANTE: Para usar caracteres especiales (ñ, &, acentos, espacios, etc.) la cadena se debe codificar ([URL ENCODE](#)). **Ejemplo razonsoc=A&B:**
`https://qeq.mx/datos/qws/pepse?razonsoc=A%26B`

4. El resultado es un XML con los atributos de la persona o empresa, limitado a los primeros 10 registros, por lo que debe afinar su búsqueda.

Ver esquema de XML:

[QeQWS.xsd](#)

Ejemplo de consumo del QeQ Web Service utilizando código de pascal objet en Borlan Delphi 10.0 Enterprice Edition

Requiere usar un componente TIdHTTP de Indy Clients con AllowCookies en True

```
Uses IdURI; //codifica el URL de consumo (URL ENCODE) con TIdURI.URLEncode()
.
.
procedure TRssForm.btnUpdateClick(Sender: TObject);
var
  strXml : string;
  I: Integer;
  Cliente: IXMLNode;
begin
  IdHTTP1.Get ('https://qeq.mx/datos/qws/access?var1='+
TIdURI.URLEncode(usr.text)+'&var2='+ TIdURI.URLEncode(pwd.text));
//var1=usuario&var2=contraseña
  if persona.Checked then strXml := IdHTTP1.Get
('https://qeq.mx/datos/qws/pepsp?nombre='+ TIdURI.URLEncode(nombre.text)+'&paterno='+
TIdURI.URLEncode(paterno.text)+'&materno='+ TIdURI.URLEncode(materno.text)+'&rfc='+
TIdURI.URLEncode(rfc.text));
  if empresa.Checked then strXml := IdHTTP1.Get
('https://qeq.mx/datos/qws/mkt?razonsoc='+ TIdURI.URLEncode(razonsoc.text)+'&rfc='+
TIdURI.URLEncode(rfc.text));
  XMLDocument1.LoadFromXML(strXml);
  XMLDocument1.Active := True;
//Leer nodo
// Cliente := XMLDocument1.DocumentElement.ChildNodes[0];
// ShowMessage( Cliente.ChildNodes[0].Text );

  XMLDocument1.SaveToFile('.\result2.xml');
  memo1.Lines.LoadFromFile('.\result2.xml');

end;
.
.
```

Ejemplo de consumo del QeQ Web Service utilizando código de C# en Visual Studio Express 2012

```
.
.
using System.Net;
using System.IO;
using System.Xml;
using System.Xml.XPath;
.
.
private void btnBuscar_Click(object sender, EventArgs e)
{
    string sURL, aURL;

    sURL = "https://qeq.mx/datos/qws/access?var1=usuario&var2=contraseña";
    aURL = "https://qeq.mx/datos/qws/pepsp?nombre=miguel&paterno=osorio&materno=chong";

    /// IMPORTANTE: Considerar función para codificar el URL de
    ///                consumo (URL ENCODE) ejemplo buscar josé peña:
    /// https://qeq.mx/datos/qws/pepsp?nombre=jos%C3%A9&paterno =pe%C3%B1a

    CookieContainer cookieQ = new CookieContainer();
    HttpRequest request = (HttpRequest)HttpRequest.Create(sURL);
    request.CookieContainer = cookieQ;

    HttpResponse response = (HttpResponse)request.GetResponse();

    request = (HttpRequest)HttpRequest.Create(aURL);
    request.CookieContainer = cookieQ;

    response = (HttpResponse)request.GetResponse();

    using (response)
    {
        XmlDocument xmlDoc = new XmlDocument();
        xmlDoc.Load(response.GetResponseStream());
        XmlNode nombre = xmlDoc.SelectSingleNode("/xml/persona/nombre");
        XmlNode paterno = xmlDoc.SelectSingleNode("/xml/persona/paterno");
        MessageBox.Show(nombre.InnerText + ' ' + paterno.InnerText); //mostrar info
    }
}
.
.
```

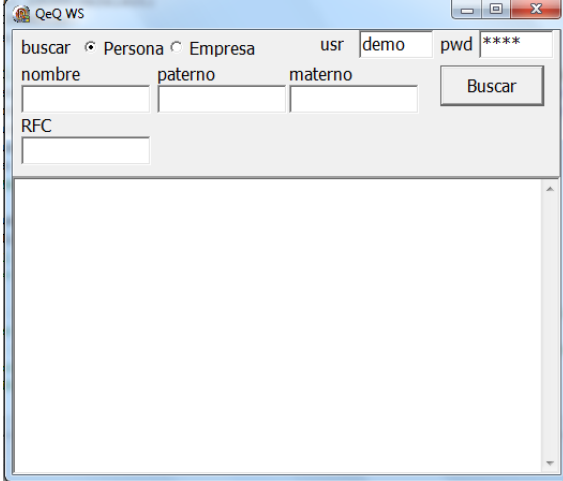
Aplicación demo para consumo

La siguiente aplicación muestra el consumo de búsqueda de información de personas y empresas

1. Ejecute QeQWS.exe

2.- seleccione el tipo de consulta que ejecutar e ingrese su usuario y contraseña

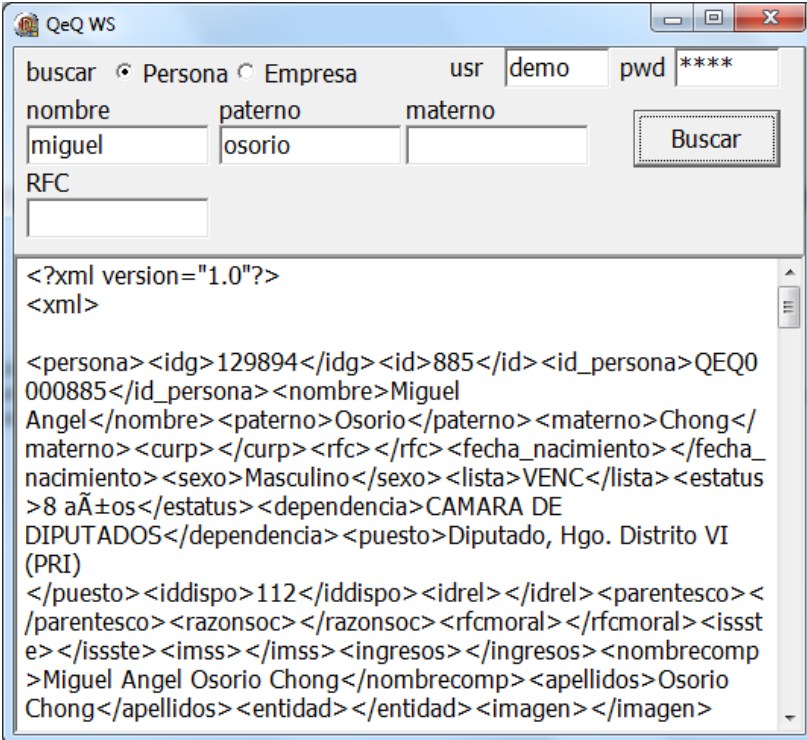
3.- llene los campos que se le solicitan respectivamente y proceda a dar clic buscar



va a

en

La acción anterior ejecutara la acción de consumo directamente del QEQ WS entregando los resultados en el estándar de etiquetas XML correspondiente, de la siguiente forma.



NOTA: Este programa genera un archivo XML en la ruta donde lanzó el ejecutable, es el mismo que se muestra en el resultado.

El archivo devuelto podrá ser interpretado de acuerdo al estándar XML.

4.-¿Qué tiene que lograr mi aplicación?:

Como ya lo explicamos anteriormente, independiente a la plataforma de desarrollo que se elija para implementar la aplicación de consumo, esta deberá lograr la cadena de consulta y conexión respectivamente como en el siguiente ejemplo:

Cadena de Autenticación ejemplo:

`https://qeq.mx/datos/qws/access?var1=usuario&var2=contrasena`

Cadena de consulta ejemplo:

`https://qeq.mx/datos/qws/pepsp?nombre=Miguel&paterno=Osorio&materno=chong`

5. Errores

Bad Request

List Bound (0)