**Como generar un XML a partir de una Clase en C#**

Publicado en [febrero 13, 2015](http://www.qualityinfosolutions.com/como-generar-un-xml-a-partir-de-una-clase-en-c/) por [Victor Riquelme](http://www.qualityinfosolutions.com/author/vriquelme/) — [Sin Comentarios ↓](http://www.qualityinfosolutions.com/como-generar-un-xml-a-partir-de-una-clase-en-c/#respond)

En esta oportunidad les mostrare una manera muy sencilla de generar un XML con la estructura que vosotros queráis a partir de una clase (objeto).

Para este ejemplo utilizaremos la siguientes clases:

**Notificacion**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | namespace QualityInfoSolutions  {      public class Notificacion      {          public Cabecera cabecera;          public string Detalle;            public Notificacion()          {          }      }  } |

Cabecera

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | namespace QualityInfoSolutions  {      public class Cabecera      {          public int IdTransaccion;      }  } |

Como podéis observar la estructura de la clase Notificación contiene un atributo de tipo Cabecera, esto lo hice para que vean que podéis crear vuestro XML con una estructura de árbol sin problemas.

Ahora les mostrare como convertir esta clase:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | ....  // Objeto utilizado para almacenar el resultado  StringWriter stringwriter = new StringWriter();    //Llenamos el objeto  Notificacion n = new Notificacion();  n.cabecera = new Cabecera()  {      IdTransaccion = 15  };  n.Detalle = "Ejemplo";    //Inicializa el serializador con el tipo Notificación  XmlSerializer x = new XmlSerializer(n.GetType());    //Convierte a XML y lo almacena en un StringWriter  x.Serialize(stringwriter, n);    //Imprimimos el resultado  MessageBox.Show(stringwriter.ToString());  .... |

Así se muestra la información como pueden ver crea la estructura del xml:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version=\"1.0\" encoding=\"utf-16\"?>  <Notificacion xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\">  <cabecera>      <IdTransaccion>15</IdTransaccion>    </cabecera>  <Detalle>Ejemplo</Detalle>  </Notificacion> |

Espero que sea de vuestra utilidad

Saludos

**Recibir XML por parámetro en Web Services C#**

Publicado en [febrero 11, 2015](http://www.qualityinfosolutions.com/recibir-xml-por-parametro-en-web-services-c/) por [Victor Riquelme](http://www.qualityinfosolutions.com/author/vriquelme/) — [Sin Comentarios ↓](http://www.qualityinfosolutions.com/recibir-xml-por-parametro-en-web-services-c/#respond)

Para poder recibir un XML por parámetros tenemos varias opciones, sin embargo en este tutorial veremos un objeto llamado XmlDocument el cual nos permitirá recibir la estructura completa, y luego podremos aprovechar las herramientas que nos provee .NET para poder leerlo de una manera muy simple.

**XML**

|  |  |
| --- | --- |
| 1  2  3  4 | <?xml version="1.0" encoding="utf-8" ?>  <Datos>     <tabla id="Prueba"/>  </Datos> |

**CÓDIGO**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | using System;  using System.Xml.Linq;  using System.Linq;  using System.Data;  using System.Web.Services;  using System.Web.Services.Protocols;  using System.Xml;  using System.Collections.Generic;    namespace WsQualityInfoSolutions  {      [WebService(Namespace = "http://qualityinfosolutions.com/")]      [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]      [System.ComponentModel.ToolboxItem(false)]        public class wsPruebaXML: System.Web.Services.WebService      {          [WebMethod]          public string GetDatosXML(XmlDocument xml)          {             string dato = "";               dato = (from row in XDocument.Parse(xml.OuterXml).Descendants("tabla")                     select row.Attribute("id").Value.ToUpper()).First();               return dato;          }      }  } |

Como podemos apreciar recibimos un objeto XmlDocument el cual es del package System.Xml y luego lo convertimos en un objeto XDocument (System.Xml.Linq), este objeto nos permite utilizar las características de Linq sobre un XML, lo cual nos hace mucho más simple la lectura.

**from row**//asignamos el alias row  
**in XDocument.Parse(xml.OuterXml).** //Conversión a XML compatible con linq  
**Descendants(“tabla”)**// Obtenemos todos los tag con nombre “tabla”  
**select row.Attribute(“id”).Value.ToUpper()** // Seleccionamos solo el atributo id de cada tag  
**).First();** // Obtenemos solo el primer resultado de la consulta

**Como enviar el XML al Ws**

Desde nuestra aplicación creamos un objeto XmlDocument y se lo enviamos al ws de la siguiente manera.

Vamos a suponer que ya agregamos la referencia web a nuestro proyecto y el import correspondiente.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public string pruebaXML() {       XmlDocument doc = new XmlDocument();     doc.Load(@"C:\\Prueba.xml");       wsPruebaXML PruebaXML = new wsPruebaXML();       return PruebaXML.GetDatosXML(doc);  } |

Esperamos sea de su utilidad

Saludos