**[Linq to xml学习笔记](http://www.cnblogs.com/babycool/p/3513065.html)**

来源:http://www.cnblogs.com/babycool/

最近需要用到操作xml文档的方法，学习了一下linq to xml，特此记录。

测试代码：

class Program

{

//参考: LINQ to XML 编程基础 - luckdv - 博客园 http://www.cnblogs.com/luckdv/articles/1728088.html

static void Main(string[] args)

{

string path = @"E:\def\5.xml";

//第一种方法 创建

//XDocument xdoc = new XDocument(new XDeclaration("1.0", "utf-8", "yes"),new XElement("nianhui",new XElement("mingdan",new XText("这里是人名单")),new XElement("jiangxiang",new XText("这里是奖项"))));

//xdoc.Save(path);

//第二种方法 创建

//XElement root = new XElement("nianhui",

// new XElement("first", "名单"),

// new XElement("second", "第二项")

// );

//root.Save(path);

//修改一个Element

//XElement rootload = XElement.Load(path);

//XElement md = rootload.Element("mingdan");

//if (md != null)

//{

// 替换一个节点

// md.ReplaceWith(new XElement("mingdan", "哈哈哈哈"));

//}

//else

//{

// 添加一个节点

// rootload.Add(new XElement("mingdan", "又建了一个"));

//}

//rootload.Save(path);

//创建根节点

//XElement root=new XElement("nianhui");

//root.Save(path);

//获取Value值

//string x= GetElementValue(path,"mingda");

//Console.WriteLine(x);

//删除节点

//DelElement(path, "mingdan");

//XElement root = XElement.Load(Xpath);

//XElement element = root.Element(Xname);

//if (element != null)

//{

// element.Remove();

//}

////保存修改 操作完成后一定要将修改保存回去

//root.Save(Xpath);

//为一级节点添加子节点

//XElement root = XElement.Load(path);

//XElement element = root.Element("xing");

//element.Add(new XElement("heihei","what"));

//root.Save(path);

//------------带属性的xml----------------------------------------------

//创建带属性的xml节点

//XElement xdoc = new XElement("root", new XElement("first",new XAttribute("ID","001")),new XElement("second","行内的value"));

//xdoc.Save(path);

//创建带多个属性的xml节点

//XElement xdoc = new XElement("root",new XElement("first",new XAttribute("name","002"),new XAttribute("path","abc.txt")));

//xdoc.Save(path);

//更新某属性的值

//XElement xdoc = XElement.Load(path);

//获取某节点元素

//XElement x = xdoc.Element("first");

//string x2 = x.Attribute("path").Value;

//Console.WriteLine(x2);

//更新某属性的值

//XElement x3 = xdoc.Element("first");

//if (x3 != null)

//{

// XAttribute a1 = x3.Attribute("path");

// if (a1 != null)

// {

// a1.SetValue("wohuanle");

// }

//}

//xdoc.Save(path);

//获取某属性的值

//XElement xdoc = XElement.Load(path);

//XElement x4 = xdoc.Element("first");

//if (x4 != null)

//{

// XAttribute a2 = x4.Attribute("path");

// if (a2 != null)

// {

// Console.WriteLine(a2.Value);

// }

// else

// {

// Console.WriteLine("");

// }

//}

//为某节点添加属性

//XElement xdoc = XElement.Load(path);

//XElement x4 = xdoc.Element("first");

//if (x4 != null)

//{

// x4.SetAttributeValue("hah", "what");

//}

//xdoc.Save(path);

//修改某节点的属性 如果不存在该属性则添加

//XElement xdoc = XElement.Load(path);

//XElement x4 = xdoc.Element("mingdan");

//if (x4 != null)

//{

// XAttribute attr = x4.Attribute("number");

// if (attr != null)

// {

// //存在该属性

// attr.SetValue("woshixinzengjiade");

// }

// else

// {

// //不存在该属性

// x4.SetAttributeValue("number", "meizhegeshuxing");

// }

//}

//xdoc.Save(path);

//尝试获取xml文档中任意一个节点 但测试不成功

//XElement xdoc = XElement.Load(path);

////XElement x4 = xdoc.Element("mingdan");

//IEnumerable<XElement> element = from e in xdoc.Elements()

// where e.Elements().Any(c=>c.Name=="xing")

// select e;

//foreach (XElement it in element)

//{

// Console.WriteLine( it.Value);

//}

//为某一级节点添加属性

//Dictionary<string,string> ss=new Dictionary<string,string>();

//ss.Add("na","01");ss.Add("nb","02");

//CreateAttr(path, "heihei",ss);

//删除某一级子节点的属性

XElement xdoc = XElement.Load(path);

XElement x4 = xdoc.Element("mingdan");

if (x4 != null)

{

if (x4.Attribute("number") != null)

{

x4.Attribute("number").Remove();

}

}

xdoc.Save(path);

Console.ReadKey();

}

#region 为某一级节点添加属性

/// <summary>

/// 为某一级节点添加属性

/// </summary>

/// <param name="Xpath">文件路径</param>

/// <param name="Xname">一级节点名称</param>

/// <param name="Attr">属性集</param>

static void CreateAttr(string Xpath, string Xname, Dictionary<string, string> Attr)

{

XElement xdoc = XElement.Load(Xpath);

XElement x4 = xdoc.Element(Xname);

if (x4 != null)

{

foreach (var item in Attr)

{

x4.Add(new XAttribute(item.Key, item.Value));

}

}

xdoc.Save(Xpath);

}

#endregion

#region 初始化根节点为Xroot的xml文档

/// <summary>

/// 初始化根节点为Xroot的xml文档

/// </summary>

/// <param name="Xpath"></param>

/// <param name="Xroot"></param>

void InitElement(string Xpath, string Xroot)

{

XElement root = new XElement(Xroot);

root.Save(Xpath);

}

#endregion

#region 创建或更新节点

/// <summary>

/// 创建或更新节点

/// </summary>

/// <param name="Xpath"></param>

/// <param name="Xname"></param>

/// <param name="XValue"></param>

void CreateElement(string Xpath, string Xname, string XValue)

{

//加载xml文档路径

XElement root = XElement.Load(Xpath);

//加载某个名称为name的节点

XElement first = root.Element(Xname);

if (first != null)

{

//该名称的节点已存在，替换

first.ReplaceWith(new XElement(Xname, XValue));

}

else

{

//该名称的节点不存在，创建

root.Add(new XElement(Xname, XValue));

}

//不要忘记保存修改

root.Save(Xpath);

}

#endregion

#region 获取某节点的Value值

/// <summary>

/// 获取某节点的Value值

/// </summary>

/// <param name="Xpath"></param>

/// <param name="Xname"></param>

/// <returns></returns>

static string GetElementValue(string Xpath, string Xname)

{

XElement root = XElement.Load(Xpath);

XElement element = root.Element(Xname);

if (element != null)

{

return element.Value;

}

else

{

return "";

}

}

#endregion

#region 删除Name为Xname的子节点

/// <summary>

/// 删除Name为Xname的子节点

/// </summary>

/// <param name="Xpath"></param>

/// <param name="Xname"></param>

static void DelElement(string Xpath, string Xname)

{

XElement root = XElement.Load(Xpath);

XElement element = root.Element(Xname);

if (element != null)

{

element.Remove();

}

//保存修改

root.Save(Xpath);

}

#endregion

}

2014-1-10 10:08:37 新增加一个：

/// <summary>

/// 创建或添加一级节点（1个或多个）

/// </summary>

/// <param name="Xpath"></param>

/// <param name="Xelement"></param>

static void AddElement(string Xpath, Dictionary<string, string> Xelement)

{

//加载xml文档路径

XElement root = XElement.Load(Xpath);

foreach (var e in Xelement)

{

//判断当前一级节点是否存在，若存在，则重新设置值，若不存在，则添加

XElement first = root.Element(e.Key);

if (first != null)

{

first.SetValue(e.Value);

}

else

{

root.SetElementValue(e.Key, e.Value);

}

}

root.Save(Xpath);

}

参考：

LINQ to XML 编程基础luckdv博客园  http://www.cnblogs.com/luckdv/articles/1728088.html