# 在学习LinQ之前,我们先来熟悉一下c#操作xml的方法

## 1创建xml实例并且保存到文件

|  |
| --- |
| static void CreateXMLTree()  {  XElement stu1 = new XElement("Student", new XElement("Name", "Jack"), new XElement("Age", 18),new XElement("Gender", "Male"));  XElement stu2 = new XElement("Student", new XElement("Name", "Mary"), new XElement("Age", 16),new XElement("Gender", "Female"));  XElement stu3 = new XElement("Student", new XElement("Name", "Jesse"), new XElement("Age", 19),new XElement("Gender", "Female"));  XElement stu4 = new XElement("Student", new XElement("Name", "Betty"), new XElement("Age", 17),new XElement("Gender", "Female"));  XElement xtree = new XElement("Students", stu1, stu2, stu3, stu4);  Console.WriteLine(xtree);  xtree.Save("students.xml");  } |

### 运行这个函数,就会在Debug文件夹下面生成一个students.xm文件,内容如下

|  |
| --- |
| **<?xml version="1.0" encoding="utf-8"?>**  **<Students>**  **<Student>**  **<Name>Jack</Name>**  **<Age>18</Age>**  **<Gender>Male</Gender>**  **</Student>**  **<Student>**  **<Name>Mary</Name>**  **<Age>16</Age>**  **<Gender>Female</Gender>**  **</Student>**  **<Student>**  **<Name>Jesse</Name>**  **<Age>19</Age>**  **<Gender>Female</Gender>**  **</Student>**  **<Student>**  **<Name>Betty</Name>**  **<Age>17</Age>**  **<Gender>Female</Gender>**  **</Student>**  **</Students>** |

## 2.读取xml文件

|  |
| --- |
| static void LoadXMLTree()  {  XElement xtree =XElement.Load(Environment.CurrentDirectory+"\\students.xml");  IEnumerable<XElement> stus = xtree.Descendants("Student");  foreach (var item in stus)  {  Console.WriteLine(item);  }  } |

### 效果如下

|  |
| --- |
|  |

## 也可以获取每一个标签的值并且拼接显示,这是改进后的函数

|  |
| --- |
| static void LoadXMLTree()  {  XElement xtree =XElement.Load(Environment.CurrentDirectory+"\\students.xml");  IEnumerable<XElement> stus = xtree.Descendants("Student");  foreach (var item in stus)  {  //Console.WriteLine(item);  IEnumerable<XNode> nodes = item.Nodes();  foreach (var node in nodes)  {  XElement xe = (XElement)node;  Console.Write(xe.Name+":"+ xe.Value+" ");  }  Console.WriteLine();  }  } |

### 效果如下

|  |
| --- |
|  |

# 每一个Student里面的标签元素的值,也可以使用XmlDocument类

|  |
| --- |
| static void LoadXMLTreeXMLDoc()  {  XmlDocument doc = new XmlDocument();  doc.Load(Environment.CurrentDirectory + "\\students.xml");  XmlNode xe = doc.SelectSingleNode("Students");//获取根节点  XmlNodeList nlist = xe.ChildNodes;  foreach (XmlNode n in nlist)  {  XmlNodeList attrlist = n.ChildNodes;  Console.WriteLine($"Name:{attrlist[0].InnerText},Age:{attrlist[1].InnerText}," +  $"Gender:{attrlist[2].InnerText}");    }  } |

# 当然还有使用DataSet的方法和使用Json.net库的方法,有需要再去研究