Efs Frame
Enterprise Develop Faster & Stronger

EfsFrame .Net 模型实例教程文档

1、学习 Efs 框架的前提

Efs 框架是一套完整的企业级开发应用平台，不是一套入门级编程教程，所以对学习 Efs 框架的开发人员有一定的要求，具体如下:

1、有一定 B/S 架构项目开发经验;

2、作为平台的 Web 表现层开发人员，要熟悉 HTML,JavaScript,了解 XML 编程，熟悉一种 Asp、C#、Jsp 至少一种表现层开发脚本语言，适当了解数据库设计原理;

3、作为业务逻辑层组件开发人员，至少熟悉一种高级编程语言(如: pascal，VB，C#，C++、Java 等)，熟悉 XML 编程，熟悉数据设计库设计原理(对表设计、视图、存储过程、自定义函数等有一定的认识)。

2、基于 Efs 框架开发实例分析

为了能让大家更好的学习和理解 Efs 框架，下面我们以一个简单的实例开始我们的 Efs 学习之旅。

本实例简单以 Aspx + C# + Sql Server 2005 模型完成。

2.1、项目描述

很多编程实例都是 Hello Word! 开始，可谓简单而又经典。

我们将以一个简单项目入手，项目虽然简单，但是我们的重点是通过这个小项目迅速了解 Efs 的整个使用流程，体会 Efs 框架的开发优点。

项目名称《学生档案管理》，简单对学生的基本信息进行增加、修改、删除、查询。

3、业务表设计及事务事件设计

3.1、表设计:

表名称: PERSON

字段描述	字段编码	数据类型	主键	是否为空	备注
学生编码	PERSONID	VARCHAR(10)	Y	N	系统自动编码2位年+2位月+6位顺序码
学生姓名	NAME	VARCHAR(60)	N	N	
身份证号	IDCARD	VARCHAR(18)	N	Y	
性别	SEX	VARCHAR(1)	N	Y	字典 DIC_SEX
出生日期	BIRTHDAY	DATETIME	N	Y	
籍贯	PLACECODE	VARCHAR(6)	N	Y	字典 DIC_CODE
年龄	YEAROLD	INT	N	Y	
联系电话	TEL	VARCHAR(60)	N	Y	
邮箱	EMAIL	VARCHAR(60)	N	Y	
备注	BAK	NTEXT	N	Y	

Ms Sql 脚本

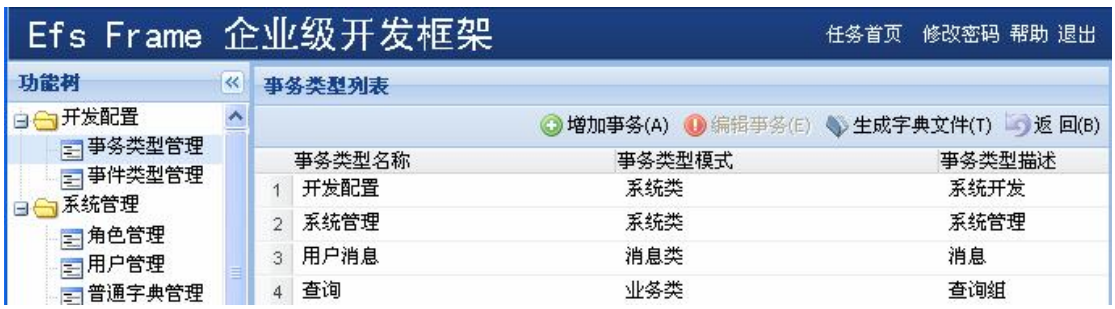
```
CREATE TABLE [PERSON] (
  [PERSONID] varchar(10) NOT NULL,
  [NAME] varchar(60),
  [IDCARD] varchar(18),
  [SEX] varchar(1),
  [BIRTHDAY] datetime,
  [YEAROLD] int,
  [TEL] varchar(50),
  [EMAIL] varchar(60),
  [BAK] varchar(4000),
  [PLACECODE] varchar(6),
```

```
  CONSTRAINT [PK_PERSON] PRIMARY KEY CLUSTERED ([PERSONID])

)

ON [PRIMARY]

GO
```

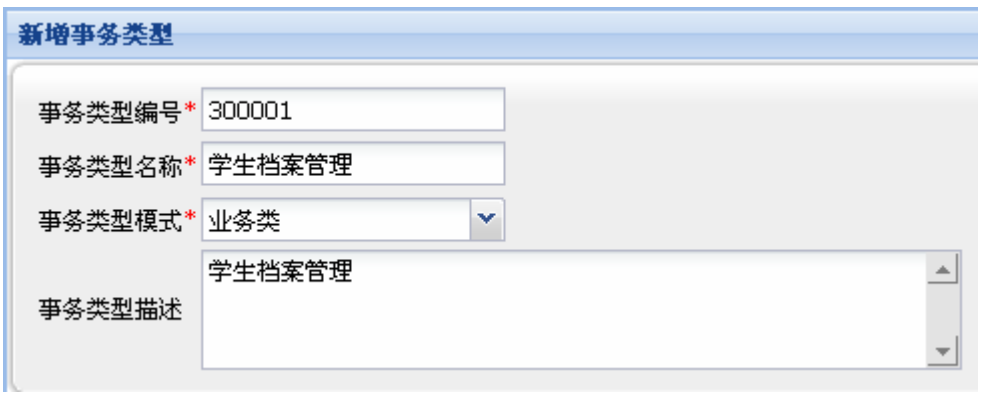
3.2、事务事件设计

事务编号	300001	事务名称	学生档案管理	
	事件名称	事件编码	操作 URL	备注
	添加学生	300101	person/psnAdd.aspx	
	管理学生信息	300102	person/qryPsnList.aspx	

新增事务如下：在事务列表界面，点击“增加事务”



根据事务事件设计，输入新增事务类型信息：

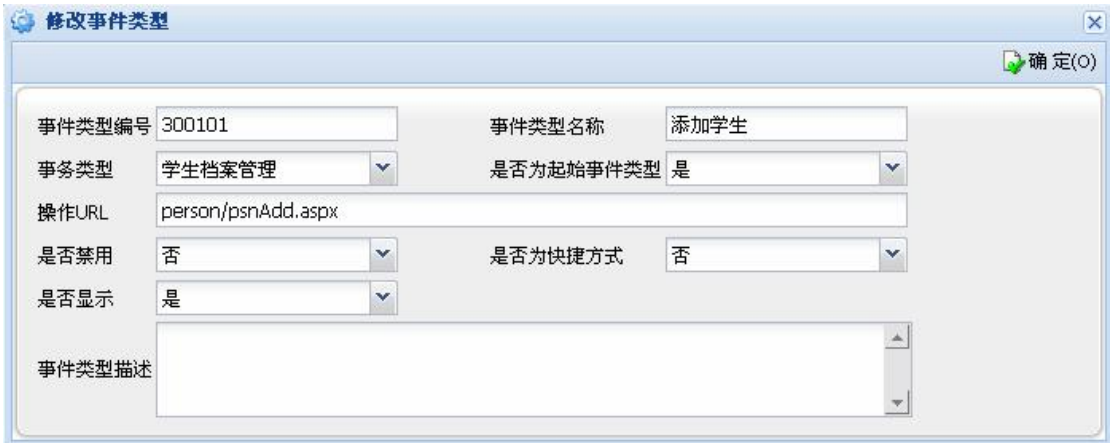


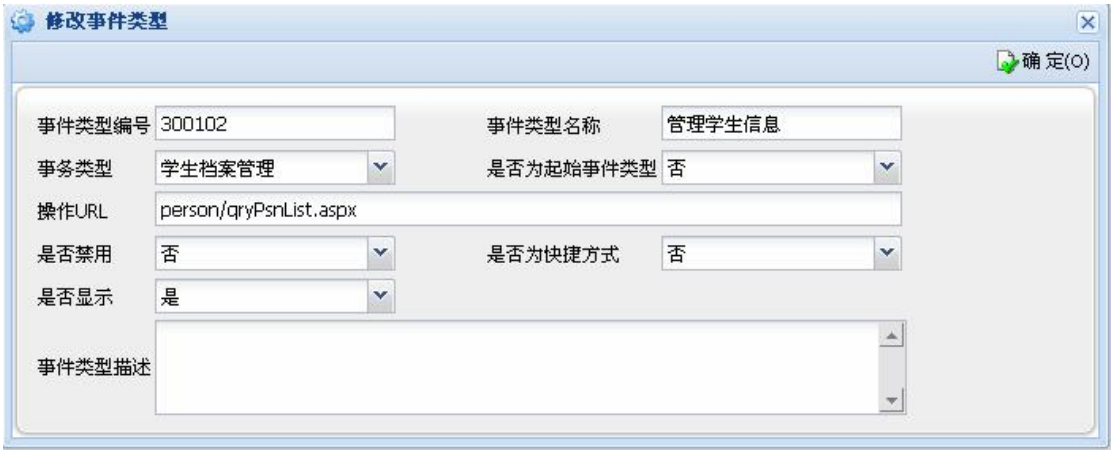
根据事务事件设计，输入新增事件类型信息：

备注：1、事件属于事务的下一级，比如事件“添加学生”，就属于事务“学生档案管理”。当添加一个事务之后，就会生成“事务类型字典”，添加事件的时候需要在“事务类型字典”里面选择所属事务。

2、“操作 URL”里面录入的信息，表示“添加学生”页面的地址。当用户点

击“添加学生”时，就链接到这个页面。

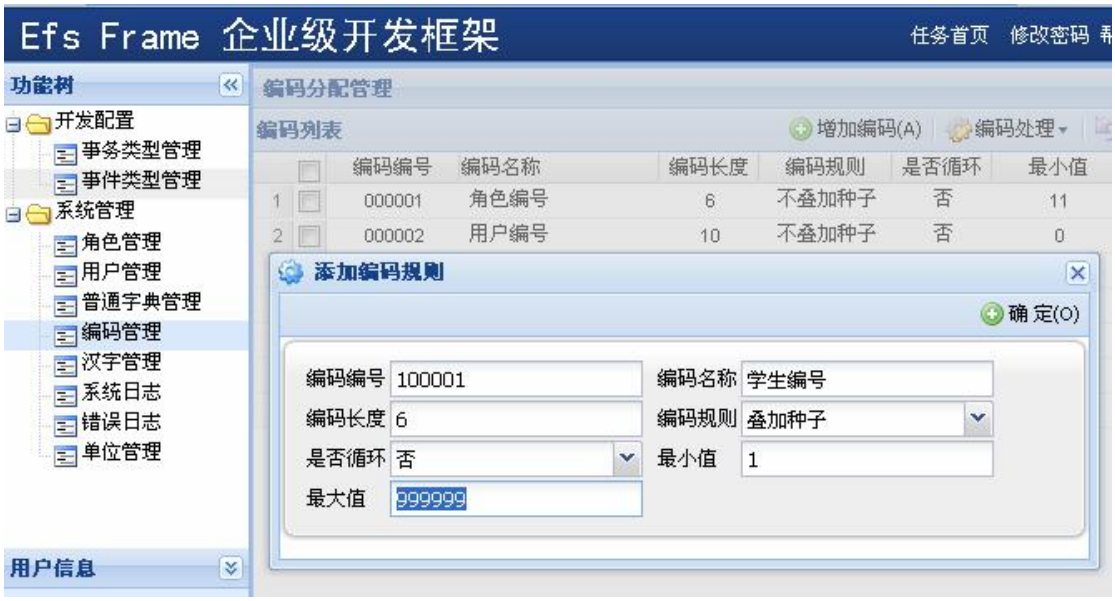




3.3、编码规则设计

编码编号	编码名称	编码长度	编码规则	是否循环	最小值	最大值
100001	学生编号	**6**	叠加种子	否	**1**	**999999**

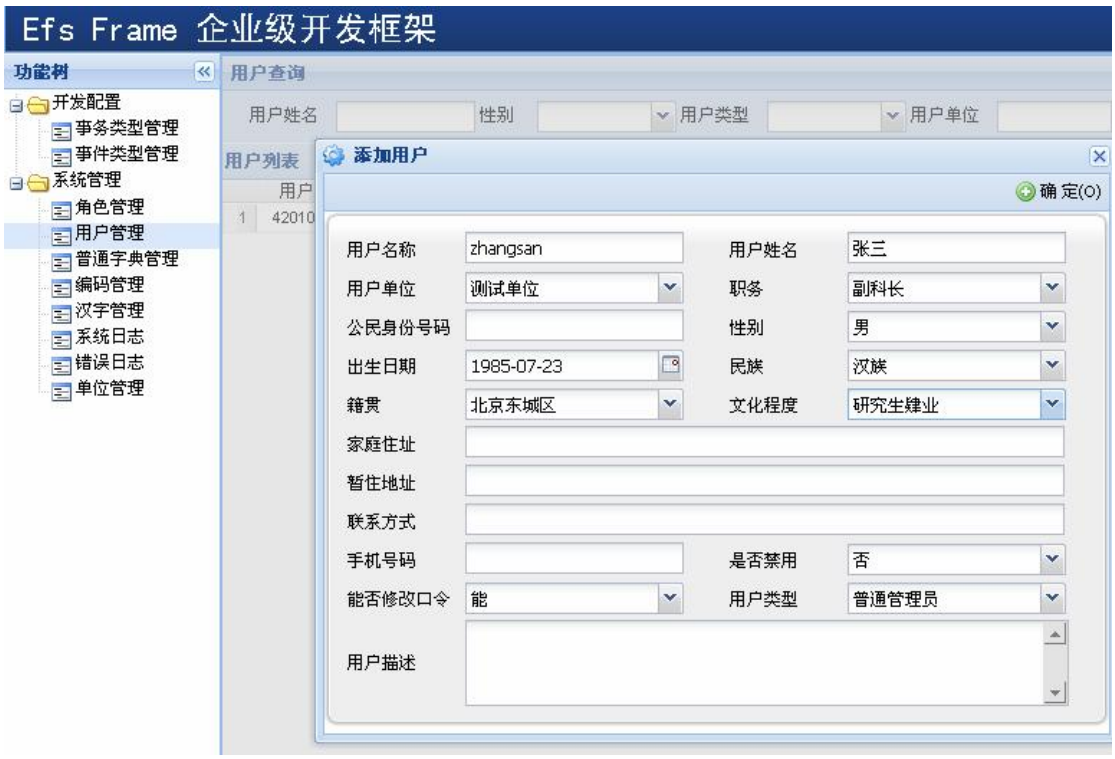
添加编码规则的意义在于，根据规则自动生成新的序号，比如“学生编号”，从 **000001** 开始，再次添加学生的时候序号自动变成 **000002**，以此类推。



4、添加用户、角色及角色权限

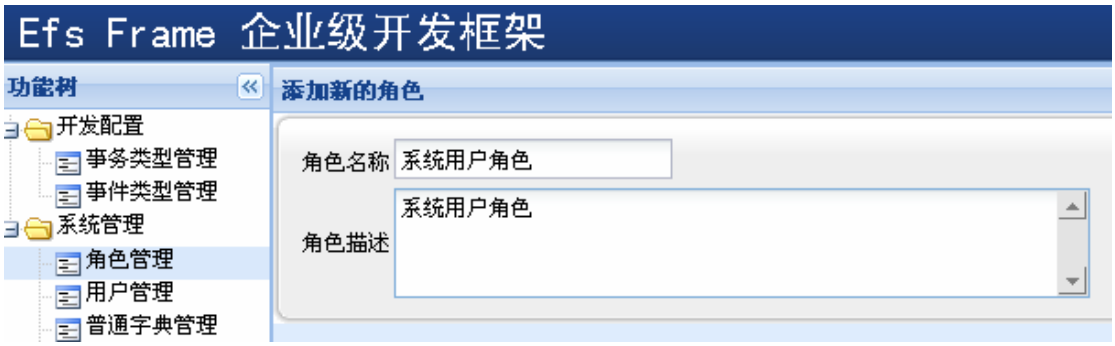
4.1、添加用户

这里添加系统用户，表示这个用户可以登录系统，根据不同角色赋有的权限，登陆之后就可以获取相应的权限，对系统进行操作。



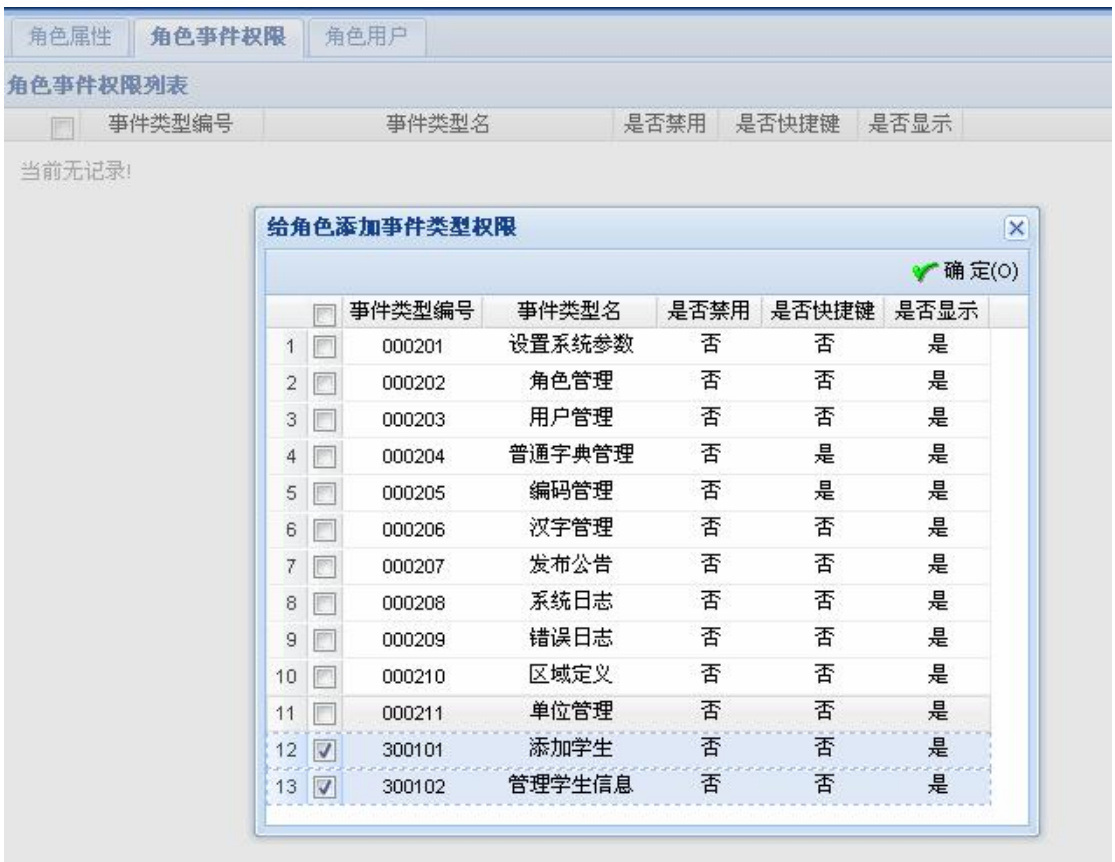
4.2、添加角色

角色这个概念很重要，表明是一定的权限，哪个角色有哪些权限，这个都需要管理员进行配置，我们现在新增一个角色“系统用户角色”。

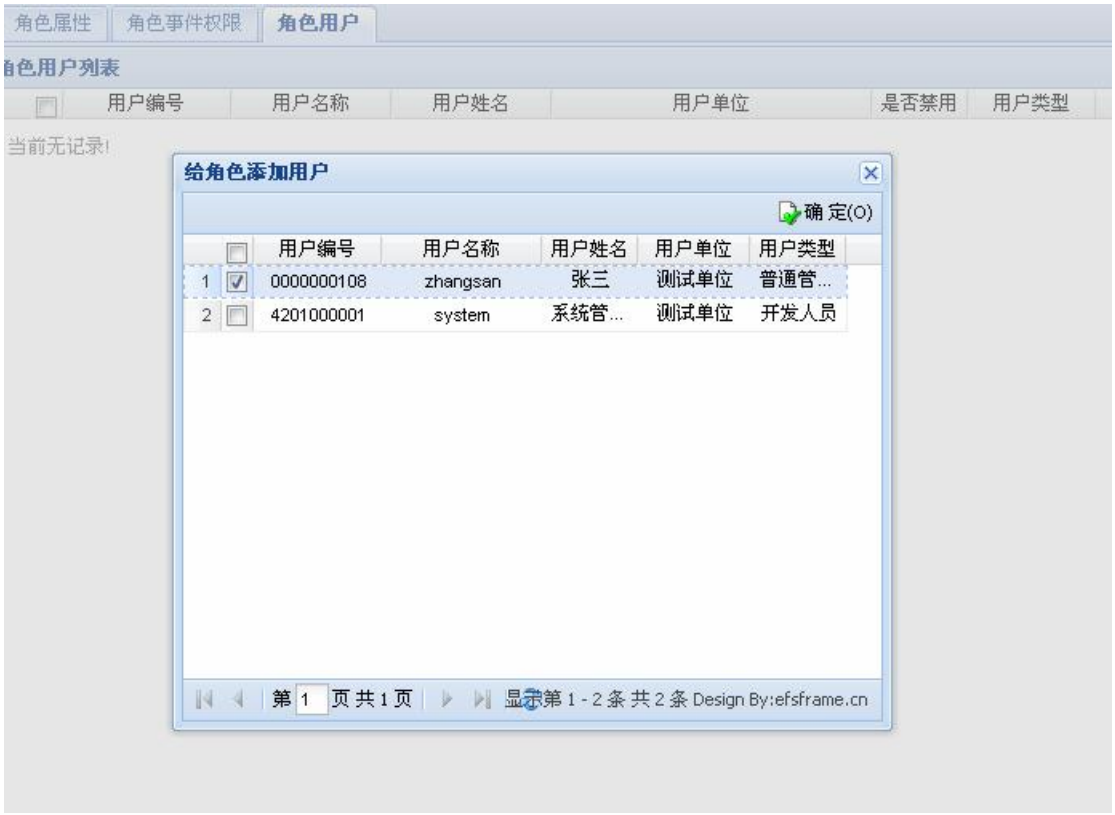


4.3、给角色添加权限

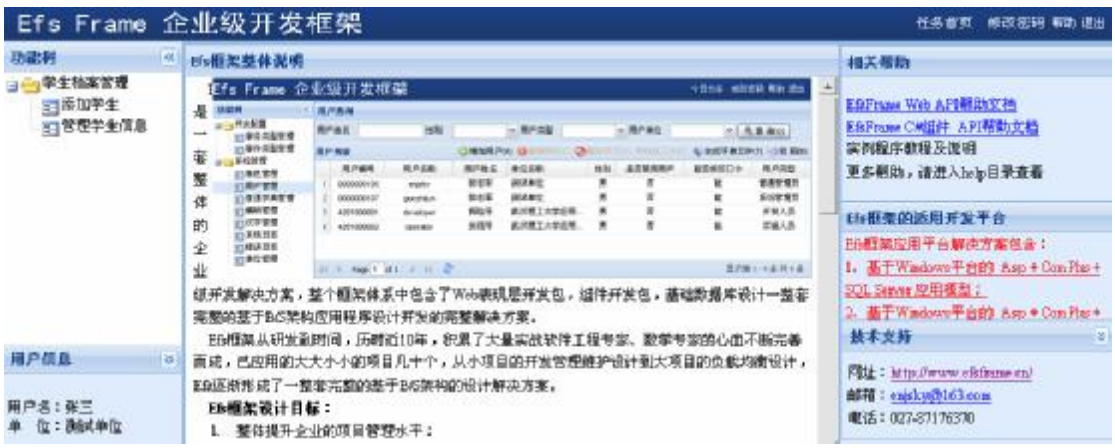
在列表里，选中系统用户角色”，进行编辑，添加事件类型权限。我们给这个角色添加两个事件，也就是功能“添加学生”与“管理学生信息”。



对该角色添加用户就是角色用户，只要属于这个角色里的用户，那么就拥有这个角色所拥有的所有权限。我们将“张三”这个用户添加到这个角色里面，那么“张三”就有了这个角色里的权限“添加学生”与“管理学生信息”。



用张三登陆，左边功能树里就有该角色所赋有的权限，登陆界面如下:



5、功能模块开发

5.1、添加学生

5.1.1、添加学生页面源代码

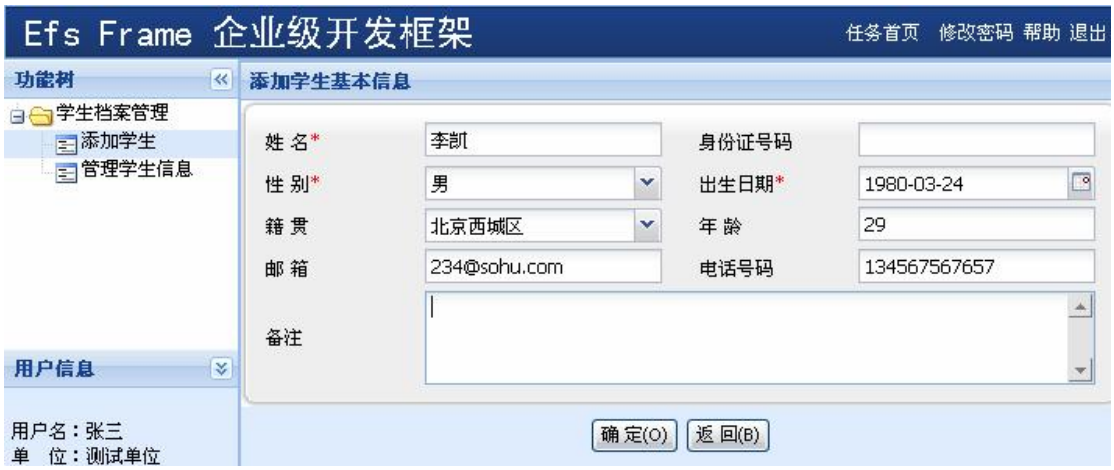
添加页面路径及名称: efs\person\ psnAdd.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="psnAdd.aspx.cs" Inherits="person_psnAdd" %>

<!--#include file="../checkLog.inc" -->
<!--
//*******************************
//** 设计人员:     Enjsky
//** 设计日期:     2009-10-28
//** 联系邮箱:     enjsky@163.com
//*******************************
-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML XMLNS:ELEMENT>
<head>
<title>添加学生基本信息</title>
<link rel="stylesheet" type="text/css" href="../css/ext-all.css" />
<link rel="stylesheet" type="text/css" href="../css/efs-all.css" />
<script type="text/javascript" src="../js/efs-all.js"></script>

<SCRIPT language="JavaScript">
<!--
function doRet()
{
  location.href = "qryPsnList.aspx";
}

// 提交信息
function doSubmit()
{
  Efs.getExt("frmData").submit();
}

// 获取异步提交的返回监听函数
function frmPostSubBack(bln,from,action)
{
  if(bln){
    location.href = "qryPsnList.aspx";
  }
```

```
  else
  {
    var xml_http = action.response;
    var objXML = xml_http.responseXML;

    alert("提交失败：" + objXML.selectSingleNode("//FUNCERROR").text);
    objXML = null;
    xml_http = null;
  }
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<div xtype="panel" iconCls="icon-panel" title="添加学生基本信息" border="false" buttonAlign="center"
autoScroll="true">
  <form id="frmData" class="efs-box" method="post" url="../sysadmin/baseRefWeb.aspx?method=PsnAdd"
method="post" onEfsSuccess="frmPostSubBack(true)" onEfsFailure="frmPostSubBack(false)">
      <TABLE class="formArea">
        <TR>
          <TD width="100" labelFor="name">姓　名</TD>
          <TD><INPUT id="name" type="text" kind="zhunicode" must="true" maxlength="50"
fieldname="PERSON/NAME" datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">身份证号码</TD>
          <TD><INPUT type="text" kind="idcard" fieldname="PERSON/IDCARD" sex="sex" birthday="birthday"
datatype="0" state="0"></TD>
        </TR>
        <TR>
          <TD width="100" labelFor="sex">性　别</TD>
          <TD><INPUT type="text" kind="dic" src="DIC_SEX" id="sex" fieldname="PERSON/SEX" must="true"
datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100" labelFor="birthday">出生日期</TD>
          <TD><INPUT type="text" kind="date" id="birthday" fieldname="PERSON/BIRTHDAY" datatype="3"
state="0" must="true"></TD>
        </TR>
        <TR>
          <TD width="100">籍　贯</TD>
          <TD><INPUT type="text" kind="dic" src="DIC_CODE" fieldname="PERSON/PLACECODE" datatype="0"
state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">年　龄</TD>
          <TD><INPUT type="text" kind="int" range=[0,119] fieldname="PERSON/YEAROLD" datatype="1"
```

```
state="0"></TD>
        </TR>
        <TR>
          <TD width="100">邮  箱</TD>
          <TD><INPUT type="text" kind="email" fieldname="PERSON/EMAIL" datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">电话号码</TD>
          <TD><INPUT type="text" kind="mask" mask="###-########" fieldname="PERSON/TEL" datatype="0"
state="0"></TD>
        </TR>
        <tr>
          <td>备注</td>
          <td colspan="4"><TEXTAREA class="Edit" kind="text" style="height:60px;width:430px"
fieldname="PERSON/BAK" state="0" datatype="0"></TEXTAREA>
          </td>
        </tr>
        <INPUT type="hidden" kind="text" fieldname="PERSON/PERSONID" datatype="0" state="0" operation="0"
writeevent="0" ><!--operation="0"  定义为添加接口-->
      </TABLE>
    </form>
     <div xtype="buttons">
        <div text="确  定" onEfsClick="doSubmit()"></div>
        <div text="返  回" onEfsClick="doRet()"></div>
  </div>
</div>

</BODY>
</HTML>
```

界面简单截图



5.1.2、添加学生组件接口方法源代码

Person.cs 里添加 addNew 方法，作用是处理添加学生信息，与学生相关的一些信息都在这个类里，包括添加学生、修改/删除学生、查询学生列表、查询学生详细信息等。

添加学生组件路径及名称: namespace Efsframe.cn.person.Person

```
/// <summary>
    /// 添加学生档案信息
    /// </summary>
    /// <param name="strXml">XML 数据信息</param>
    /// <returns>XML 返回信息</returns>
    public static string addNew(string strXml)
    {
      DataDoc doc = new DataDoc(strXml);
      // 创建数据层执行对象
      DataStorage storage = new DataStorage();
      // 创建标准返回结构Dom类对象
      ReturnDoc returndoc = new ReturnDoc();
      try
      {
        int size = doc.getDataNum(Table.PERSON);

        // 解析sql语句
        for (int i = 0; i < size; i++)
        {
          XmlElement ele = (XmlElement)doc.getDataNode(Table.PERSON, i);

          //A.001~~~~
          // 为PersonID分配唯一编码
          XmlNode node = ele.SelectSingleNode(Field.PERSONID);
          string strId = NumAssign.assignID_B("100001", General.curYear2() +
General.curMonth());
          node.InnerText = strId;
          //end A.001~~~~

          storage.addSQL(SQLAnalyse.analyseXMLSQL(ele));
        }
        // 执行SQL
        string strReturn = storage.runSQL();
        if (!General.empty(strReturn))
        {
          // 执行失败，返回异常描述
          returndoc.addErrorResult(Common.RT_FUNCERROR);
```

```
            returndoc.setFuncErrorInfo(strReturn);
        }
        else
            // 执行成功，返回成功节点
            returndoc.addErrorResult(Common.RT_SUCCESS);
    }
    catch (Exception e)
    {
        // 发生异常，返回异常描述
        returndoc.addErrorResult(Common.RT_FUNCERROR);
        returndoc.setFuncErrorInfo(e.Message);
    }
    // 标准的返回XML结构文档
    return returndoc.getXml();
}
```

在 baseRef.cs 中添加方法

在这里，获取页面传递过来的 xml，然后调用相应方法进行处理。

反射类路径及名称： namespace Efsframe.cn.baseCls.baseRef

```
/// <summary>
/// 添加学生
/// </summary>
/// <param name="Request"></param>
/// <param name="Response"></param>
/// <returns></returns>
public string PsnAdd(HttpRequest Request, HttpResponse Response)
{
    string strXml = Request["txtXML"];

    UserSession userSession = ((UserSession)Session["RoleUser"]);

    strXml = PageCommon.setOpDocXML(strXml, userSession);

    return Person.addNew(strXml);
}
```

5.2、查询学生列表

5.2.1、查询学生列表页面源代码

查询学生列表页面路径及名称： efs\person\ qryPsnList.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="qryPsnList.aspx.cs" Inherits="person_qryPsnList"
%>
<!--#include file="../checkLog.inc" -->
<!--
//*******************************
//** 设计人员:     Enjsky
//** 设计日期:     2009-10-28
//** 联系邮箱:     enjsky@163.com
//*******************************
-->
<HTML>
<head>
<title>查询学生列表</title>
<link rel="stylesheet" type="text/css" href="../css/ext-all.css" />
<link rel="stylesheet" type="text/css" href="../css/efs-all.css" />
<script type="text/javascript" src="../js/efs-all.js"></script>

<SCRIPT language="JavaScript">

var g_XML = Efs.Common.getQryXml();

var sPersonID = "";
function doGridClick(data){
     sPersonID = data["PERSONID"]
     if(sPersonID != ""){
          Efs.getExt("cmdEdit").enable();
    Efs.getExt("cmdDel").enable();
     }
}

// 进入查询
function doQry()
{
  var strXml = Efs.Common.getQryXml(Efs.getExt("frmQry"));
  Efs.getDom("psnList").txtXML = strXml;
  Efs.getExt("psnGrid").store.load();
}

// 修改人员档案
function onEditEx() {

  if(sPersonID == "")
  {
    alert("没有选择学生");
```

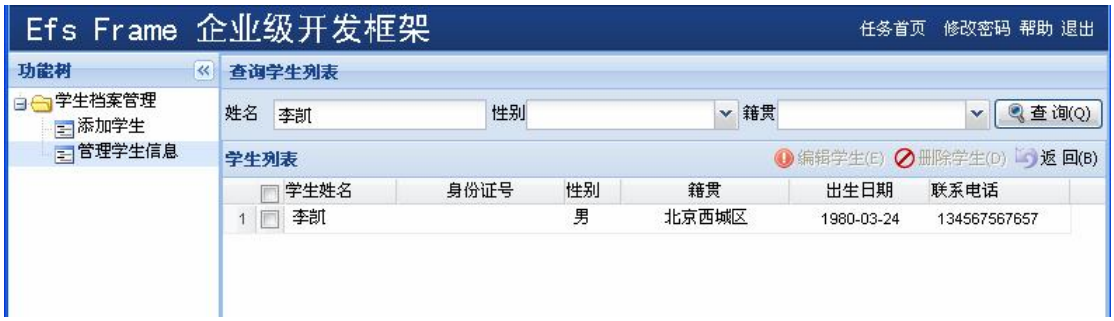
```
    return false;
  }
  Efs.getExt("frmData").reset();
  var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
  xmlhttp.Open("POST","../sysadmin/baseRefWeb.aspx?method=QryPersonDetail&txtPersonID=" +
sPersonID,false);
  xmlhttp.Send();
  var xmlReturnDoc = new ActiveXObject("MSXML2.DOMDocument");
  xmlReturnDoc = xmlhttp.responseXML;
  Efs.Common.setEditValue(xmlReturnDoc.xml,Efs.getExt("frmData"), "QUERYINFO");
  xmlReturnDoc = null;
  xmlhttp = null;

  Efs.getExt("PsnMWin").show();
}

// 提交修改人员信息
function doPsnEdit() {
  Efs.getExt("frmData").submit();
}


// 获取异步提交的返回监听函数
function frmPostSubBack(bln,from,action)
{
  if(bln)
  {
    Efs.getExt("PsnMWin").hide();
    doQry();
  }
  else
  {
    var xml_http = action.response;
    var objXML = xml_http.responseXML;
    alert("处理失败：" + objXML.selectSingleNode("//FUNCERROR").text);
    objXML = null;
    xml_http = null;
  }
}

// 删除人员信息
function onDelEx()
{
  Efs.getExt("frmData").submit(Efs.getExt("psnGrid").getDelXml());
```

```
}

</SCRIPT>
</HEAD>
<BODY>
<div iconCls="icon-panel" region="north" height="60" title="查询学生列表" border="false">
 <form id="frmQry"  method="post">
  <TABLE class="formAreaTop" width="100%" height="100%" cellpadding="0" cellspacing="0">
      <tr>
        <td> </td>
        <td width="60">姓名</td>
        <td width="160"><input type="text" class="Edit" kind="text" fieldname="NAME" operation="like"
maxlength="30" hint="模糊查询"></td>
        <td width="40">性别</td>
        <td width="160"><input type="text" class="Edit" kind="dic" src="DIC_SEX" fieldname="SEX"></td>
        <td width="40">籍贯</td>
        <td width="160"><input type="text" class="Edit" kind="dic" src="DIC_CODE"
fieldname="PLACECODE"></td>
        <td><input iconCls="icon-qry" type="button" value="查 询" onEfsClick="doQry()"></td>
        <td> </td>
      </tr>
    </TABLE>
  </form>
</div>

<div id="psnGrid" region="center" xtype="grid" pagingBar="true" pageSize="25"
onEfsRowClick="doGridClick()" onEfsRowDblClick="onEditEx()">
   <div xtype="tbar">
     <span style="font-size:9pt;font-weight:bold;color:#15428B;">学生列表</span>
    <div text="->"></div>
     <div iconCls="icon-edit" id="cmdEdit" text="编辑学生#E" onEfsClick="onEditEx()" disabled></div>
    <div text="-"></div>
     <div iconCls="icon-Del" id="cmdDel" text="删除学生#D" onEfsClick="onDelEx()" disabled></div>
    <div text="-"></div>
     <div iconCls="icon-back" text="返 回" onEfsClick="top.showTask()"></div>
   </div>
     <div id="psnList" xtype="store" url="../sysadmin/baseRefWeb.aspx?method=QryPersonList"
baseParams="{txtXML:g_XML}" autoLoad="true">
         <div xtype="xmlreader" fieldid="PERSONID" record="ROW" tabName="PERSON"
totalRecords="QUERYINFO@records">
             <div name="PERSONID" mapping="PERSONID"></div>
             <div name="NAME" mapping="NAME"></div>
             <div name="IDCARD"></div>
             <div name="SEX"></div>
```

```
      <div name="PLACECODE"></div>
            <div name="BIRTHDAY"></div>
            <div name="TEL"></div>
        </div>
    </div>
    <div xtype="colmodel">
        <div type="checkbox"></div>
    <div header="学生编码" width="80" sortable="true" dataIndex="PERSONID"></div>
        <div header="学生姓名" width="80" sortable="true" dataIndex="NAME"></div>
        <div header="身份证号" width="120" sortable="true" dataIndex="IDCARD" align="center"></div>
        <div header="性别" width="40" sortable="true" dataIndex="SEX" kind="dic" src="DIC_SEX"></div>
    <div header="籍贯" width="120" sortable="true" dataIndex="PLACECODE" kind="dic" src="DIC_CODE"
align="center"></div>
        <div header="出生日期" width="100" sortable="true" dataIndex="BIRTHDAY" align="center"></div>
        <div header="联系电话" width="100" sortable="true" dataIndex="TEL"></div>
    </div>
</div>

<!-- window开始 -->
<div iconCls="icon-panel" id="PsnMWin" xtype="window" width="560" height="255" title="修改学生"
resizable="true" modal="true">
  <div region="center" xtype="panel" title="" border="false" autoScroll="true">
    <div xtype="tbar">
      <div text="->"></div>
      <div iconCls="icon-ok2" id="cmdUser" text="确　定" onEfsClick="doPsnEdit()"></div>
    </div>
    <form id="frmData" class="efs-box" method="post" url="../sysadmin/XmlDataDeal.aspx"
onEfsSuccess="frmPostSubBack(true)" onEfsFailure="frmPostSubBack(false)">
      <TABLE class="formArea">
        <TR>
          <TD width="100" labelFor="name">姓　名</TD>
          <TD><INPUT id="name" type="text" kind="zhunicode" must="true" maxlength="50"
fieldname="PERSON/NAME" datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">身份证号码</TD>
          <TD><INPUT type="text" kind="idcard" fieldname="PERSON/IDCARD" sex="sex" birthdate="birthday"
datatype="0" state="0"></TD>
        </TR>
        <TR>
          <TD width="100" labelFor="sex">性　别</TD>
          <TD><INPUT type="text" kind="dic" src="DIC_SEX" id="sex" fieldname="PERSON/SEX" must="true"
datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100" labelFor="birthday">出生日期</TD>
```

```
          <TD><INPUT type="text" kind="date" id="birthday" fieldname="PERSON/BIRTHDAY" datatype="3"
state="0" must="true"></TD>
        </TR>
        <TR>
          <TD width="100">籍　贯</TD>
          <TD><INPUT type="text" kind="dic" src="DIC_CODE" fieldname="PERSON/PLACECODE" datatype="0"
state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">年　龄</TD>
          <TD><INPUT type="text" kind="int" range=[0,100] fieldname="PERSON/YEAROLD" datatype="1"
state="0"></TD>
        </TR>
        <TR>
          <TD width="100">邮　箱</TD>
          <TD><INPUT type="text" kind="email" fieldname="PERSON/EMAIL" datatype="0" state="0"></TD>
          <TD width="20"></TD>
          <TD width="100">电话号码</TD>
          <TD><INPUT type="text" kind="text" fieldname="PERSON/TEL" datatype="0" state="0"></TD>
        </TR>
        <tr>
          <td>备注</td>
          <td colspan="4"><TEXTAREA class="Edit" kind="text" style="height:60px;width:430px"
fieldname="PERSON/BAK" state="0" datatype="0"></TEXTAREA>
          </td>
        </tr>
      </TABLE>
      <INPUT type="hidden" kind="text" fieldname="PERSON/PERSONID" datatype="0" state="5" operation="1"
writeevent="0"><!--operation="0"  定义为修改接口-->
    </form>
  </div>
</div>
<!-- window结束 -->

</BODY>
</HTML>
```

界面简单截图



5.2.2、查询学生列表组件接口方法源代码

在 Person.cs 里添加 personList 方法，作用是查询学生列表信息。查询学生列表的时候调用该方法。

查询学生列表组件路径及名称: namespace Efsframe.cn.person.Person

```
    /// <summary>
    /// 查询学生档案列表
    /// </summary>
    /// <param name="strXML">标准查询条件结构</param>
    /// <returns>标准查询返回结构</returns>
    public static string personList(string strXML)
    {
      // 构造标准查询XML接口分析类对象
      QueryDoc obj_Query = new QueryDoc(strXML);
      int int_PageSize = obj_Query.getIntPageSize();

      int int_CurrentPage = obj_Query.getIntCurrentPage();

      // 查询字典
      string str_Select = "s.PERSONID PERSONID,s.NAME NAME,s.IDCARD IDCARD,s.SEX
SEX,s.PLACECODE PLACECODE,s.BIRTHDAY BIRTHDAY,s.TEL TEL";
      // 查询表
      string str_From = Table.PERSON + Common.SPACE + Table.S;
      // 构建标准的查询条件
      string str_Where = obj_Query.getConditions();
      str_Where = General.empty(str_Where) ? str_Where : Common.WHERE + str_Where;

      // 日期型字段列表
      string[] str_DateList = { Field.BIRTHDAY };

      // 标准的、统一的分页查询接口
      return CommonQuery.basicListQuery(str_Select,
                                        str_From,
                                        str_Where,
```

```
                                    Field.PERSONID,
                                    str_DateList,
                                    int_PageSize,
                                    int_CurrentPage);
    }
```

在 baseRef.cs 中添加方法

在这里，获取页面传递过来的 xml，然后调用相应方法进行处理。

反射类路径及名称： namespace Efsframe.cn.baseCls.baseRef

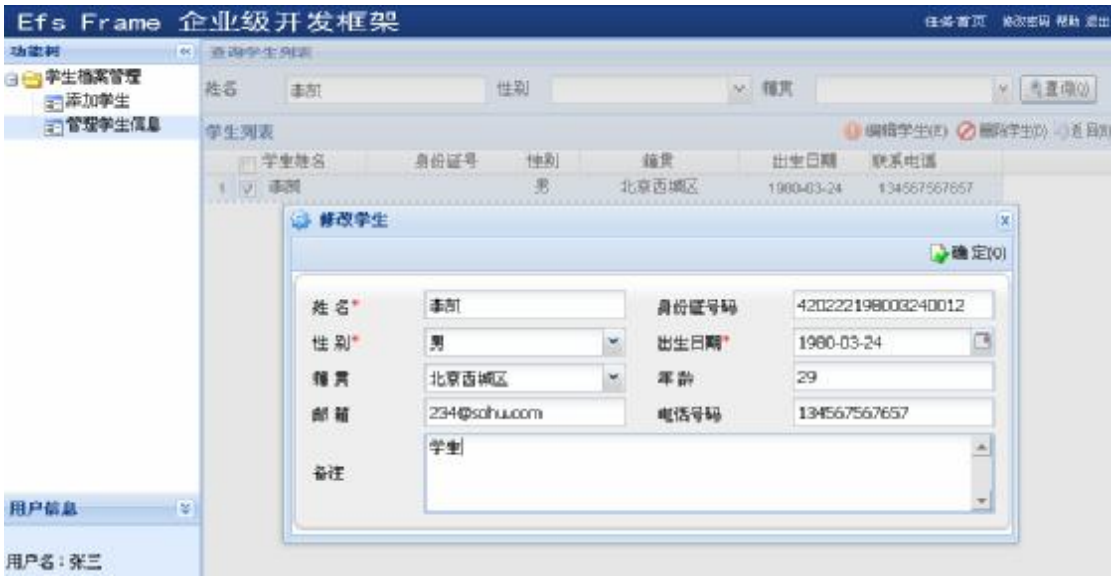
```
/// <summary>
    /// 查询学生列表
    /// </summary>
    /// <param name="Request"></param>
    /// <param name="Response"></param>
    /// <returns></returns>
    public string QryPersonList(HttpRequest Request, HttpResponse Response)
    {
      string strXML = Request["txtXML"];
      return Person.personList(strXML);
    }
```

5.3、修改/删除学生

5.3.1、修改/删除学生页面源代码

查询学生列表页面路径及名称： WebRoot\person\ qryPsnList.aspx
修改删除学生页面代码都包含在qryPsnList.aspx里面，用到的模态窗口

界面简单截图



5.3.2、修改/删除学生组件接口方法源代码

修改/删除学生信息统一调用公共页面：**sysadmin\XmlDataDeal.aspx**
该页面为统一的标准XML操作型数据处理页面，对于标准的XML可进行统一处理，不需要为标准接口再添加新的组件和页面进行单独处理。

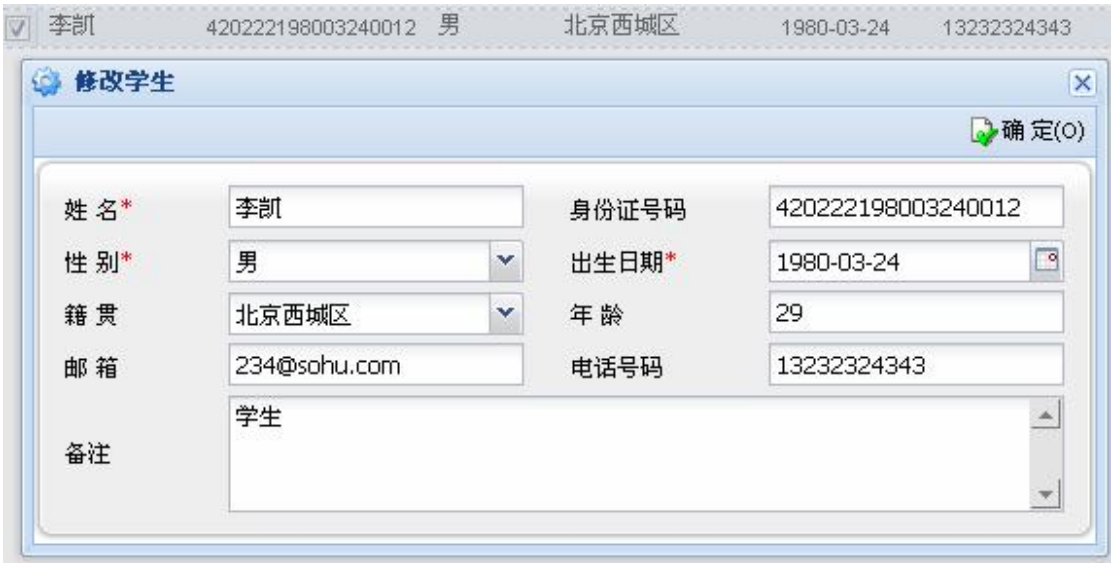
Operation.dealWithXml 为通用方法，基本业务操作，可以处理通用型的，不需要任何业务数据修改的[添加]和[修改]类业务

5.4、查询学生详细信息

5.4.1、查询学生详细信息页面源代码

查询学生列表页面路径及名称：**efs\person\ qryPsnList.aspx**
修改学生信息的时候就包含了查询学生详细信息，相关页面代码都包含在qryPsnList.aspx里面

界面简单截图



5.4.2、查询学生详细信息组件接口方法源代码

在 Person.cs 里添加 personDetail 方法，作用是查询学生相信信息。在修改信息里就调用到了查询详细信息方法。

查询学生列表组件路径及名称: namespace Efsframe.cn.person.Person

```
/// <summary>
    /// 查询学生详细档案信息
    /// </summary>
    /// <param name="sPersonID">学生编号</param>
    /// <returns>标准查询返回结构</returns>
    public static string personDetail(string sPersonID)
    {
      string str_Select = "s.PERSONID PERSONID,s.NAME NAME,s.IDCARD IDCARD,s.SEX SEX,s.BIRTHDAY
BIRTHDAY,s.PLACECODE PLACECODE,s.YEAROLD YEAROLD,s.EMAIL EMAIL,s.TEL TEL,s.BAK BAK";

      string str_From = Table.PERSON + Common.SPACE + Table.S;

      string str_Where = Common.WHERE + Common.SPACE + Field.PERSONID + Common.EQUAL +
General.addQuotes(sPersonID);

      string[] str_DateList = { Field.BIRTHDAY };

      return CommonQuery.basicListQuery(str_Select,
                                        str_From,
                                        str_Where,
                                        Field.PERSONID,
                                        str_DateList,
                                        1, 1,
```

```
                                    Table.PERSON);

    }
```

在 baseRef.cs 中添加方法

在这里，获取页面传递过来的 xml，然后调用相应方法进行处理。

反射类路径及名称： namespace Efsframe.cn.baseCls.baseRef

```
/// <summary>
    /// 查询学生详细信息
    /// </summary>
    /// <param name="Request"></param>
    /// <param name="Response"></param>
    /// <returns></returns>
    public string QryPersonDetail(HttpRequest Request, HttpResponse Response)
    {
      string strPersonID = Request["txtPersonID"];
      return Person.personDetail(strPersonID);
    }
```

6、接口文档设计补充说明

6.1、标准业务操作型 XML 文档数据接口分析说明

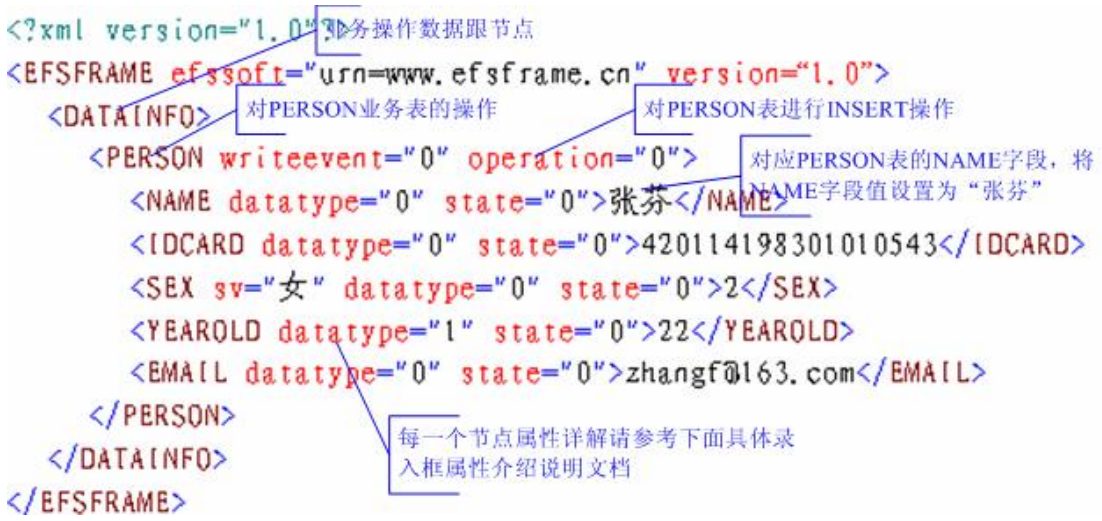
在开始了解接口文档设计之前，请大家先了解一下接口规范设计的说明，有助于大家加深对整个框架的理解，具体实例如下：

注：Web 表现层提交给组件接口的标准添加操作的 XML 文档结构

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
  <DATAINFO>
    <PERSON writeevent="0" operation="0">
      <NAME datatype="0" state="0">张芬</NAME>
      <IDCARD datatype="0" state="0">420114198301010543</IDCARD>
      <SEX sv="女" datatype="0" state="0">2</SEX>
```

```
        <YEAROLD datatype="1" state="0">22</YEAROLD>
        <EMAIL datatype="0" state="0">zhangf@163.com</EMAIL>
      </PERSON>
    </DATAINFO>
</EFSFRAME>
```

对照图解说明:



最终构造的T-SQL语句为:

```
INSERT INTO PERSON (NAME,IDCARD,SEX,YEAROLD,EMAIL,BIRTHDAY) VALUES ('张芬','420114198301010543','2',22,'zhangf@163.com')
```

以上标准操作型 XML 文档数据结构，是整个 Efs 框架的核心，我们的 Web 表现层是围绕这样一个标准结构的快速方便构成而设计，业务逻辑组件层也是围绕能快速分析处理这样一个通用标准结构而设计，它作为整个 Efs 企业级开发框架的核心纽带而存在。

Efs 框架中的 Web 表现层可以通过 Efs.Common.getOpXml()方法轻松实现该标准接口，业务逻辑组件层将通过 SQLStorage.dealWithEventSQL()模块方法轻松构造出标准的 T-SQL 语句。

对于以上文档中的每一个节点属性的说明请参考:

6.1.1、operation 属性

描述: 指对该节点所指向的业务表的操作类型。

枚举值:

枚举值	枚举值说明	备注
0	对指定业务表进行添加操作	在转化为 T-SQL 过程中，将对该业务表的操作语句转化为 **insert** 语句。
1	对指定业务表进行更新操作	在转化为 T-SQL 过程中，将对该业务表的操作语句转化为 **update** 语句。
2	对指定业务表进行删除操作	在转化为 T-SQL 过程中，将对该业务表的操作语句转化为 **delete** 语句。

备注：如上示例中的 **operation=”0”** 表示对业务表 PERSON 进行 **insert** 操作。

6.1.2、writeevent 属性

描述：指对该节点所指向的业务表是否进行同步历史记录操作。

枚举值：

枚举值	枚举值说明	备注
0	不记录历史操作（默认值）	在转化为 T-SQL 过程中，忽略对业务表的历史表进行同步添加功能。
1	记录历史操作	在转化为 T-SQL 过程中，对业务表的历史表同步进行 **insert** 操作，记录历史操作信息。

6.1.3、datetype 属性

描述：指节点所对应的数据库中的业务表与表字段的数据类型。

枚举值：

枚举值	枚举值说明	备注
0	字符类型字段	该数据类型特点为在转化成标准 T-SQL 过程中，给字段的值加上单引号。
1	数字类型字段	该数据类型特点为在转化成标准 T-SQL 过程中，给字段的值不加单引号。
2	系统当前时间	该数据类型特点为在转化成标准 T-SQL 过程中，将该字段值默认为系统时间。

3	日期类型字段，精确到日	该数据类型特点为在转化成标准 T-SQL 过程中，将该字段的值强制类型转化为日期型。
4	日期时间类型字段，精确到分	该数据类型特点为在转化成标准 T-SQL 过程中，将该字段的值强制类型转化为日期型。
5	二进制字段	该数据类型特点为在转化成标准 T-SQL 过程中，将该字段的值转化为二进制流写入。

备注：如上示例中的 **datatype=”1”** 表示 PERSON 业务表中的 YEAROLD 字段为数字型，在转化为标准 T-SQL 过程中，不需要对其值加单引号。

6.1.4、state 属性

描述：指该节点所对应的数据中的业务表与表字段的当前操作类型。

枚举值:

枚举值	枚举值说明	备注
0	普通操作类型	在转化为 T-SQL 过程中，该字段作为普通操作处理，如 **insert** 语句中作为添加字段，**update** 语句中作为更新字段，**delete** 语句中作为无效字段
5	数据操作条件字段	在转化为 T-SQL 过程中，该字段转化为条件字段处理，如 **update**、**delete** 语句中作为 **where** 条件
9	忽略处理字段	在转化为 T-SQL 过程中，忽略该字段不做处理，同 **state** 为空的情况一样

备注：如上示例中的 **state=”0”** 在对表示 PERSON 业务表中的 YEAROLD 进行操作的过程中，将其作为普通操作字段处理即可。

6.2、标准接口返回 XML 文档数据格式分析

在 Efs 框架中，在没有特殊处理需要时，我们将统一按照以下规范 XML 文档数据作为逻辑组件执行完后的返回标准，统一的标准便于我们进行统一的处理。

方法返回实例

正确返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
   <ERRORINFO> <!—接口方法执行完后的错误信息节点-->
      <ERRORRESULT>0</ERRORRESULT>
      <!—错误编码 ， 0 或者 00 表示处理成功-->
   </ERRORINFO>
</EFSFRAME>
```

错误返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
   <ERRORINFO>
      <ERRORRESULT>1</ERRORRESULT>
      <!--错误编号-->
      <FUNCERROR>列名 'DES' 无效。 Insert into PERSON
(NAME,IDCARD,SEX,BIRTHDAY,YEAROLD,CLASSID,EMAIL,TEL,PERSONID,DES) Values ('张三
','420118198506040231','1',CAST('1985-06-04' AS
DATETIME),25,'','zhangsan@163.com','000-00000000','0907000004','描述')</FUNCERROR>
      <!--错误异常描述-->
   </ERRORINFO>
</EFSFRAME>
```

6.3、标准查询型 XML 文档数据接口分析说明

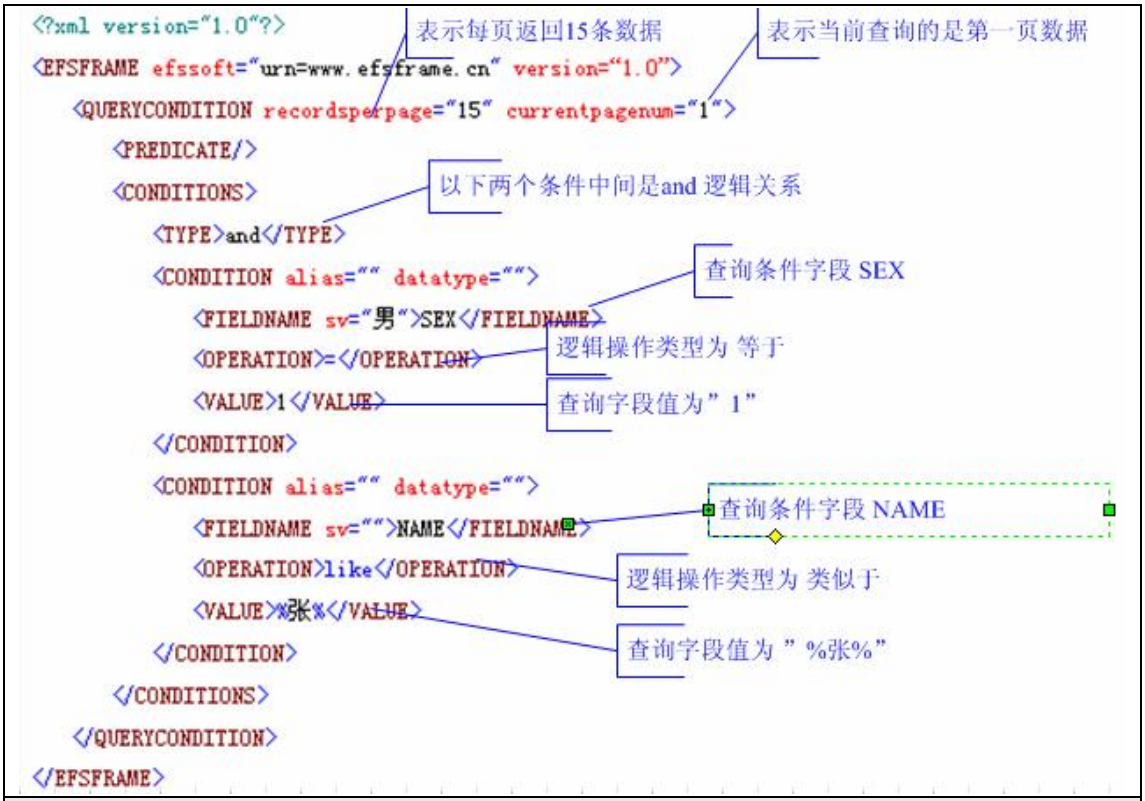
在 Efs 框架中，我们将统一按照以下规范来完成分页查询或者其他查询条件的组织，Web 表现层将有 Efs.Common.getQryXml()方法轻松实现该接口，逻辑业务组件层将通过 QueryDoc.getConditions() 类方法轻松构造出标准的 Where 查

询条件，具体接口实例如下：

注：Web 表现层提交给组件接口的标准查询操作的 XML 文档结构

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <QUERYCONDITION recordsperpage="15" currentpagenum="1">
        <PREDICATE/>
        <CONDITIONS>
            <TYPE>and</TYPE>
            <CONDITION alias="" datatype="">
                <FIELDNAME sv="男">SEX</FIELDNAME>
                <OPERATION>=</OPERATION>
                <VALUE>1</VALUE>
            </CONDITION>
            <CONDITION alias="" datatype="">
                <FIELDNAME sv="">NAME</FIELDNAME>
                <OPERATION>like</OPERATION>
                <VALUE>%张%</VALUE>
            </CONDITION>
        </CONDITIONS>
    </QUERYCONDITION>
</EFSFRAME>
```

对照图解说明：



以上结构最终构造的出来的T-SQL中的查询语句条件:

```
SEX='1' AND NAME LIKE '%张%'
```

6.4、添加学生接口

包名称: namespace Efsframe.cn.person

类名称: Person

方法名称: public static string addNew(string strXml)

入口参数说明:

sXml	XML 标准业务操作数据接口

返回参数说明:

strRetXML	标准的返回 XML 结构

入口 sXML 参数实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
  <DATAINFO>
    <PERSON writeevent="0" operation="0">
```

```
        <NAME datatype="0" state="0">张三</NAME>
        <IDCARD datatype="0" state="0">42011819850604024X</IDCARD>
        <SEX sv="女" datatype="0" state="0">2</SEX>
        <BIRTHDAY datatype="3" state="0">19850604</BIRTHDAY>
        <YEAROLD datatype="1" state="0">25</YEAROLD>
        <EMAIL datatype="0" state="0">zhangsan@163.com</EMAIL>
        <TEL datatype="0" state="0">000-00000000</TEL>
        <PERSONID datatype="0" state="0"></PERSONID>
    </PERSON>
  </DATAINFO>
  <USERINFO>
    <USERID>0000000107</USERID>
    <USERTITLE>guozhijun</USERTITLE>
    <USERNAME>郭志军</USERNAME>
    <UNITID>420100000000</UNITID>
    <UNITNAME>测试单位</UNITNAME>
    <MTYPE>10</MTYPE>
    <LOGID>00000000002300</LOGID>
    <USERTYPE>2</USERTYPE>
  </USERINFO>
</EFSFRAME>
```

方法返回实例(约定俗成的返回实例，所以一般接口中无需写返回实例)

正确返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
  <ERRORINFO>
    <ERRORRESULT>0</ERRORRESULT>
    <!--0 或者 00 表示处理成功-->
```

```
    </ERRORINFO>
</EFSFRAME>
```

错误返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <ERRORINFO>
        <ERRORRESULT>1</ERRORRESULT>
        <!--错误编号-->
        <FUNCERROR>列名 'DES' 无效。 Insert into PERSON
(NAME,IDCARD,SEX,BIRTHDAY,YEAROLD,CLASSID,EMAIL,TEL,PERSONID,DES) Values ('张三
','420118198506040231','1',CAST('1985-06-04' AS
DATETIME),25,'','zhangsan@163.com','000-00000000','0907000004','描述')</FUNCERROR>
        <!--错误异常描述-->
    </ERRORINFO>
</EFSFRAME>
```

6.5、修改学生接口

包名称: namespace Efsframe.cn.baseCls

类名称: Operation

方法名称: public static string dealWithXml(string strXML)

入口参数说明:

sXml	XML 标准业务操作数据接口

返回参数说明:

strRetXML	标准的返回 XML 结构

入口 sXML 参数实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <DATAINFO>
        <PERSON writeevent="0" operation="1">
```

```
        <NAME datatype="0" state="0">张三</NAME>
        <IDCARD datatype="0" state="0">420118198506040 24X</IDCARD>
        <SEX sv="女" datatype="0" state="0">2</SEX>
        <BIRTHDAY datatype="3" state="0">19850604</BIRTHDAY>
        <YEAROLD datatype="1" state="0">25</YEAROLD>
        <EMAIL datatype="0" state="0">zhangsan@163.com</EMAIL>
        <TEL datatype="0" state="0">000-00000000</TEL>
        <PERSONID datatype="0" state="5">0907000006</PERSONID>
      </PERSON>
   </DATAINFO>
   <USERINFO>
      <USERID>0907000001</USERID>
      <USERTITLE>test</USERTITLE>
      <USERNAME>测试用户</USERNAME>
      <UNITTILE>测试单位</UNITTILE>
      <UNITID>0000000001</UNITID>
      <UNITNAME>10</UNITNAME>
   </USERINFO>
</EFSFRAME>
```

方法返回实例(同添加接口实例)

6.6、删除学生接口

包名称: namespace Efsframe.cn.baseCls

类名称: Operation

方法名称: public static string dealWithXml(string strXML)

入口参数说明:

sXml	XML 标准业务操作数据接口

返回参数说明:

strRetXML 标准的返回 XML 结构

入口 sXML 参数实例：

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
   <DATAINFO>
      <PERSON writeevent="0" operation="2">
         <PERSONID datatype="0" state="5">0907000006</PERSONID>
      </PERSON>
   </DATAINFO>
   <USERINFO>
      <USERID>0907000001</USERID>
      <USERTITLE>test</USERTITLE>
      <USERNAME>测试用户</USERNAME>
      <UNITTILE>测试单位</UNITTILE>
      <UNITID>0000000001</UNITID>
      <UNITNAME>10</UNITNAME>
   </USERINFO>
</EFSFRAME>
```

方法返回实例(同添加接口实例)

6.7、查询学生列表接口

包名称： namespace Efsframe.cn.person

类名称： Person

方法名称： public static string personList(string strXML)

入口参数说明：

sXml XML 标准查询操作数据接口

返回参数说明：

strRetXML 标准的返回 XML 结构

入口 sXML 参数实例：

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
   <QUERYCONDITION currentpagenum="1" recordsperpage="18">
      <PREDICATE/>
      <CONDITIONS>
         <TYPE>and</TYPE>
         <CONDITION alias="" datatype="">
            <FIELDNAME sv="">NAME</FIELDNAME>
            <OPERATION>like</OPERATION>
            <VALUE>%张%</VALUE>
         </CONDITION>
         <CONDITION alias="" datatype="">
            <FIELDNAME sv="男">SEX</FIELDNAME>
            <OPERATION>=</OPERATION>
            <VALUE>1</VALUE>
         </CONDITION>
      </CONDITIONS>
   </QUERYCONDITION>
</EFSFRAME>
```

方法返回实例

正确返回实例：

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
   <QUERYINFO totalpages="1" records="1">
      <ROW>
         <PERSONID>0907000006</PERSONID>
         <NAME>张谦</NAME>
```

```
            <IDCARD>420118198506040024X</IDCARD>
            <SEX>2</SEX>
            <BIRTHDAY>1985-06-04</BIRTHDAY>
            <YEAROLD>25</YEAROLD>
            <TEL>000-00000000</TEL>
            <EMAIL>zhangqian@163.com</EMAIL>
        </ROW>
    </QUERYINFO>
    <ERRORINFO>
        <ERRORRESULT>00</ERRORRESULT>
    </ERRORINFO>
</EFSFRAME>
```

错误返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <ERRORINFO>
        <ERRORRESULT>1</ERRORRESULT>
        <FUNCERROR>查询为空，没有符合条件的记录</FUNCERROR>
    </ERRORINFO>
</EFSFRAME>
```

6.8、查询学生详细信息接口

包名称: namespace Efsframe.cn.person

类名称: Person

方法名称: public static string personDetail(string sPersonID)

入口参数说明:

sPersonID	学生编号

返回参数说明:

strRetXML	标准的返回 XML 结构

方法返回实例

正确返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <QUERYINFO totalpages="1" records="1">
        <PERSON>
            <PERSONID>0907000006</PERSONID>
            <NAME>张谦</NAME>
            <IDCARD>420118198506040024X</IDCARD>
            <SEX sv="女">2</SEX>
            <BIRTHDAY sv="19850604">1985-06-04</BIRTHDAY>
            <YEAROLD>25</YEAROLD>
            <TEL>000-00000000</TEL>
            <EMAIL>zhangqian@163.com</EMAIL>
        </PERSON>
    </QUERYINFO>
    <ERRORINFO>
        <ERRORRESULT>00</ERRORRESULT>
    </ERRORINFO>
</EFSFRAME>
```

错误返回实例:

```
<?xml version="1.0"?>
<EFSFRAME efsframe="urn=www-efsframe-cn" version="1.0">
    <ERRORINFO>
        <ERRORRESULT>1</ERRORRESULT>
        <FUNCERROR>查询为空，没有符合条件的记录</FUNCERROR>
    </ERRORINFO>
</EFSFRAME>
```

7、发布测试

通过以上步骤的开发，基本完成了对学生信息的添加、修改、删除、查询等操作，接着将 Web 页面、组件、数据库统一发布，即可进行项目功能测试。如果大家在接口约定过程中没有出现什么问题，都按照接口约定来完成了各自的开发，整个过程出错的可能很小，即使有错误，也很容易排查和修正。