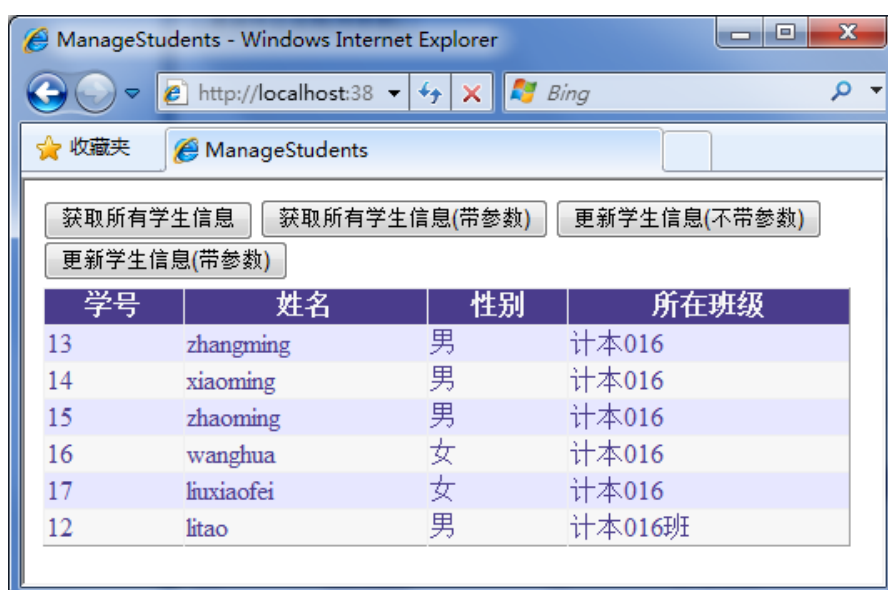
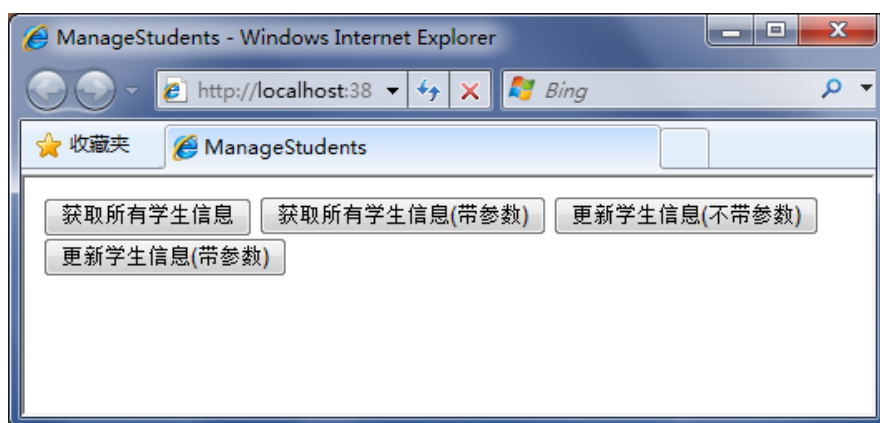


.net 程序示例

瀚高基础软件股份有限公司版权所有
仅允许不作任何修改的转载和转发

功能概述

1. 从 HighGo DB 的学生表中查询出所有学生信息，结果以表格形式显示。
2. 该表由学号、姓名、性别、所在班级组成。
3. 更新学生信息分别实现带参数和不带参数的方式。
4. 查询学生信息分别实现带参数和不带参数的方式
5. 运行界面如下，比较简单，四个按钮和一个数据表格。



学生信息管理示例代码

1. 数据访问类 dataaccess.cs 文件如下:

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using Nhgdb; //使用名空间 Nhgdb

namespace ManageStudents
{
    public class dataaccess
    {
        static private string strConn =
ConfigurationManager.ConnectionStrings["highgo"].ConnectionString; // 1. 数据库连接串

        //执行数据查询操作
        public DataTable ExeQuery(string sql)
        {
            DataTable aTable = new DataTable();
            NpgsqlConnection conn = new NpgsqlConnection(strConn); //建立数据库连接
            NpgsqlDataAdapter adpater = new NpgsqlDataAdapter(sql, conn); //构建sql语句的适配
器对象
            NpgsqlCommandBuilder builder = new NpgsqlCommandBuilder(adpater);
            try
            {
                conn.Open(); //打开数据库连接
                adpater.Fill(aTable); //执行sql填充结果集.
                return aTable;
            } catch (Exception ex) {
                throw ex;
                //return null;
            } finally {
                conn.Close(); //关闭数据库连接
            }
        }

        //执行更新, 删除和插入操作
        public void ExeNoQuery(string sql)
```

```

{
    NpgsqlConnection conn = new NpgsqlConnection(strConn);
    NpgsqlCommand cmd = new NpgsqlCommand(sql, conn); //构建sql命令
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery(); //执行sql 命令。
    } catch {
        return ;
    }finally {
        conn.Close();
    }
}
//执行带参数的sql 语句，需要指定参数和参数个数。
public void ExeNonQueryParams(string sql, NpgsqlParameter[] Pars, int count)
{
    NpgsqlConnection conn = new NpgsqlConnection(strConn);
    NpgsqlCommand cmd = new NpgsqlCommand(sql, conn);
    try
    {
        conn.Open();
        for (int i = 0; i < count; ++i)
        {
            cmd.Parameters.Add(Pars[i]); //添加命令参数
        }
        cmd.ExecuteNonQuery();
    } catch {
        return;
    } finally{
        conn.Close();
    }
}

```

```

public DataTable ExeQueryParams(string sql, NpgsqlParameter[] Pars, int count)
{
    DataTable aTable = new DataTable();
    NpgsqlConnection conn = new NpgsqlConnection(strConn);
    NpgsqlDataAdapter adpater = new NpgsqlDataAdapter(sql, conn);
    for (int i = 0; i < count; ++i)
    {
        adpater.SelectCommand.Parameters.Add(Pars[i]);
    }
    NpgsqlCommandBuilder builder = new NpgsqlCommandBuilder(adpater);
    try

```

```

        {
            conn.Open();
            adpater.Fill(aTable);
            return aTable;
        }catch (Exception ex) {
            throw ex;
        } finally {
            conn.Close();
        }
    }
}
}
}

```

2. Default.aspx.cs 文件代码如下:

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using Nhgdb;
namespace ManageStudents
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        //查询学号为12 的学生信息
        protected void btnGetStudent_Click(object sender, EventArgs e)
        {
            string sql = "select stunum,name,sex,inclass from student where stunum=:num";
            dataaccess adb = new dataaccess();
            NpgsqlParameter[] par = new NpgsqlParameter[1];
            par[0] = new NpgsqlParameter("num", DbType.Int32);
            par[0].Value = 12;
            DataTable aTable = adb.ExeQueryParams(sql, par, 1);
            GridView1.DataSource = aTable;
            GridView1.DataBind();
        }
    }
}

```

```

//更新学号为12 的学生信息
protected void btnUpdateStudent_Click(object sender, EventArgs e)
{
    string sql =
        "update student set name=' litao',inclass=' 计本016班' where stunum=:num ";
    NpgsqlParameter[] par = new NpgsqlParameter[1];
    par[0] = new NpgsqlParameter("num", DbType.Int32);
    par[0].Value = 12;
    dataaccess adb = new dataaccess();
    adb.ExeNoQueryParams(sql, par, 1);
    btnGetStudent_Click(sender, e);
}
protected void btnGetAllStus_Click(object sender, EventArgs e)
{
    string sql = "select stunum,name,sex,inclass from student";
    dataaccess adb = new dataaccess();
    DataTable aTable = adb.ExeQuery(sql);
    GridView1.DataSource = aTable;
    GridView1.DataBind();
}
protected void btnEditStudent_Click(object sender, EventArgs e)
{
    string sql = "update student set name=' liming', inclass=' 计本016班' where stunum=12 ";
    dataaccess adb = new dataaccess();
    adb.ExeNoQuery(sql);
    btnGetStudent_Click(sender, e);
}
}
}

```

3. Default.aspx 文件代码如下：

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="HgdbTest._Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>ManageStudents</title>
    <link href="main.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
    <table width="500">
        <tr><td>

```

```

        <asp:Button ID="btnGetAllStus" runat="server" Text="获取所有学生信息"
            onclick="btnGetAllStus_Click" />
        <asp:Button ID="btnGetStudent" runat="server" Text="获取所有学生信息(带参
数)"
            onclick="btnGetStudent_Click" />
        <asp:Button ID="btnEditStudent" runat="server" Text="更新学生信息(不带参
数)" onclick="btnEditStudent_Click" />
        <asp:Button ID="btnUpdateStudent" runat="server" Text="更新学生信息(带参
数)" onclick="btnUpdateStudent_Click"/>
    </td></tr>
</tr>
<td>
    <asp:GridView ID="GridView1" runat="server" Width="98%"
        style="white-space:nowrap" AutoGenerateColumns="False">
        <RowStyle BackColor="#E7E7FF" ForeColor="#4A3C8C" CssClass="gvRow" />
        <HeaderStyle BackColor="#4A3C8C" Font-Bold="True" ForeColor="#F7F7F7"
CssClass="gvHeader" />
        <AlternatingRowStyle BackColor="#F7F7F7" CssClass="gvAlternatingRow" />
        <Columns>
            <asp:BoundField DataField="stunum" HeaderText="学号" />
            <asp:BoundField DataField="name" HeaderText="姓名" />
            <asp:BoundField DataField="sex" HeaderText="性别" />
            <asp:BoundField DataField="inclass" HeaderText="所在班级" />
        </Columns>
    </asp:GridView>
</td>
</tr>
</table>
</form>
</body>
</html>

```

4. Web.config 文件配置如下:

```

<?xml version="1.0"?>
<configuration>
    <configSections>
        <section name="hibernate-configuration"
            type="NHibernate.Cfg.ConfigurationSectionHandler, NHibernate"
            />
    </configSections>
    <appSettings/>
    <connectionStrings>
        <add name="highgo" connectionString="Server=127.0.0.1;Port=5866;User
Id=ldl;Password=1;Database=highgo;"/>
    </connectionStrings>

```

```

<system.web>
  <!--
    设置 compilation debug="true" 可将调试符号插入
    已编译的页面中。但由于这会
    影响性能，因此只在开发过程中将此值
    设置为 true。
  -->
  <compilation debug="false">
</compilation>
  <!--
    通过 <authentication> 节可以配置 ASP.NET 用来
    识别进入用户的
    安全身份验证模式。
  -->
  <authentication mode="Windows"/>
  <!--
    如果在执行请求的过程中出现未处理的错误，
    则通过 <customErrors> 节可以配置相应的处理步骤。具体说来，
    开发人员通过该节可以配置
    要显示的 html 错误页
    以代替错误堆栈跟踪。
  -->
  <customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
    <error statusCode="403" redirect="NoAccess.htm" />
    <error statusCode="404" redirect="FileNotFound.htm" />
  </customErrors>
  -->
</system.web>
<system.codedom>
</system.codedom>
  <!--
    在 Internet 信息服务 7.0 下运行 ASP.NET AJAX 需要 system.webServer
    节。对早期版本的 IIS 来说则不需要此节。
  -->
  <system.webServer>
</system.webServer>
</configuration>

```

5. 建学生表并初始化测试数据

```

CREATE TABLE student
(
  stunum integer NOT NULL,
  "name" character varying(30),
  sex integer,
  inclass character varying(60)

```

)

```
insert into student values(12,'liming',1,'计本 016');
```

```
insert into student values(13, 'zhangming',1,'计本 016');
```

```
insert into student values(14,'xiaoming',1,'计本 016');
```

```
insert into student values(15,'zhaoming',1,'计本 016');
```

```
insert into student values(16,'wanghua',0,'计本 016');
```

```
insert into student values(17,'liuxiaofei',0,'计本 016');
```