

三层列表展示

Model 添加类 UserInfo.cs

```
public class UserInfo
{
    public int ID { get; set; }
    public string UserName { get; set; }
    public string UserPass { get; set; }
    public DateTime RegTime { get; set; }
    public string Email { get; set; }
}
```

在 DAL 中引用 Model

配置文件中配置数据库

```
<connectionStrings>
  <add name="connStr"
connectionString="server=.;uid=home;pwd=;database=EFFristModel"/>
</connectionStrings>
```

为 DAL 引用 System.Configuration

DAL 中添加类 SqlHelper.cs

```
public class SqlHelper
{
    private readonly static string connStr =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    public static DataTable GetTable(string sql, CommandType type, params SqlParameter[]
pars)
    {
        using (SqlConnection conn = new SqlConnection(connStr))
        {
```

```

        using (SqlDataAdapter atper = new SqlDataAdapter(sql, conn))
        {
            atper.SelectCommand.CommandType = type;
            if (pars != null)
            {
                atper.SelectCommand.Parameters.AddRange(pars);
            }
            DataTable da = new DataTable();
            atper.Fill(da);
            return da;
        }
    }
}

public static int ExecuteNonQuery(string sql, CommandType type, params SqlParameter[]
pars)
{
    using (SqlConnection conn = new SqlConnection(connStr))
    {
        using (SqlCommand cmd = new SqlCommand(sql, conn))
        {
            cmd.CommandType = type;
            if (pars != null)
            {
                cmd.Parameters.AddRange(pars);
            }
            conn.Open();
            return cmd.ExecuteNonQuery();
        }
    }
}
}

```

DAL 中添加类 UserInfoDal.cs

```

public class UserInfoDal:IDAL.IUserInfoDal
{
    /// <summary>
    /// 返回列表
    /// </summary>
    /// <returns></returns>
    public List<UserInfo> GetEntityList()
    {
        string sql = "select * from UserInfo";
    }
}

```

```

        DataTable da = SqlHelper.GetTable(sql, CommandType.Text);
        List<UserInfo> list = null;
        if (da.Rows.Count > 0)
        {
            list = new List<UserInfo>();
            UserInfo userInfo = null;
            foreach (DataRow row in da.Rows)
            {
                userInfo = new UserInfo();
                LoadEntity(row, userInfo);
                list.Add(userInfo);
            }
        }
        return list;
    }

    public void LoadEntity(DataRow row, UserInfo userInfo)
    {
        userInfo.ID = Convert.ToInt32(row["ID"]);
        userInfo.UserName = row["UserName"] != DBNull.Value ?
row["UserName"].ToString() : string.Empty;
        userInfo.UserPass = row["UserPass"] != DBNull.Value ? row["UserPass"].ToString() :
string.Empty;
        userInfo.Email = row["Email"] != DBNull.Value ? row["Email"].ToString() :
string.Empty;
        userInfo.RegTime = Convert.ToDateTime(row["RegTime"]);
    }
    /// <summary>
    /// 返回一条记录
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    public UserInfo GetEntityModel(int id)
    {
        string sql = "select * from UserInfo where ID=@ID";
        DataTable da = SqlHelper.GetTable(sql, CommandType.Text, new
SqlParameter("@ID", id));
        UserInfo userInfo = null;
        if (da.Rows.Count > 0)
        {
            userInfo = new UserInfo();
            LoadEntity(da.Rows[0], userInfo);
        }
        return userInfo;
    }
}

```

```

    /// <summary>
    /// 删除一条记录
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    public int DeleteEntityModel(int id)
    {
        string sql = "delete from UserInfo where ID=@ID";
        return SqlHelper.ExecuteNonQuery(sql, CommandType.Text, new
SqlParameter("@ID", id));
    }
    /// <summary>
    /// 插入一条记录
    /// </summary>
    /// <param name="userInfo"></param>
    /// <returns></returns>
    public int InsertEntityModel(UserInfo userInfo)
    {
        string sql = "insert into UserInfo(UserName,UserPass,Email,RegTime)
values(@UserName,@UserPass,@Email,@RegTime)";
        SqlParameter[] pars = {
            new SqlParameter("@UserName",SqlDbType.NVarChar,32),
            new SqlParameter("@UserPass",SqlDbType.NVarChar,32),
            new SqlParameter("@Email",SqlDbType.NVarChar,32),
            new SqlParameter("@RegTime",SqlDbType.DateTime)
        };
        pars[0].Value = userInfo.UserName;
        pars[1].Value = userInfo.UserPass;
        pars[2].Value = userInfo.Email;
        pars[3].Value = userInfo.RegTime;
        return SqlHelper.ExecuteNonQuery(sql, CommandType.Text, pars);
    }
    /// <summary>
    /// 更新一条记录
    /// </summary>
    /// <param name="userInfo"></param>
    /// <returns></returns>
    public int UpdateEntityModel(UserInfo userInfo)
    {
        string sql = "update UserInfo set
UserName=@UserName,UserPass=@UserPass,Email=@Email,RegTime=@RegTime
where ID=@ID";
        SqlParameter[] pars = {
            new SqlParameter("@UserName",SqlDbType.NVarChar,32),

```

```

        new SqlParameter("@UserPass", SqlDbType.NVarChar, 32),
        new SqlParameter("@Email", SqlDbType.NVarChar, 32),
        new SqlParameter("@RegTime", SqlDbType.DateTime),
        new SqlParameter("@ID", SqlDbType.Int, 4)
    };

    pars[0].Value = userInfo.UserName;
    pars[1].Value = userInfo.UserPass;
    pars[2].Value = userInfo.Email;
    pars[3].Value = userInfo.RegTime;
    pars[4].Value = userInfo.ID;
    return SqlHelper.ExecuteNonQuery(sql, CommandType.Text, pars);
}
}

```

在 DAL 新建类 UserInfoService.cs

```

public class UserInfoService
{
    //DAL.UserInfoDal UserInfoDal = new DAL.UserInfoDal();//问题：业务层与数据层紧耦合。
    // IDAL.IUserInfoDal UserInfoDal = new DAL.UserInfoDal();//变化点。
    IDAL.IUserInfoDal UserInfoDal = DALFactory.FactoryClass.CreateUserInfoDal();
    /// <summary>
    /// 列表
    /// </summary>
    /// <returns></returns>
    public List<UserInfo> GetEntityList()
    {
        return UserInfoDal.GetEntityList();
    }
    /// <summary>
    /// 返回一条记录
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    public UserInfo GetModel(int id)
    {
        return UserInfoDal.GetEntityModel(id);
    }

    /// <summary>
    /// 删除
    /// </summary>
    /// <param name="id"></param>

```

```

    /// <returns></returns>
    public bool DeleteEntity(int id)
    {
        return UserInfoDal.DeleteEntityModel(id)>0;
    }
    /// <summary>
    /// 插入
    /// </summary>
    /// <param name="userInfo"></param>
    /// <returns></returns>
    public bool InsertEntity(UserInfo userInfo)
    {
        return UserInfoDal.InsertEntityModel(userInfo)>0;
    }
    /// <summary>
    /// 更新
    /// </summary>
    /// <param name="userInfo"></param>
    /// <returns></returns>
    public bool UpdateEntity(UserInfo userInfo)
    {
        return UserInfoDal.UpdateEntityModel(userInfo)>0;
    }
}

```

在 WebApp 添加一般处理程序与模板

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title></title>
    <link href="css/tableStyle.css" rel="stylesheet" />
    <script src="js/jquery-1.7.1.js"></script>
    <script type="text/javascript">
        $(function () {
            $("a:contains('删除')").click(function () {
                if (!confirm("确定要删除吗?")) {
                    return false;
                }
            });
        });
    
```

```

    });
</script>
</head>
<body>
    <a href="AddUserInfo.html">添加用户</a>
    <table>
        <tr>
            <th> 编 号 </th><th> 用 户 名 </th><th> 密 码 </th><th> 时 间 </th><th> 邮 箱
</th><th>详细</th><th>删除</th><th>编辑</th>
        </tr>
        $tbody
    </table>
</body>
</html>

```

```

public void ProcessRequest(HttpContext context)
{
    context.Response.ContentType = "text/html";
    BLL.UserInfoService UserInfoService = new BLL.UserInfoService();
    List<UserInfo> list=UserInfoService.GetEntityList();
    StringBuilder sb = new StringBuilder();
    foreach (UserInfo userInfo in list)
    {
        sb.AppendFormat("<tr><td>{0}</td><td>{1}</td><td>{2}</td><td>{3}</td><td>{4}</td><t
d><a href='ShowDetail.ashx?id={5}'> 详 细 </a></td></tr>", userInfo.ID, userInfo.UserName,
userInfo.UserPass, userInfo.RegTime.ToShortDateString(), userInfo.Email, userInfo.ID);
    }
    string filePath = context.Request.MapPath("UserInfoList.html");
    string fileContent = File.ReadAllText(filePath);
    fileContent= fileContent.Replace("$tbody",sb.ToString());
    context.Response.Write(fileContent);
}

```