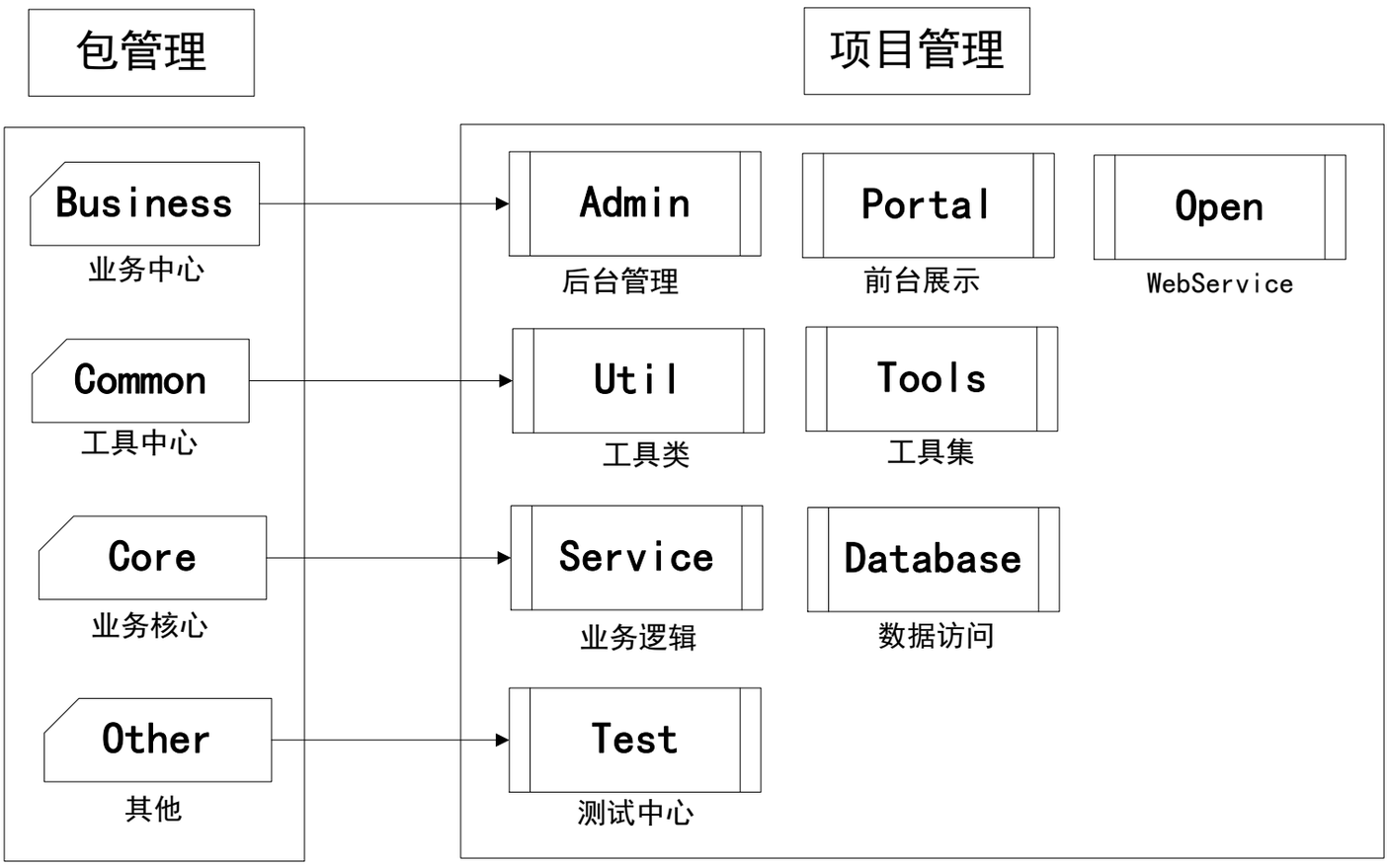
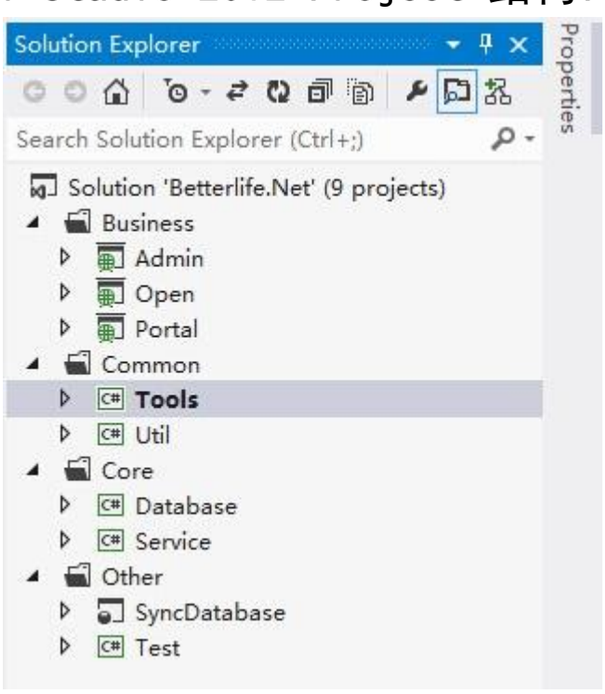
网站部署结构图



详情描述:

*.整个解决方案分为四个包：Business|Common|Core|Other
*.Business包括项目：Admin、Portal和Open;
Admin是后台管理平台，
Portal是前台网站，是整个解决方案的核心所在，其他的项目都是围绕它进行服务的。
Open是提供给第三方的Webservice，暂未投入实用
*.Common包括项目：Util、Tools;
Util是工具类，提供给所有项目可以使用的通用工具类。
Tools是开发者工具，提供给开发者使用的工具，提高开发效率，数据快速导入、规格生成的工具。
如代码生成、数据模型文档生成，从Mysql移植到Sqlserver的工具等等。
*.Core包括项目：Service和Database
Service是业务逻辑，所有核心的业务逻辑都通过它处理，它是商务的核心逻辑
Database是数据访问层，是Dao，数据库的数据读写通过它处理完成
*.Other包括项目：Test
Test是测试中心，包括所有的单元测试、功能测试等。

Visual Studio 2012 Project 结构:

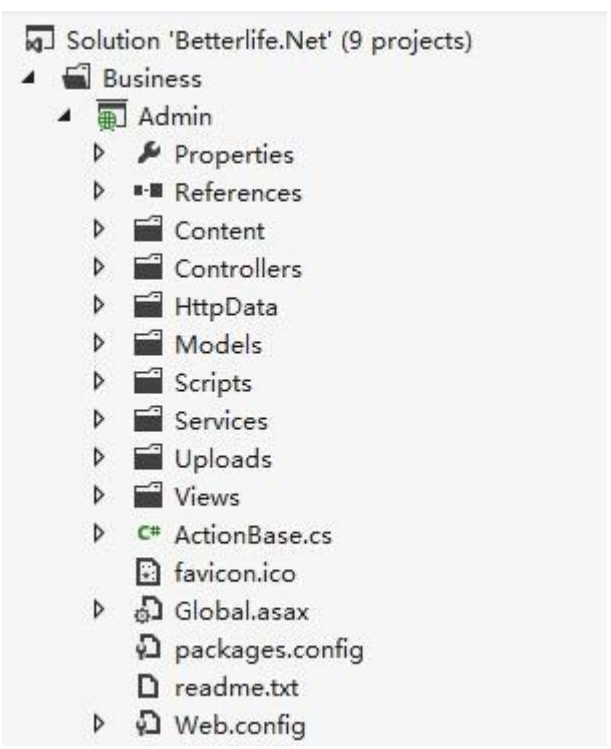


后台代码结构说明

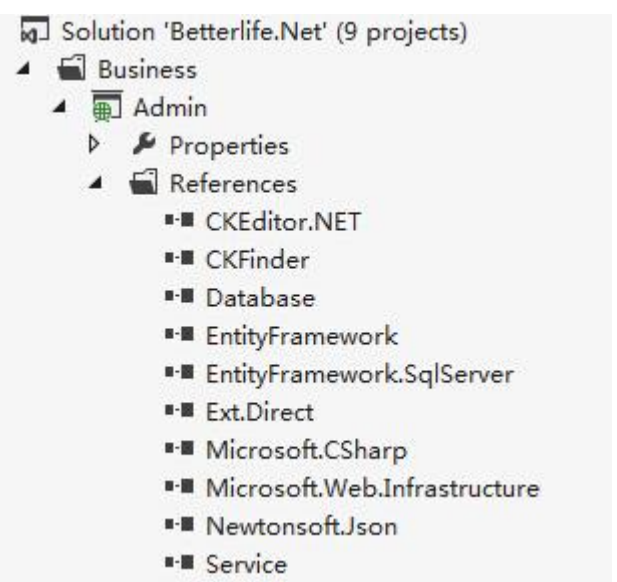
Admin

后台管理

后台目录结构：



引用如下：



Service + Dao + DB, Dao层采用了Entity Framework，数据模型和Dao都在Database项目里

在后台管理里，目录采用标准的MVC目录结构：

- Content : Css和Images文件都放置在这里，第三方的JS和Css也放置在这里
- Scripts : Js脚本文件
- Controllers: 控制器层
- Models : 类似VO，只要和表示层页面的显示数据一一对应
- Views : 显示层页面

新增目录如下：

- Services : ExtJS中通过Ajax请求直接访问的服务
- Upload : 后台上传图片的目录路径

从前端到后台访问的途径：

下面以查询系统管理员分页列表为例，描述从前端到后端整个调用的流程过程。

服务端

Database

数据访问

```
[DirectAction("ExtServiceAdmin")]
```

ExtServiceAdminHandler

通过Ajax请求直接访问的服务

C#片段：

```
public ExtServiceAdminHandler queryPageAdmin(Dictionary<String, Object> condition)
```

类名 注解名称	方法名
ExtServiceAdmin	queryPageAdmin

客户端

admin.js

通过Ajax请求直接访问服务

Views/Home/Admin.cshtml

Ext.Direct方式

ExtServiceAdmin.queryPageAdmin

Js代码片段：

```
/**
 * 查询符合条件的系统管理人员
 */
doSelectAdmin: function () {
    if (this.topToolbar) {
        var aUsername = this.topToolbar.aUsername.getValue();
        var aRealname = this.topToolbar.aRealname.getValue();
        this.filter = { 'Username': aUsername, 'Realname': aRealname };
    }
    var condition = { 'start': 0, 'limit': Bn.Admin.Config.PageSize };
    Ext.apply(condition, this.filter);
    ExtServiceAdmin.queryPageAdmin(condition, function (provider, response) {
        if (response.result && response.result.data) {
            var result = new Array();
            result['data'] = response.result.data;
            result['totalCount'] = response.result.totalCount;
            Bn.Admin.Store.adminStore.loadData(result);
        } else {
            Bn.Admin.Store.adminStore.removeAll();
            Ext.Msg.alert('提示', '无符合条件的系统管理人员！');
        }
    });
},
```

后台代码详细说明

在Web.conf文件里引用如下配置：

```
<sectionGroup name="system.web.extensions" type="System.Web.Configuration.SystemWebExtensionsSectionGroup, Sy
  <sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup, System.Web.Extensions,
    <section name="scriptResourceHandler" type="System.Web.Configuration.ScriptingScriptResourceHandlerSectio
    <sectionGroup name="webServices" type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.
      <section name="jsonSerialization" type="System.Web.Configuration.ScriptingJsonSerializationSection, Sys
    </sectionGroup>
  </sectionGroup>
</sectionGroup>
```

在表示层布局文件Views/Shared/_Layout.cshtml里，引用了Extjs库文件：

```
<script type="text/javascript" src="@Url.Content("~/Content/common/js/ajax/ext/adapter/ext/ext-base.js")"></script>
<script type="text/javascript" src="@Url.Content("~/Content/common/js/ajax/ext/ext-all-debug-w-comments.js")"></script>
<script type="text/javascript" src="@Url.Content("~/Content/common/js/ajax/ext/locale/ext-lang-zh_CN.js")"></script>
<script type="text/javascript" src="@Url.Content("~/Scripts/components/FileUploadField.js")"></script>
```

在表示层文件Views/Home/Admin.cshtml里，引用了admin.js文件：

```
<script type="text/javascript" src="@Url.Content("~/Services/ExtServiceAdmin.ashx")"></script>
<script type="text/javascript" src="@Url.Content("~/Scripts/core/admin.js")"></script>
```

在admin.js文件里：

Js代码片段：

```
/**
 * 查询符合条件的系统管理人员
 */
doSelectAdmin: function () {
    if (this.topToolbar) {
        var aUsername = this.topToolbar.aUsername.getValue();
        var aRealname = this.topToolbar.aRealname.getValue();
        this.filter = { 'Username': aUsername, 'Realname': aRealname };
    }
    var condition = { 'start': 0, 'limit': Bn.Admin.Config.PageSize };
    Ext.apply(condition, this.filter);
    ExtServiceAdmin.queryPageAdmin(condition, function (provider, response) {
        if (response.result && response.result.data) {
            var result = new Array();
            result['data'] = response.result.data;
            result['totalCount'] = response.result.totalCount;
            Bn.Admin.Store.adminStore.loadData(result);
        } else {
            Bn.Admin.Store.adminStore.removeAll();
            Ext.Msg.alert('提示', '无符合条件的系统管理人员！');
        }
    });
},
```

在ExtServiceAdmin.ashx文件里：

在类上定义注释DirectAction名称:ExtServiceAdmin，Js通过该名称调用指定的函数：queryPageAdmin

C#代码片段：

```
[DirectAction("ExtServiceAdmin")]
public class ExtServiceAdminHandler : ExtServiceBasic
{}
```

C#代码片段：

```
/// <summary>
/// 分页方法:系统管理员
/// </summary>
/// <see cref="http://diaosbook.com/Post/2012/9/21/linq-paging-in-entity-framework"/>
/// <param name="condition"></param>
/// <returns></returns>
[DirectMethod]
public ExtServiceAdminHandler queryPageAdmin(Dictionary<String, Object> condition)
```

在ExtServiceAdmin里通过以下代码获取数据环境上下文：

```
protected static BetterlifeNetEntities db = DatabaseCenter.Instance();
```

在queryPageAdmin方法里通过以下代码获取数据返回json数据到客户端js文件：

C#代码片段：

```
/// <summary>
/// 分页方法:系统管理员
/// </summary>
/// <see cref="http://diaosbook.com/Post/2012/9/21/linq-paging-in-entity-framework"/>
/// <param name="condition"></param>
/// <returns></returns>
[DirectMethod]
public ExtServiceAdminHandler queryPageAdmin(Dictionary<String, Object> condition)
{
    int limit = 10;
    int start = 0;

    int RowCount = 0;
    RowCount = db.Admin.Count();

    var admins = db.Admin.OrderByDescending(p => p.ID).Skip(start).Take(limit);
    List<Administrator> listAdmins=admins.ToList<Administrator>();
    foreach (Administrator row in listAdmins)
    {
        //row.Department = null;
        this.Stores.Add(row);
    }
    this.TotalCount = RowCount;
    this.Success = true;
    return this;
}
```

在上文提到的admin.js文件里，对返回的Json数据进行处理以列表的形式显示出来：

Js代码片段：

```
/**
 * 查询符合条件的系统管理人员
 */
doSelectAdmin: function () {
    if (this.topToolbar) {
        var aUsername = this.topToolbar.aUsername.getValue();
        var aRealname = this.topToolbar.aRealname.getValue();
        this.filter = { 'Username': aUsername, 'Realname': aRealname };
    }
    var condition = { 'start': 0, 'limit': Bn.Admin.Config.PageSize };
    Ext.apply(condition, this.filter);
    ExtServiceAdmin.queryPageAdmin(condition, function (provider, response) {
        if (response.result && response.result.data) {
            var result = new Array();
            result['data'] = response.result.data;
            result['totalCount'] = response.result.totalCount;
            Bn.Admin.Store.adminStore.loadData(result);
        } else {
            Bn.Admin.Store.adminStore.removeAll();
            Ext.Msg.alert('提示', '无符合条件的系统管理人员！');
        }
    });
},
```

最后，显示页面如下：

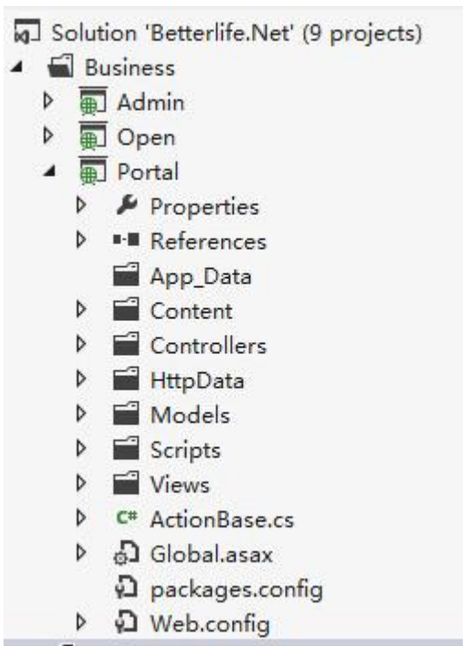
前台代码结构说明

Portal

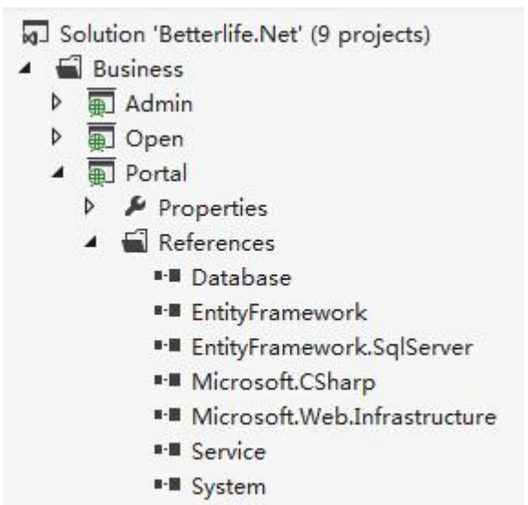
前台展示

Service + Dao + DB, Dao层采用了Entity Framework，数据模型和Dao都在Database项目里

前台目录结构：



引用如下：



在前端展示项目里，目录采用标准的MVC 3目录结构：

Content   ：Css和Images文件都放置在这里，第三方的JS和Css也放置在这里
Scripts   ：Js脚本文件
Controllers：控制器层
Models   ：类似VO，只要和表示层页面的显示数据一一对应
Views   ：显示层页面

在前端展示项目里， Controllers：控制器层调用Service层：
基本上我们会把Bo定义好Interface，把传入参数与返回值都定义好，并且会把注解都写好，并且会在Method中写好逻辑的步骤

下面的代码就是从Service层做好业务逻辑后，接受的数据信息：

```
/// <summary>
/// 服务:系统管理员
/// </summary>
public class ServiceAdmin:ServiceBasic
{
    /// <summary>
    /// 添加系统管理员
    /// </summary>
    /// <param name="username">用户名称</param>
    /// <param name="password">密码</param>
    /// <param name="roletype">角色</param>
    /// <param name="department_id">部门标识</param>
    /// <returns></returns>
    public bool addAdmin(string username, string password, int roletype, int department_id)
    {
        //1.确认用户名和密码是否为空，如果为空，返回false
        //2.查看部门标识的部门是否存在，如果不存在，标识为普通部门
        //3.查看角色类型是否定义过，如果没有，标识为普通用户
        //4.密码需要进行加密，采用md5不可逆编码
        //5.确认用户名称是否已经使用过，如果已经使用过，返回false
        return true;
    }
}
```

前端用户体验JS框架都统一使用JQuery采用Ajax无刷新的方式进行交互