## Restful服务调用



Restful Service清单，请大家一定要即时更新。

TFS路径：$/ECCentral/02\_Design

### Restful Service Web Method选用及restClient调用方法

1. Get   Portal: restClient.Query

什么时候使用Get：

* 没有条件或者条件是稳定的，这是最重要的先决条件
* 稳定的条件，最多不能超过5个；建议3个以上采用Post方法。

2. Post   Portal: restClient.Query  或  restClient. QueryDynamicData

查询、创建、综合性业务处理，一般使用Post。

3. Put   Portal: restClient.Update

更新数据，对业务处理并发时只关注最后一次处理，使用Put。

4. Delete Portal: restClient.Delete

什么时候用Delete：服务端是非常明确的，并且仅仅是处理物理删除时，采用delete

### Restful URL规则

我们并不参照标准RESTful URL规则，而是自定义一套简单、明确的URL规则，规则如下：

URL构成：

BaseURL+      Service路由 +      URL二级路径 + URL三级路径 （如果是Get方法，则在三级路径后面加GET的参数）

例如：

<http://localhost:778> +  CustomerService     +       Customer +      Query       ——Query Customer的URL

<http://localhost:778> +  CustomerService     +       Customer +       Create    ——创建Customer

<http://localhost:778> +  CustomerService     +       Customer +       Update   ——更新Customer

注：在Portal端，BaseURL是配置在Silverlight配置项里的，这样方便部署时修改。



* 服务接口唯一：CustomerService
* URL二级路径（路由后的分类路径）：

Customer

Gift

Point

Visit

CSTool

…

* URL三级路径（二级路径下具体的行为），如：

Query

Create

Update

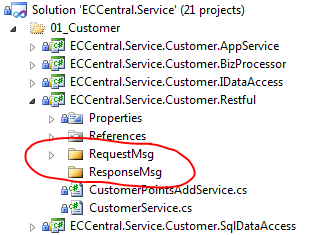
Void

…

### 传输数据Data Contract的规则

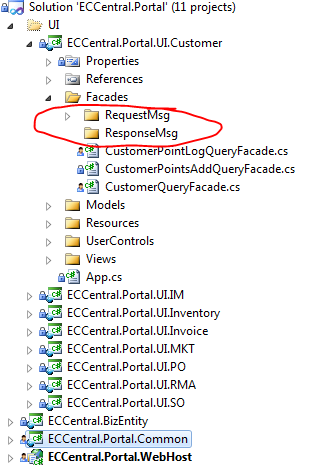
**原则：**如果能够用BizEntity来在Service端和Client端进行传输（即DataContract），那么则尽量直接使用BizEntity来作为Data Contract；只有当BizEntity无法满足使用时，才定义Request Message和Response Message。

**自定义Request /Response Message：**



Service端，Msg是定义在Restful这层的，其他层根本不用关心message。

在Portal这段，**Msg是定义在Facades里面的**，因为Portal这边也仅仅是Facade需要知道Msg，对于View，ViewModel是不需要知道Msg的，只会有Facade来协调将Msg里的数据fill到ViewModel里就ok了。



### 调用Restful服务示例

统一使用ECCentral.Portal.Common.Utilities的RESTClient对象，初始化时传入RESTful接口的Base URL作为参数，所有Domain Service的Base URL都必须按照Domain在Control Panel > Settings > Silverlight Config 中进行配置。

例如：

|  |  |  |
| --- | --- | --- |
|  | **Method** | **Service URL** |
| GetCustomerInfo | [**Get**](http://10.16.230.42/NESO.CustomerCallService/CustomerCallService/help/operations/GetCallReasonTree) | http://localhost:778/CustomerService/Customer/{customerid} |

红色为BaseUrl，蓝色为路由，紫色为相对路径。

需要在Silverlight Configuration中配置BaseUrl：



// Declare Client

string customerCallServiceServiceBase = Window.Configuration.GetConfigValue("Customer", "ServiceBaseUrl");

string routePrefix = "CustomerService";

string preUrl= customerCallServiceServiceBase + routePrefix ;

RestClient restClient = new RestClient(preUrl, this);

// Query CallReasonNotes with POST method

string relativeUrl = string.format("/Customer/{0}",customerID);

restClient.Query<CustomerInfo>( relativeUrl, (obj, result) =>{});