影像云 Open API 编码规范

Author: 祝建军 Mail: [zhujj@tomtaw.com.cn](mailto:zhujj@tomtaw.com.cn)

# 概述

影像云Open API，亦称为能力开放平台，是公司产品重要战略中面向企业服务的业务开放平台。

# 定义

## 能力

基于NopCommerce框架的插件机制，开发人员可以创建插件以实现一组能力。通常，也称为能力分组。

每个插件中必须创建文档Description.txt，用于描述插件，同时，能力开放系统也从中获取Group属性用于标识一组能力，各属性引用如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **引用为** | **描述** | **示例** |
| Group | 能力名称 | 标识一组能力 | 远程医疗 |
| SystemName | 能力代码 | 提供给开发者申请应用的能力代码 | WebAPI.Clinic |
| DisplayOrder | 能力排序号 | 用于API文档中心的排序 |  |

## 子能力

基于Asp.NET WebApi，开发人员可以通过创建控制器（Controller）实现子能力，其名称引用为子能力的代码（SubCode）。

# 授权认证

第一要则：所有接口必须定义授权方式。

开发人员可以在API的Controller上使用ApiClientAuthorize声明控制器下所有Action的授权方式，亦可以在单个Action上声明。

Action的授权方式声明优先于Controller，也就是说，Action的声明将覆盖Controller上的。

通常，我们可以在Controller上声明全局的授权方式，然后在Action上声明与Controller不同的授权方式，以达到快速声明授权方式的目的。当然，开发人员可以自行区分全局的授权方式。

## 用户授权

与当前登录用户息息相关的接口，必须采用该授权方式，其定义如下：

eWorld.Web.Framework.OpenAPI.ApiUserAuthorizeAttribute

## 客户端授权

与当前登录用户没有任何业务交互的接口，可以采用该授权方式，其定义如下：

eWorld.Web.Framework.OpenAPI.ApiClientAuthorizeAttribute

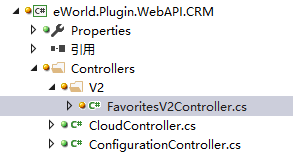
需要注意的是，开发人员必须采用上述两种授权定义，而不是其它的。

## 授权响应

|  |  |  |
| --- | --- | --- |
| HTTP状态码 | 响应内容 | 说明 |
| 200 |  | 授权通过 |
| 401 | {code:60001, msg: “未经授权的请求。”} | 未授权情况下访问接口 |
| {code: 60001, msg: “未开通的应用-{应用id} 能力-{能力代码}:{子能力代码}”} | 用户访问了未开通的应用能力 |
| {code: 60001, msg: “未经授权的应用-{应用id} 能力-{能力代码}:{子能力代码}”} | 应用的能力没有通过审核 |
| {code:60001, msg: “非法客户端请求。”} | 客户端授权的clientid无法匹配应用 |
| 403 | {code:10001, msg: “无效的授权模式，请使用用户授权模式授权。”}  {code:10001, msg: “无效的授权模式，请使用客户端模式授权。”} | 使用非接口要求的授权模式访问接口，比如接口要求用户授权，而请求时使用了客户端授权 |

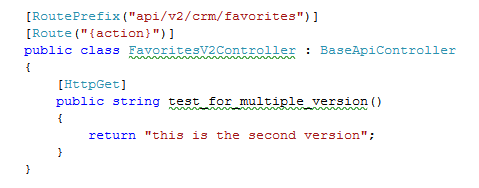
# 版本管理

得益于Asp.Net Web API的易用性，系统中提供了非常简便的版本管理方式，以满足不同版本的项目能够调用相应版本的API。其管理方式如下图：



当需要进行进行多版本管理时，开发人员需要在Controllers目录下创建对应的版本号，并将能力的实现归于所创建的版本号目录当中。需要特别注意，控制器类的命名空间必须包含版本号目录（如，eWorld.API.Controllers.V2），系统依赖于它输出相应版本的API文档。

多版本的路由定义参考《路由定义》一节。参考如下：



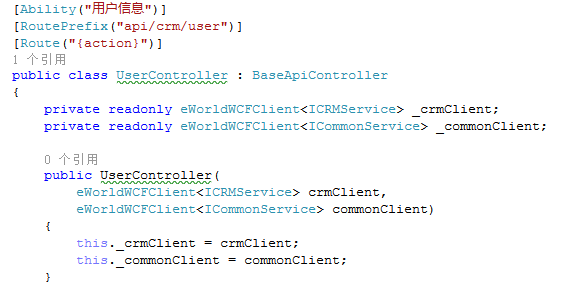
# 路由定义

Asp.Net Web API 提供了基于特性（RouteAttribute）定义路由的方式，开发人员通过该特性即可生成RESTful风格的API请求地址。

有关RESTful的更多信息：<https://www.restapitutorial.com/>。

有关RouteAttribute的更多信息：<https://docs.microsoft.com/en-us/aspnet/web-api/overview/web-api-routing-and-actions/attribute-routing-in-web-api-2>

在能力开放平台中，开发人员必须参照下图定义路由：



RoutePrefix，用于定义当前控制器（Controller）的路由模板前缀，其中，“crm”表示能力分组的区域（Area）命名，“user”表示子能力的命名。

在控制器（Controller）上声明RouteAttribute，即定义了全局的操作路由，控制器中所有操作（Action）都将使用该模板生成最终的API请求地址。

如需覆盖全局的路由模板定义，只需在操作（Action）上声明RouteAttribute即可。

需要注意的是，路由定义必须采用小写模式。

# 接口命名

开发人员应严格按照RESTful命名风格，在遵守编码规范的同时，合理命名接口。

编码规范中要求，所有接口的命名采用小写模式，单词之间使用下划线（\_）连接。

RESTful命名风格则要求，接口的命名不得包含动词，动词表示行为，而URI描述的是一种资源。因此，开发人员应该使用准确的HTTP Methods声明接口。

接口的输入和输出参数的命名也必须遵守以上规范。

# 文档输出

框架中集成了Swagger中间件，用于输出API文档。

有关Swagger中间件的更多信息：<https://github.com/domaindrivendev/Swashbuckle>

Swagger依赖于代码的注解（注释），所以，项目必须输出XML注解文档。

也就是说，开发人员必须为每个接口编写摘要、详细描述、参数描述、返回值描述、响应码。

以往，除了维护接口的注解，开发人员还需要同时维护eoLinker上的接口文档，之后就可以避免重复工作了。

# 能力清单

框架上提供了应用管理所需的业务服务数据。通过访问API地址“/abilities/v1“即可获取能力的列表，需要获取不同版本的能力列表时，将地址中的v1替换为对应版本即可。