**创建动态Web Api 控制器**

ABP可为你的应用层自动创建Asp.net Web Api层，假设我们有一个如下所示的[应用服务](http://www.cnblogs.com/kid1412/p/6003490.html" \t "http://www.cnblogs.com/kid1412/p/_blank)：

public interface ITaskAppService : IApplicationService

{

GetTasksOutput GetTasks(GetTasksInput input);

void UpdateTask(UpdateTaskInput input);

void CreateTask(CreateTaskInput input);

}

自动动态创建一个Web Api控制器：

Configuration.Modules.AbpWebApi().DynamicApiControllerBuilder.For<ITaskAppService>("**tasksystem/task**").Build();

这就完事了。一个api控制器被创建在地址“/api/services/tasksystem/task”下，这个配置应当在你的[模块](http://www.cnblogs.com/kid1412/p/5973274.html" \t "http://www.cnblogs.com/kid1412/p/_blank)的[Initialize](http://www.cnblogs.com/kid1412/p/5973274.html" \l "Initialize" \t "http://www.cnblogs.com/kid1412/p/_blank)方法里完成。

**ForAll 方法**

DynamicApiControllerBuilder提供了一个为所有应用服务创建web api控制器一次调用的方法，例如：

Configuration.Modules.AbpWebApi().DynamicApiControllerBuilder

.ForAll<IApplicationService>(Assembly.GetAssembly(typeof(SimpleTaskSystemApplicationModule)), "tasksystem")

.Build();

ForAll方法是一个接受一个接口的泛型，第一个参数是程序集，第二个服务的命名空间前缀，假设我们有一个包含ITaskAppService和IPersonAppService的程序集，应用上述配置，服务将是：“/api/services/**tasksystem/task”** “/api/services/**tasksystem/person”**。

**ForMethod**

当使用ForAll方法时，我们可用ForMethods方法进行更好的调用 ，例如：

Configuration.Modules.AbpWebApi().DynamicApiControllerBuilder

.ForAll<IApplicationService>(Assembly.GetExecutingAssembly(), "app")

.**ForMethods**(builder =>

{

if (builder.Method.IsDefined(typeof(MyIgnoreApiAttribute)))

{

builder.**DontCreate** = true;

}

})

.Build();

在这个示例里，用一个自定义特性（MyIgnoreApiAttribute）来检查所有方法，不为标记了这个特性的方法创建动态web api控制器。

**Http 动词**

默认地，所有方法被创建成POST，所以一个客户端应用发送post请求来使用已创建的web api的actions。我们可以把这种行为修改成不同的方式。

**HTTP 特性**

在服务接口里，我们可以给方法添加HttpGet、HttpPost等特性：

public interface ITaskAppService : IApplicationService

{

**[HttpGet]**

GetTasksOutput GetTasks(GetTasksInput input);

**[HttpPut]**

void UpdateTask(UpdateTaskInput input);

**[HttpPost]**

void CreateTask(CreateTaskInput input);

}

为使用这些特性，应该在项目里添加Microsoft.Asp.net.WebApi.Core的nuget包，并引用它。

**命名约定**

你可以使用WithconvertionalVerbs方法，代替为每个方法声明HTTP动词，如下所示：

Configuration.Modules.AbpWebApi().DynamicApiControllerBuilder

.ForAll<IApplicationService>(Assembly.GetAssembly(typeof(SimpleTaskSystemApplicationModule)), "tasksystem")

**.WithConventionalVerbs()**

.Build();

这种情况下，以方法名的前缀确定HTTP动词：

* Get：如果方法名以“Get”打头。
* Put：如果方法名以“Put”或“Update”打头。
* Delete： 如果方法名以“Delete”或“Remove”打头。
* Post：如果方法名以“Post”或“Create”或“Insert”打头。
* Patch：如果方法名以“Patch”打头。
* 否则：以Post作为默认的HTTP动词。

**包装结果**

ABP通过AjaxResponse对象包装动态Web Api的Action返回值，更多信息查看[ajax文档](http://www.cnblogs.com/kid1412/p/6012304.html" \t "http://www.cnblogs.com/kid1412/p/_blank)。你可以为每个方法或每个应用服务启用/禁用包装，请看如下应用服务示例：

public interface ITestAppService : IApplicationService

{

**[DontWrapResult]**

DoItOutput DoIt(DoItInput input);

}

我们为DoIt方法禁用了包装，这个属性要声明在接口里，而不是实现的类里。

**FormUri 和 FormBody 特性**

FormUri和FormBody特性可用在服务接口里，优先控制绑定。