**暴露应用服务为API**

要想将应用服务暴露为API可以在展示层或API层执行如下代码

[DependsOn(BookStoreApplicationModule)]

public class BookStoreWebModule : AbpModule

{

    public override void ConfigureServices(ServiceConfigurationContext context)

    {

        Configure<AbpAspNetCoreMvcOptions>(options =>

        {

            options

                .ConventionalControllers

                .Create(typeof(BookStoreApplicationModule).Assembly);

        });

    }

}

**API名称与Http方法约定**

如下是AppSerivce方法的约定，如果我们没有指定[Route]和[HttpXXX]特性，那么将按照如下约定生成URL Route

|  |  |  |
| --- | --- | --- |
| GetAsync(Guid id) | GET | /api/app/book/{id} |
| GetListAsync() | GET | /api/app/book |
| CreateAsync(CreateBookDto input) | POST | /api/app/book |
| UpdateAsync(Guid id, UpdateBookDto input) | PUT | /api/app/book/{id} |
| DeleteAsync(Guid id) | DELETE | /api/app/book/{id} |
| GetEditorsAsync(Guid id) | GET | /api/app/book/{id}/editors |
| CreateEditorAsync(Guid id, BookEditorCreateDto input) | POST | /api/app/book/{id}/editor |

**路由前缀**

默认的路由前缀为/api（必须的）+/app，我们可以自定义暴露的应用服务的路由前缀

Configure<AbpAspNetCoreMvcOptions>(options =>

{

    options.ConventionalControllers

        .Create(typeof(BookStoreApplicationModule).Assembly, opts =>

            {

                opts.RootPath = "volosoft/book-store";

            });

});

**RemoteService特性**

RemoteService可以指定AppService是否暴露为Api

[RemoteService(IsEnabled = false)] //or simply [RemoteService(false)]

public class PersonAppService : ApplicationService

{

}