Asp.net 5.x

**安装**

基于应用，你可以很方便的把Swagger集成到你的ABP里。

**安装Nuget包**

在你的WebApi项目里安装[Swashbuckle.core](https://www.nuget.org/packages/Swashbuckle/6.0.0-beta902)包。

**配置**

为Swagger添加配置代码你的模块的到[Initialize](http://www.cnblogs.com/kid1412/p/5973274.html" \l "Initialize" \t "http://www.cnblogs.com/kid1412/p/_blank)方法里，例如：

public class SwaggerIntegrationDemoWebApiModule : AbpModule

{

public override void Initialize()

{

//your other code...

**ConfigureSwaggerUi();**

}

**private void ConfigureSwaggerUi()**

**{**

**Configuration.Modules.AbpWebApi().HttpConfiguration**

**.EnableSwagger(c =>**

**{**

**c.SingleApiVersion("v1", "SwaggerIntegrationDemo.WebApi");**

**c.ResolveConflictingActions(apiDescriptions => apiDescriptions.First());**

**})**

**.EnableSwaggerUi(c =>**

**{**

**c.InjectJavaScript(Assembly.GetAssembly(typeof(AbpProjectNameWebApiModule)), "AbpCompanyName.AbpProjectName.Api.Scripts.Swagger-Custom.js");**

**});**

**}**

}

注意：在配置Swagger ui时我们注入了一个名为“Swagger-Custom.js”的javascript文件，这个脚本文件在从Swagger ui里测试api服务时，添加CSRF令牌，同时你也需要在你的WebApi项目里添加这个文件，当注入它时，使用它的逻辑名称来注入javascript方法，它的内容如下所示：

var getCookieValue = function(key) {

var equalities = document.cookie.split('; ');

for (var i = 0; i < equalities.length; i++) {

if (!equalities[i]) {

continue;

}

var splitted = equalities[i].split('=');

if (splitted.length !== 2) {

continue;

}

if (decodeURIComponent(splitted[0]) === key) {

return decodeURIComponent(splitted[1] || '');

}

}

return null;

};

var csrfCookie = getCookieValue("XSRF-TOKEN");

var csrfCookieAuth = new SwaggerClient.ApiKeyAuthorization("X-XSRF-TOKEN", csrfCookie, "header");

swaggerUi.api.clientAuthorizations.add("X-XSRF-TOKEN", csrfCookieAuth);

Asp.net Core

**安装**

基于Asp.net Core应用，你可以很方便的把Swagger集成到你的ABP里。

**安装Nuget包**

在你的Web项目里安装[Swashbuckle](https://www.nuget.org/packages/Swashbuckle/6.0.0-beta902)包。

**配置**

在你的Startup.cs文件里，在ConfigureServices方法里添加代码，配置入Swagger：

public IServiceProvider ConfigureServices(IServiceCollection services)

{

//your other code...

**services.AddSwaggerGen();**

//your other code...

}

然后，为使用Swagger，在Startup.cs文件里，在Configure方法里添加如下代码：

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)

{

//your other code...

**app.UseSwagger();**

**app.UseSwaggerUi(); //URL: /swagger/ui**

}

最后 ，当从Swagger UI测试动态Web Api服务时，为了发送CSRF令牌，你需要把Swagger UI的 [index.html](https://github.com/swagger-api/swagger-ui/blob/master/dist/index.htmlhttps://github.com/swagger-api/swagger-ui/blob/master/dist/index.html)文件添加到你的项目里，它应当放置在“**wwwroot\swagger\ui”**文件夹下，然后你需要在index.html里，修改Swagger UI的onComplete方法，如下所示：

onComplete: function(swaggerApi, swaggerUi){

if(typeof initOAuth == "function") {

initOAuth({

clientId: "your-client-id",

clientSecret: "your-client-secret-if-required",

realm: "your-realms",

appName: "your-app-name",

scopeSeparator: " ",

additionalQueryStringParams: {}

});

}

if(window.SwaggerTranslator) {

window.SwaggerTranslator.translate();

}

**var csrfToken = abp.security.antiForgery.getToken();**

**var csrfCookieAuth = new SwaggerClient.ApiKeyAuthorization(abp.security.antiForgery.tokenHeaderName, csrfToken, "header");**

**swaggerUi.api.clientAuthorizations.add(abp.security.antiForgery.tokenHeaderName, csrfCookieAuth);**

}