

# CORS与JSONP跨域请求

2019年1月10日 9:42

## ■ 什么是跨域请求

浏览器为了安全性，会防止当前域的网页发出请求到不同的域的网页，此限制称为同源策略。同源策略可阻止恶意站点读取另一个站点中的敏感数据。有时，你可能想允许其它站点对您的应用程序进行跨域请求。

跨域资源共享(CORS) 是一种 W3C 标准，允许服务器放宽同源策略。使用 CORS，服务器可以显式允许某些跨域请求时拒绝其他跨域请求。CORS 比早期的 JSONP技术要更安全、更灵活。

### HTTP访问控制 (CORS)

来自 <[https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Access_control_CORS)>

### 跨域资源共享 CORS 详解

来自 <<http://www.ruanyfeng.com/blog/2016/04/cors.html>>

## ■ 同源策略

如果两个URL具有相同方案、主机和端口，就是同源，在 [RFC 6454](#) 规范中定义。

## ■ 注册 CORS 服务

引用[Microsoft.AspNetCore.App 元包](#)或添加到的包引用[Microsoft.AspNetCore.Cors](#)包。

Install-Package Microsoft.AspNetCore.Cors -Version 2.2.0

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCors();
}
```

## ■ 启用 CORS 服务

中间件和MVC筛选器两种方式，中间件针对所有请求，MVC筛选器针对特定控制器或者操作。

## ■ 中间件使用方式

```
app.UseCors(builder => builder.WithOrigins("https://www.xcode.me"));
```

CorsPolicyBuilder 具有 Fluent API，支持链式配置，顺便来说：Builder链式模式，Policy策略模式，Provider提供者模式。

```
options.AddPolicy("AllowSpecificOrigin", builder => builder.WithOrigins("https://www.xcode.me"));
// Shows UseCors with named policy.
app.UseCors("AllowSpecificOrigin");
```

## ■ MVC过滤器使用方式

```
[HttpGet]
[EnableCors("AllowSpecificOrigin")]
public IEnumerable<string> Get()
{
    return new string[] { "value1", "value2" };
}
```

也可以在控制器上使用，这将影响控制器中的所有 Action 操作，如果控制器和 Action 操作都有 EnableCors 策略时，基于Action 操作的策略

具有更高的优先级，若要禁用 CROS，可使用 [DisableCors] 特性。

## CORS 策略选项配置

URL来源，请求方法，请求标头，响应标头，跨域请求中的凭据和将预检过期时间设置。

## JSONP技术跨域

技术原理讲解

来自 <<https://www.jianshu.com/p/e1e2920dac95>>

.NET Core 实现 JSONP 跨域

来自 <<https://www.nuget.org/packages/WebApiContrib.Core.Formatter.Jsonp/>>

## CORS运行原理

