

强制实施 HTTPS 安全访问

2019年2月25日 13:17

■ HTTP、HTTPS 和 HSTS

来自 <<https://www.cnblogs.com/upyun/p/7447977.html>>

SSL/TLS协议运行机制的概述

来自 <http://www.ruanyifeng.com/blog/2014/02/ssl_tls.html>

■ 将部分请求定向到 HTTPS 请求

RequireHttpsAttribute.cs

来自 <<https://github.com/aspnet/AspNetCore/blob/master/src/Mvc/Mvc.Core/src/RequireHttpsAttribute.cs>>

```
services.AddMvc(options => options.SslPort = 44328)
```

■ 将全部请求定向到 HTTPS 请求

HTTPS 策略中间件

来自 <<https://github.com/aspnet/AspNetCore/tree/master/src/Middleware/HttpsPolicy/src>>

HTTPS 重定向中间件 (UseHttpsRedirection) 将 HTTP 请求重定向到 HTTPS。

HSTS 中间件 (UseHsts) 向客户端发送严格传输安全性协议 (HSTS) 标头。

反向代理配置中如果已启用HTTPS重定向和HSTS标头，将不需要使用以上中间件。

```
app.UseHttpsRedirection();
```

默认情况重定向规则：307状态码和443端口

详解各种HTTP重定向状态码区别 <<https://www.cnblogs.com/wuguanglin/p/redirect.html>>

来HTTP状态码302、303和307的故事 <<http://www.cnblogs.com/cswuyg/p/3871976.html>>

改变默认的HTTPS端口方法之一：

- 1、环境变量 ASPNETCORE_HTTPS_PORT 设置。
- 2、UseSetting("https_port", "8080")
- 3、launchsettings.json

■ HTTP代理时部署方案

需要配置反向代理，并使用转发头中间件，转接头使用反向代理传过来的 X-Forwarded-Proto 头更新 Request.Scheme，如果使用不当，将出现重定向循环。

详见零度课堂：第42期-代理转接头与发布配置 (45分钟)

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddHttpsRedirection(options =>
    {
        options.RedirectStatusCode =
        StatusCodes.Status307TemporaryRedirect;
        options.HttpsPort = 5001;
    });
}
```

在生产环境中配置永久重定向：

```
public void ConfigureServices(IServiceCollection services)
{
    // IHostingEnvironment (stored in _env) is injected into the
    Startup class.
    if (!_env.IsDevelopment())
    {
        services.AddHttpsRedirection(options =>
        {
            options.RedirectStatusCode =
            StatusCodes.Status308PermanentRedirect;
            options.HttpsPort = 443;
        });
    }
}
```

也可以通过URL重写中间件重定向到HTTPS协议：第10期-URL重定向与重写 (73分钟)

```
public void Configure(IApplicationBuilder app)
{
    var options = new RewriteOptions().AddRedirectToHttps(301,
    5001);
    app.UseRewriter(options);
}
```

虽然如此，微软还是推荐我们使用：HTTPS 重定向中间件 (UseHttpsRedirection)。

■ HTTP 严格传输安全协议 (HSTS)

HTTP Strict Transport Security

来自 <https://developer.mozilla.org/zh-CN/docs/Security/HTTP_Strict_Transport_Security>

HSTS协议详解

来自 <<https://www.jianshu.com/p/caa80c7ad45c>>

需要客户端和服务端支持，大多数浏览器都支持，服务端通过中间件添加标头来实现。

```
services.AddHsts(options =>
{
    options.Preload = true;
    options.IncludeSubDomains = true;
    options.MaxAge = TimeSpan.FromDays(60);
    options.ExcludedHosts.Add("example.com");
    options.ExcludedHosts.Add("www.example.com");
});

app.UseHsts();
```

UseHsts 不包括环回主机：localhost、127.0.0.1 和 [::1]