参考文档：

https://www.cnblogs.com/tdfblog/p/aspnet-core-security-anti-request-forgery.html

**XSRF攻击**

利用浏览器访问某一站点会发送该站点的所有Cookie的原理

站点B的form表单指向站点A，form的内容是B想执行的内容，当用户不经意间点了提交，浏览器将form发送至A，并携带A的cookie，而A的身份验证刚好是利用cookie来保存，那么B顺利执行其想执行的内容

**XSRF防护原理**

站点A在生成form的时候，添加了一个隐藏的input，其input的值是用户的身份令牌，如果form是由站点A生成的，那么提交时就会包含这个令牌，如果form不是站点A生成的，那么提交时就不会包含这个令牌

**Asp.net Core XSRF防护**

**视图**

<!-- form 不包含 action 时自动添加防伪令牌 -->

<form asp-controller="Manage" asp-action="ChangePassword" method="post">

    <input type="type" name="Id" value="1" />

    <button type="submit">提交</button>

</form>

<!-- form 包含 action 时，需手动提交访问令牌 -->

<form action="/" method="post">

    <!-- 添加访问令牌 -->

    @Html.AntiForgeryToken()

</form>

生成的input防伪令牌如下：

<input name="\_\_RequestVerificationToken" type="hidden" value="CfDJ8PPP3OSBBV9Li6U10tbktVWUQ4wmdx5XOQ2Og9Khscnu7g0s1Qvt7h70WWisZAVvIe9y-WL\_UsJWPwMes20hTAKvyuHqI1uWzJczsCt55P1uweQPLpRwK0n5vpF4vrIGlS80396olisA-Pkb\_Wpl2U0">

**请求方法**

使用特性进行访问验证

[HttpPost]

// 防伪令牌验证

[ValidateAntiForgeryToken]

public async Task<IActionResult> RemoveLogin (RemoveLoginViewModel account)

{

}

全局防伪验证

services.AddMvc (option => {

    // 在具有 post, put, delete 请求方法上应用防伪验证

    option.Filters.Add (new AutoValidateAntiforgeryTokenAttribute ());

});