## CSRF

官方释义：跨站点伪造请求

举个例子：当你使用自己正规身份登录一个正常网站后，浏览器会记录你的身份信息。如果在这个时候你又进入了一个恶意网站，这个恶意网站上刚好又有一个链接去指向你已经登录的这个正常网站。你如果点击了这个链接，这个网站就会获取你的身份验证信息，然后通过自己的客户端使用你的身份去向你的网页发送请求，恶意操作那个正常的网页。

为了防止这个问题出现，CSRF才出现了，我们在登录正常网站的时候，CSRF会在cookie中存储一个csrftoken，可以理解为身份验证信息，这个信息通过第三方手段难以获得。当你向页面发起get请求时，Django会判断有没有使用{% csrf\_token %}如果使用了会产生一个token编码一并返回，页面再次发起post请求会判断是否有token编码，正确的话才会继续进行。有效防止第三方恶意操作。

但是如果第三方网页访问了你的页面，可以在后台手动复制你的token编码，进行请求。还是可以恶意操作的，这样只是简单的阻拦。

但是我们如果打开了csrf的话，我们在每次进行post请求的时候需要在表单内加上我们的token编码一并提交{% csrf\_token %}