**[Error: invalid csrf token](http://blog.csdn.net/renpinghao/article/details/44942711)**

标签： [bbs](http://www.csdn.net/tag/bbs)

2015-04-08 16:27 2194人阅读 [评论](http://blog.csdn.net/renpinghao/article/details/44942711#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/renpinghao/article/details/44942711#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

bbs（2） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权所有：[一介布衣](http://yijiebuyi.com/%5C%22+target=%5C%22_blank%5C%22+.=%5C%22text-decoration:+none;+color:+rgb%2851,+51,+51%29;+font-family:+Arial,+Helvetica,+sans-serif;+font-size:+12px;+line-height:+20px;)   
原文地址：[http://yijiebuyi.com/blog/74528e6546834ebfec47daf4e222d621.html](http://yijiebuyi.com/blog/74528e6546834ebfec47daf4e222d621.html%5C%22+target=%5C%22_blank%5C%22+.=%5C%22text-decoration:+none;+color:+rgb%2851,+51,+51%29;+font-family:+Arial,+Helvetica,+sans-serif;+font-size:+12px;+line-height:+20px;)   
无特别说明均为原创，欢迎各种形式转载，转载请保留此段申明。

当你的[Express](http://yijiebuyi.com/blog/38f1437bf5b43fcf90e6529a81f258f1.html%5C%22+target=%5C%22_blank%5C%22+.=%5C%22text-decoration:+none;+color:+rgb%28231,+88,+54%29;+border-bottom-width:+1px;+border-bottom-.:+dotted;+border-bottom-color:+rgb%28204,+204,+204%29;) 升级到4.0+版本后,有没有遇到过下面这个问题.

{    [Error: invalid csrf token]   expose: true,   code: 'EBADCSRFTOKEN',   statusCode: 403,   status: 403  }

上面的错误是在[**Node.js**](http://lib.csdn.net/base/nodejs) 运行环境的命令行报出来的.页面直接返回了500[服务器](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%B7%FE%CE%F1%C6%F7&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=2&seller_id=1&di=128)错误.

我执行了什么操作?

这是我的一个新站 [三思悦](http://sansiyue.com/%5C%22+target=%5C%22_blank%5C%22+.=%5C%22text-decoration:+none;+color:+rgb%28231,+88,+54%29;+border-bottom-width:+1px;+border-bottom-.:+dotted;+border-bottom-color:+rgb%28204,+204,+204%29;) ,此站有个功能是邀请注册,不对外公开注册的.

所以[管理](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%B9%DC%C0%ED&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=6&seller_id=1&di=128)后台有个页面用来显示 邀请码,同时有个按钮,用来生成验证码,一次生成唯一不重复的50个.

所以这个页面对应2个相同的url ,只不过get方式用来呈现为被使用的邀请码,post 方式用来生成50个验证码并刷新本页.

其中 post 涉及到一个 form 提交.

上面报错的 csrf 是什么? 引用百度百科的一段说法

"CSRF（Cross-site request forgery跨站请求伪造，也被称为“one click attack”或者session riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用。尽管听起来像跨站[脚本](http://baike.baidu.com/view/54.htm)（XSS），但它与XSS非常不同，并且攻击方式几乎相左。XSS利用站点内的信任用户，而CSRF则通过伪装来自受信任用户的请求来利用受信任的[网站](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%CD%F8%D5%BE&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=4&seller_id=1&di=128)。与XSS攻击相比，CSRF攻击往往不大流行（因此对其进行防范的资源也相当稀少）和难以防范，所以被认为比XSS更具危险性。"

风险:

"风险在于那些通过基于受信任的输入form和对特定行为无需授权的已[认证](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%C8%CF%D6%A4&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=3&seller_id=1&di=128)的用户来执行某些行为的web应用。已经通过被保存在用户[浏览器](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%E4%AF%C0%C0%C6%F7&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=5&seller_id=1&di=128)中的cookie进行认证的用户将在完全无知的情况下发送HTTP请求到那个信任他的站点，进而进行用户不愿做的行为。"

那么Express 中为什么要有这个 csrf ? 原因就是为了避免 上面提到的风险,防止被 CSRF攻击.

在Express 中如何开启:

var csurf = require('csurf'); //使用前要 npm install csurfapp.use(function (req, res, next) {     csurf()(req, res, next);     next(); });

下面用一个客户端公共变量来储存此值

app.use(function (req, res, next) {   res.locals.csrf = req.csrfToken ? req.csrfToken() : '';   next(); });

这样我在页面中 直接用 <%- csrf %> 就可以得到此值.

而上面的报错,正是因为post 的form 里面没有提交这个 csrfToken 值,导致上面的代码赋值时报错.

[客户端](http://cpro.baidu.com/cpro/ui/uijs.php?rs=1&u=http%3A%2F%2Fyijiebuyi%2Ecom%2Fblog%2F74528e6546834ebfec47daf4e222d621%2Ehtml&p=baidu&c=news&n=10&t=tpclicked3_hc&q=50068150_cpr&k=%BF%CD%BB%A7%B6%CB&k0=%BF%CD%BB%A7%B6%CB&kdi0=8&k1=%B7%FE%CE%F1%C6%F7&kdi1=8&k2=%C8%CF%D6%A4&kdi2=8&k3=%CD%F8%D5%BE&kdi3=8&k4=%E4%AF%C0%C0%C6%F7&kdi4=8&k5=%B9%DC%C0%ED&kdi5=8&sid=a40527cc230c54c2&ch=0&tu=u1835001&jk=3353526106ca812e&cf=29&rb=0&fv=16&stid=9&urlid=0&luki=1&seller_id=1&di=128)可以用隐藏字段保存此值

<input type='hidden' name='\_csrf' value='<%= csrf %>'/>

这样提交 Express 就不会报错了.