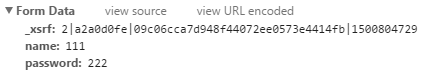
**一、1.1设置xsrf防止跨域攻击**

第一步：在Application加入配置，开启xsrf服务。

xsrf\_cookies=True,

第二步：在form表单设置{% module xsrf\_form\_html() %}

<form method="post" action="/login"> *<!--action是返回的路由映射-->* {% module xsrf\_form\_html() %}  
 <p>用户名<input type="text" name="name"></p>  
 <p>密码<input type="text" name="password"></p>  
 <input type="submit">  
</form>



**1.2 XSRF令牌和AJAX请求**

AJAX请求也需要一个\_xsrf参数，但不是必须显式地在渲染页面时包含一个\_xsrf值，而是通过脚本在客户端查询浏览器获得cookie值。下面的两个函数透明地添加令牌值给AJAX POST请求。第一个函数通过名字获取cookie，而第二个函数是一个添加\_xsrf参数到传递给postJSON函数数据对象的便捷函数。

function getCookie(name) {

var c = document.cookie.match("\\b" + name + "=([^;]\*)\\b");

return c ? c[1] : undefined;

}

jQuery.postJSON = function(url, data, callback) {

data.\_xsrf = getCookie("\_xsrf");

jQuery.ajax({

url: url,

data: jQuery.param(data),

dataType: "json",

type: "POST",

success: callback

});

}

**二、摸板继承**

把多个页面相同的内容提取出来放在一个base.html文件中，各个子html文件不同的内容 使用块语句占位，子html文件重写这个块中的内容

从父模板继承

{% extends filename %}

继承时子模板替换父模板中同名的块

{% block name %}

这里写的内容会替换父摸板中的内容

如果不写使用父模板中的内容

{% end %}

引入其它模板文件

{% include filename %}

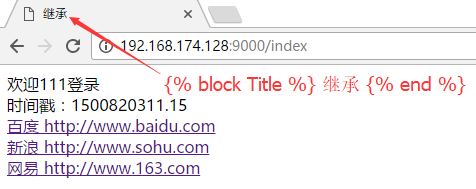
**class** IndexHandler(BaseHandler):  
 @tornado.web.authenticated  
 **def** get(self):  
 bb = [  
 (**'http://www.baidu.com'**, **u"百度"**),  
 (**'http://www.sohu.com'**, **u"新浪"**),  
 (**'http://www.163.com'**, **u"网易"**)  
 ]  
 self.render(**'10\_lesson\_base.html'**, 跳转至子模板  
 name=self.current\_user,  
 time=time,bb=bb)

10\_lesson.html 此时父模板为空没内容

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>{% block Title %} {% end %}</**title**>  
</**head**>  
<**body**>  
 {% block body %} {% end %}  
</**body**>  
</**html**>

10\_lesson\_base.html 子模板继承父模，并在需要的地方修改，像PPT的母版一样。

*<!--继承的父摸板-->*{% extends "10\_lesson.html" %}  
*<!--继承title块-->*{% block Title %} 继承 {% end %}  
*<!--继承body块-->*{% block body %}  
 欢迎{{name}}登录<**br**>  
 时间戳：{{time.time()}}<**br**>  
 {% for a in bb %}  
 <**a href="{{a[0]}}"**>{{a[1]}} {{a[0]}}</**a**><**br**>  
 {% end %}<**br**>  
{% end %}



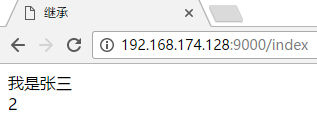
**三、 摸板中使用类或函数的方式**

1、在render方法中把函数或类作为参数传入

**class** JiSuan(object):  
 **def** sub(self, a, b):  
 **return** a-b  
  
**class** Index1Handler(BaseHandler):  
 @tornado.web.authenticated  
 **def** aa(self):  
 **return u"我是张三"**  self.render(**'10\_lesson\_base.html'**,aa=self.aa,jisuan=JiSua)

10\_lesson\_base.html

*<!--继承的父摸板-->*{% extends "10\_lesson.html" %}  
*<!--继承title块-->*{% block Title %} 继承 {% end %}  
*<!--继承body块-->*{% block body %}  
 {{aa()}}<**br**>  
 {{ jisuan().sub(9,7) }}  
{% end %}



2、在html文件中通过import 导入方法或类

导入模块 {% import module %}

导入指定模块 {% from x import y %}

**class** Index2Handler(BaseHandler):  
 @tornado.web.authenticated  
 **def** aa(self):  
 **return u"我是张三"  
 def** get(self):  
 self.render(**'10\_lesson\_base.html'**,namelist=[**'111'**,**'222'**,**'333'**])

新建ten\_lesson\_base.py

*#coding=utf-8***def** add(a, b):  
 c = a+b  
 **return** c  
  
**class** JiSuan(object):  
 @classmethod  
 **def** add(cls, a, b):  
 **return** a+b  
 **def** sub(self, a, b):  
 **return** a-b  
 **def** data(self):  
 **return '<a href="http://www.baidu.com">百度</a>'  
 def** args(self,\*args):  
 **return** [a+**"---aaa" for** a **in** args[0]]

导入模块10\_lesson\_base.html

*<!--继承的父摸板-->*{% extends "10\_lesson.html" %}  
*<!--继承title块-->*{% block Title %} 继承 {% end %}  
*<!--继承body块-->*{% block body %}  
*<!--在html文件中通过import 导入方法或类-->* {% import time %}  
 {{ time.time() }}<**br**>  
 {{'-----------------------------------------'}}<**br**>  
*<!--导入自定义模块-->* {% from ten\_lesson\_base import add,JiSuan %}  
 {{ add(1,5) }}<**br**/>  
 {{ JiSuan().add(7, 1) }}<**br**/>  
 {{ JiSuan().sub(7, 1) }}<**br**/>  
 {{ JiSuan().data() }}<**br**/>  
 {{ JiSuan().args(namelist)}}<**br**/>  
 {% for a in JiSuan().args(namelist) %}  
 {{a}}<**br**/>  
 {% end %}  
{% end %}

