

Django中JSON的使用

原创 忘尘~ 2018-05-24 10:14:13 9158 收藏 10 版权

分类专栏: # Django 文章标签: JSON Django javascript python ajax

一、后端向前端传JSON数据

后端python的json.dumps方法 + 前端js的JSON.parse方法。因为数据只能以字符串的形式进行传送，因此需要将对象转换成字符串，然后再来传数据。

因此后端python的json.dumps方法，就是将python中的**对象**编码成**json字符串**。而前端js的JSON.parse方法就是将这个**json字符串**再转成**对象**。这边如果对对象和json字符串有疑惑，请看最下面的**补充**。

首先后端views.py中：

```
def login(request):
    Users = models.User.objects.all()
    username = []
    for i in Users:
        username.append(i.username)
    return render(request, "login.html", {'username': json.dumps(username)})
```

其中 username是我User表中的一个字段，

好了，通过render我们将json格式的username字符串传给login.html页面

前端html中，js配合django的django的用法：

```
var usernames = "{ { username } }"; //django的django用法，获取username字符串
var usernames_r = usernames.replace(/\"/g, '\\'); //然后反转义一下，将\"转为\"
var usernames_r_p = JSON.parse(usernames_r); //最后用js的JSON.parse方法，将字符串转为Array数组对象（即python中的
```

最终前端得到的usernames_r_p是个数组array(原谅我取的变量名)，然后前端js就能使用数组中的数据啦！我们可以alert一下，看看usernames_r_p输出的是啥。

json可以转化的类型很多，这边只举了个list的例子，其他类型的还有(自己尝试)：

python 原始类型向 json 类型的转化对照表：

Python	JSON
dict	object
list, tuple	array
str, unicode	string
int, long, float	number
True	true
False	false
None	null

二、前端向后端传数据

js的JSON.stringify()方法+python的json.loads()方法，JSON.stringify() 用于将 JavaScript 值转换为 JSON 字符串。具体原理和上面类似，js将对象转成json字符串，后端python将json字符串转回对象，这边只放个object的例子。

前端html中：

```
<script>
    function sub()
    {
        var a = document.getElementById("aa").value;
        var b = document.getElementById("bb").value;
        var str = {"a": a, "b": b};
        var str_json = JSON.stringify(str);
        document.getElementById("cc").value = str_json;
        return true;
    }
</script>

<input type="text" id="aa"/>
<input type="text" id="bb"/>
<form method="POST" onsubmit="return sub()">
    {% csrf_token %}
    <input type="hidden" name="data" id="cc"/>
    <input type="submit" value="提交"/>
</form>
```

后端views.py中：

```
def login(request):
    if request.method == 'POST':
        print(request.POST['data'])
        result = json.loads(request.POST['data'])
        print(result['a'])
```

三、拓展：用ajax (jquery实现) 传json数据

1、前端js

```
<script type="text/javascript">
var token_csrf = "{ csrf_token }"; #因为django的form表单中都包含csrf_token，因此在用ajax传表单数据时也要带上
var email = $("#Email").val();
var psw = $("#Pwd").val();
var login_data = { "email":email , "password":psw };

$.ajax({
    type: "POST",
    contentType: "application/json; charset=utf-8",
    headers: { "X-CSRFToken": token_csrf },
    url: "/login/",
    dataType: "json",
    cache: false,
    data: JSON.stringify(login_data),
    error: function () {
        ShowErrorMessage("登录失败！");
    },
},
```

```

        success: function (data) {
            if (null != data && "" != data) {
                console.log(data);
            }
        }
    });
</script>

```

2、后端python

```

def post(self, request):
    data = json.loads(request.body)

    email = data["email"]
    pass_word = data["password"]
    print(email, pass_word)

    data = {
        'name': 'zhangsan',
        'age': 18,
    }
    return HttpResponse(json.dumps(data), content_type="application/json")
# 或者直接使用 JsonResponse 函数
return JsonResponse(data)

```

四、补充

小案例解释json对象与json字符串

```

>>> import json
>>> username = ["小红", "小明"]
>>> json_username = json.dumps(username) # 将列表对象转化成json字符串
>>> print(type(json_username))
<class 'str'>
>>> print(json_username)
["\u5c0f\u7ea2", "\u5c0f\u660e"]
>>> json.loads(json_username) # 将json字符串转化成列表对象
['小红', '小明']

# 其实json就是一种字符串和其他类型对象转换的一种方式，方便数据的传输，并保持数据的类型。
# 懂了以后我们可以自己来制作一个json字符串

>>> lalala = ["\u6211\u662f", "\u5927\u5e05"] # 注意这边格式，字符串外边用单引号，里面用双引号，否则会报错，无法识别
>>> json.loads(lalala)
['我是', '大师']

# 换个字典类型的对象

>>> usage = {"小明": 12, "小红": 11}
>>> json_usage = json.dumps(usage)
>>> print(json_usage)
{"\u5c0f\u660e": 12, "\u5c0f\u7ea2": 11}

# 同样我们可以自己构造一个json类型的字典字符串
>>> lalala = '{"\u5c0f\u660e": 12, "\u5c0f\u7ea2": 11}'

```

```
>>> mydict = json.loads(lalala)
{'小明': 12, '小红': 11}
>>> mydict["小明"]
12
>>> print(mydict)
```

我喜欢程序员，他们单纯、固执、容易体会到成就感；面对压力，能够挑灯夜战不眠不休；面对困难，能够迎难而上挑战自我。他们也会感到困惑与彷徨，但每个程序员的心中都有一个比尔盖茨或是乔布斯的梦想“用智慧开创属于自己的事业”。我想说的是，其实我是一个程序员