

# 基于请求方法的类视图：

1. 基于方法的类视图，是根据请求的method来执行不同的方法的。如果用户是发送的get请求，那么将会执行这个类的get方法。如果用户发送的是post请求，那么将会执行这个类的post方法。其他的method类似，比如delete、put。

2. 这种方式，可以让代码更加简洁。所有和get请求相关的代码都放在get方法中，所有和post请求相关的代码都放在post方法中。就不需要跟之前的函数一样，通过request.method == 'GET'。

#####

```
class LoginView(views.MethodView):
```

```
    def __render(self, error=None):
```

```
        return render_template('Container/login.html', error=error)
```

```
    def get(self,):
```

```
        return self.__render()
```

```
    def post(self):
```

```
        email = request.form.get('email')
```

```
        password = request.form.get('password')
```

```
        user = User.query.filter(User.email == email).first()
```

```
        print(user)
```

```
        if user:
```

```
            session['userId'] = user.id
```

```
            # 记住登录状态31天
```

```
            session.permanent = True
```

```
            return redirect(url_for('index'))
```

```
        else:
```

```
            return self.__render(error=u'用户名或密码错误')
```

```
app.add_url_rule('/login/', view_func=LoginView.as_view('login'))
```

#####