

[登录](#)

2019年05月04日 阅读 1151

[关注](#)

Flask框架从入门到精通之模板表单(二十)

知识点: 1、WTF使用

一、概况

当前端使用form表单进行参数传递时候, 前端一般都会用js来校验用户输入的参数是否合法。作为后端, 不能依赖前端的校验。要在前端校验的基础上在进行一遍校验, 防止程序出现异常。当参数过多时, 我们要对每个参数都进行校验, 显得非常麻烦。在Flask中我们可以用Flask-WTF帮助我们快速校验。Flask-WTF是集成WTForms, 并带有 csrf 令牌的安全表单和全局的 csrf 保护的功能。

二、WTForms支持的HTML标准字段

字段	说明
StringField	文本字段
TextAreaField	多行文本字段
PasswordField	密码文本字段
HiddenField	隐藏文本字段
DateField	文本字段, 值为datetime.date格式
DateTimeField	文本字段, 值为datetime.datetime格式
IntegerField	文本字段, 值为整数
DecimalField	文本字段, 值为decimal.Decimal
FloatField	文本字段, 值为浮点数
BooleanField	复选框, 值为True和False
RadioField	一组单选框
SelectField	下拉列表
SelectMultipleField	下拉列表, 可选择多个值

名称	说明
FileField	文本上传字段
SubmitField	表单提交按钮
FormField	把表单作为字段嵌入另一个表单
FieldList	一组指定类型的字段

三、WTForms常用验证函数

函数	说明
DataRequired	确保字段中有数据
EqualTo	比较两个字段的值, 常用于比较两次密码输入
Length	验证输入的字符串长度
NumberRange	验证输入的值在数字范围内
URL	验证URL
AnyOf	验证输入值在可选列表中
NoneOf	验证输入值不在可选列表中

使用Flask-WTF需要配置参数SECRET_KEY。

CSRF_ENABLED是为了CSRF (跨站请求伪造) 保护。SECRET_KEY用来生成加密令牌, 当CSRF激活的时候, 该设置会根据设置的密匙生成加密令牌。

在HTML页面中直接写form表单：

#模板文件

```
<form method='post'>
  <input type="text" name="username" placeholder='Username'>
  <input type="password" name="password" placeholder='password'>
  <input type="submit">
</form>
```

复制代码

视图函数中获取表单数据：

```
from flask import Flask,render_template,request

@app.route('/login',methods=['GET','POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        print username,password
```

复制代码

```
return render_template('login.html',method=request.method)
```

使用Flask-WTF实现表单。

配置参数：

复制代码

```
app.config['SECRET_KEY'] = 'python is good'
```

视图函数：

复制代码

```
from flask import Flask, render_template, redirect, url_for, session, request, flash
# 导入wtfl扩展的表单类
from flask_wtf import FlaskForm
# 导入自定义表单需要的字段
from wtforms import SubmitField, StringField, PasswordField
# 导入wtfl扩展提供的表单验证器
from wtforms.validators import DataRequired, EqualTo

app = Flask(__name__)
app.config['SECRET_KEY'] = 'python is good'

# 自定义表单类, 文本字段、密码字段、提交按钮
class Login(FlaskForm):
    name = StringField(label='用户:', validators=[DataRequired('用户名不能为空')])
    pwd = PasswordField(label='密码', validators=[DataRequired('密码不能为空'), EqualTo('pwd1', '密码不一样')])
    pwd1 = PasswordField(label='确认密码', validators=[DataRequired('密码不能为空')])
    submit = SubmitField('提交')

@app.route('/login')
def login():
    return render_template('login.html')

# 定义根路由视图函数, 生成表单对象, 获取表单数据, 进行表单数据验证
@app.route('/', methods=['GET', 'POST'])
def index():
    form = Login()
    if form.validate_on_submit():
        name = form.name.data
        pwd = form.pwd.data
        pwd1 = form.pwd1.data
        print(name, pwd, pwd1)
        return redirect(url_for('login'))
    return render_template('index.html', form=form)

if __name__ == '__main__':
    app.run(debug=True)
```

模板页面：

复制代码

```
<!DOCTYPE html>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form method="post">
  <!--设置csrf_token-->
  {{ form.csrf_token() }}
  {{ form.name.label }}
  <p>{{ form.name }}</p>

  {% for msg in form.name.errors %}
    <p>{{ msg }}</p>
  {% endfor %}
  {{ form.pwd.label }}
  <p>{{ form.pwd }}</p>

  {% for msg in form.pwd.errors %}
    <p>{{ msg }}</p>
  {% endfor %}
  {{ form.pwd1.label }}
  <p>{{ form.pwd1 }}</p>

  {% for msg in form.pwd1.errors %}
    <p>{{ msg }}</p>
  {% endfor %}
  <p>{{ form.submit() }}</p>
  {% for x in get_flashed_messages() %}
    {{ x }}
  {% endfor %}
</form>

</body>
</html>

```

我们在浏览器调试：

- 不填任何参数 -

The screenshot shows a web browser at the address 127.0.0.1:5000. The browser's address bar and toolbar are visible at the top. Below the toolbar, the text "用户：" (User:) is displayed. Underneath, there is a text input field for the username. A red-bordered box highlights this field, and a red error message bubble with an exclamation mark icon appears below it, containing the text "请填写此字段。" (Please fill in this field). Below the username field is another text input field for the password, labeled "确认密码" (Confirm Password). At the bottom of the form is a button labeled "提交" (Submit).

<https://jikedaochang.blog.csdn.net>

- 密码不同

The screenshot shows the same web browser at 127.0.0.1:5000. The browser's address bar and toolbar are visible at the top. Below the toolbar, the text "用户：" (User:) is displayed. Underneath, there is a text input field for the username. Below the username field is another text input field for the password, labeled "确认密码" (Confirm Password). At the bottom of the form is a button labeled "提交" (Submit).

用户:

密码

密码不一样

确认密码

提交

<https://jikedaochang.blog.csdn.net>

- 全部正确

← → ↻ ⓘ 127.0.0.1:5000/login

应用 Google GitHub 极客导航 空指针 颜色转换 网站统计 JSON 序列化 Md2All mysql android

login suc

<https://jikedaochang.blog.csdn.net>

可以看到,利用Flask-WTF可以帮助我们快速校验参数。欢迎关注我的公众号:



文章分类

后端

文章标签

Flask

相关推荐

