

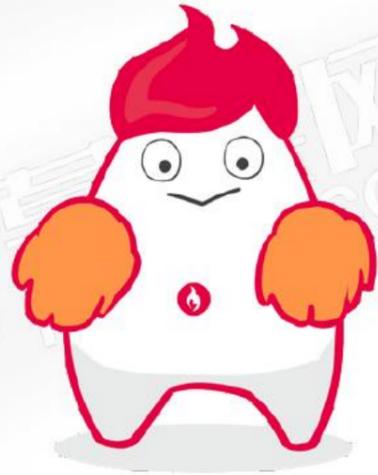
Flask表单

MTV模型



课程介绍

- ◆ 什么是表单
- ◆ 为什么要学习表单
- ◆ 怎样才能学好表单



什么是表单



注册表单



搜索表单

什么是表单

◆ 思考：实战项目中有哪些表单

用户登录/注册

按名称搜索问题

用户发布问题

后台管理（问题、评论等）

用户回答问题

为什么要学习表单

◆ 重要

◆ 离不开它

◆ 所以要学



怎样学好表单

- ◆ 掌握HTML各表单元素的使用
- ◆ 掌握前后端验证表单的方法
- ◆ 掌握从后端获取表单传递的数据
- ◆ 掌握在Flask中使用表单
- ◆ 实战演练夯实基础



章节概要

- ◆ 表单知识点回顾
- ◆ wtf表单介绍
- ◆ 表单常用字段类型及渲染
- ◆ 通过表单保存数据
- ◆ 表单验证
- ◆ 图片上传

表单知识点回顾

常见的表单元素

- ◆ 表单标签——<form>
- ◆ 单行文本框/多行文本框
- ◆ 选择（单选、多选、下拉选择）
- ◆ 隐藏表单域
- ◆ 表单按钮
- ◆ 文件上传框

<form> 标签

- ◆ action——表单提交的URL地址
- ◆ method——表单请求的方式 (GET/POST)
- ◆ enctype——请求内容的形式

application/x-www-form-urlencoded

multipart/form-data



文本标签

- ◆ textarea——多行文本
- ◆ 单行文本(type的不同值)

text——单行文本

url——URL

password——密码

number——数字

email——邮箱

color——颜色

日期时间等(date, month, week, time, datetime, datetime-local)

选择标签

◆ 单选

```
<input type="radio">
```

◆ 多选

```
<input type="checkbox">
```

◆ 下拉选择

```
<select> <option> </option> </select>
```



其他

◆ 文件上传

```
<input type="file">
```

◆ 隐藏表单域

```
<input type="hidden">
```

◆ 按钮

```
<input type="button"> <button> </button>
```



在视图中获取表单值

◆ GET请求

```
request.args.get( 'name' , None)
```

◆ POST请求

```
request.form.get( 'name' , None)
```

◆ 思考：HTML表单在Flask中如何快速使用？

wtf表单介绍

wtf表单介绍



flask-wtf表单介绍

- ◆ 集成 wtforms
- ◆ CSRF保护
- ◆ 与 Flask-Uploads 一起支持文件上传

wtf表单介绍

◆ 安装

PIP安装：pip install Flask-WTF

源码安装：python setup.py install

◆ 配置 (CSRF保护)

```
WTF_CSRF_SECRET_KEY = 'a random string'
```

第一个表单模型

◆ 示例代码

```
1 from flask_wtf import FlaskForm
2 from wtforms import StringField
3
4
5 class LoginForm(FlaskForm):
6     """ 用户名 """
7     username = StringField(Label='用户名')
```

表单常用字段类型及渲染

表单字段的常用核心参数

- ◆ label——label标签（如：输入框前的文字描述）
- ◆ default——表单的默认值
- ◆ validators——表单验证规则
- ◆ widget——定制界面显示方式（如：文本框、选择框）
- ◆ description——帮助文字

表单渲染

◆ 使用模板语法渲染表单内容

表单输入区域：{{ form.username }}

表单label：{{ form.username.label.text }}

表单常用字段类型

◆ 文本/字符串

- StringField——字符串输入
- PasswordField——密码输入
- TextAreaField——长文本输入
- HiddenField——隐藏表单域

表单常用字段类型

◆ 数值（整数，小数）

- FloatField——浮点数输入
- IntegerField——整数输入
- DecimalField——小数输入（更精确）

表单常用字段类型

◆ 选择

- RadioField——radio单选
- SelectField——下拉单选
- SelectMultipleField——下拉多选
- BooleanField——勾选（复选框）

表单常用字段类型

◆ 日期/时间

- DateField——日期选择
- DateTimeField——日期时间选择

表单常用字段类型

◆ 文件/文件上传

- FileField——文件单选
- MultipleFileField——文件多选

表单常用字段类型

◆ 其它

- SubmitField——提交按钮
- FieldList——自定义的表单选择列表（如：选择用户对象）
- FormField——自定义多个字段构成的选项

通过表单保存数据

通过表单保存数据

- ◆ 第一步：检测表单是否已经通过验证

```
form.validate_on_submit()
```

- ◆ 第二步：获取表单中传递过来的值

```
form.field_name.data
```

- ◆ 第三步：业务逻辑代码编写（可结合ORM）

CSRF表单保护

- ◆ 默认开启CSRF保护

- ◆ 关闭单个表单CSRF保护

```
form = LoginForm(csrf_enabled=False)
```

- ◆ 全局关闭（**不推荐**）

```
WTF_CSRF_ENABLED = False
```

同步请求CSRF保护

- ◆ 模板中添加csrf_token

```
{{ form.csrf_token }}
```

```
<input type="hidden" name="csrf_token"
```

```
value="{{ csrf_token() }}" />
```

异步请求 (AJAX) CSRF保护

- ◆ 第一步：模板中添加csrf_token

```
<meta name="csrf-token" content="{{  
csrf_token() }}">
```

```
<script type="text/javascript">
```

```
var csrftoken = "{{ csrf_token() }}"
```

```
</script>
```

异步请求 (AJAX) CSRF保护

◆ 第二步：添加 X-CSRFToken 头

```
1 var csrftoken = $('meta[name=csrf-token]').attr('content')
2
3 $.ajaxSetup({
4   beforeSend: function(xhr, settings) {
5     if (!/^(GET|HEAD|OPTIONS|TRACE)$/i.test(settings.type)
6       && !this.crossDomain) {
7       xhr.setRequestHeader("X-CSRFToken", csrftoken)
8     }
9   }
10 })
11
```

表单验证

为什么要验证表单

◆ 思考：以手机注册为例，不验证表单会怎样？

- 用户输入可能不是手机号
- 用户输入的可能是他的手机号
- 不断的提交表单

表单验证

◆ 思考：表单验证为了什么？

- 更好的用户体验
- 更少安全隐患
- 永远不要相信用户的输入

表单验证

- ◆ 步骤1：导入内置的表单验证器（或自定义）

```
from wtforms.validators import DataRequired
```

- ◆ 步骤2：配置到表单字段

```
username = StringField('用户名',  
validators=[InputRequired(), my_validator])
```

内置表单验证器

- ◆ DataRequired/ InputRequired——必填
- ◆ Email/ URL/UUID——电子邮箱/URL/UUID
- ◆ Length(min=-1, max=-1, message=None)
长度范围验证
- ◆ EqualTo(fieldname, message=None)
两个字段输入的值相等（如：密码确认）

自定义表单验证

◆ 场景一：只有本表单使用

◆ 场景二：多个表单中使用

如：验证手机号码

自定义表单验证

◆ 示例代码

```
1 from flask_wtf import FlaskForm
2 from wtforms import StringField
3
4
5 class LoginForm(FlaskForm):
6     """ 用户名 """
7     username = StringField(Label='用户名')
8
9     def validate_username(form, field):
10         if len(field.data) > 50:
11             raise ValidationError('不能超过50个字符')
```

图片上传

思考

◆ 现实生活中，哪些场景下有文件上传？

- 设置个人头像
- 发朋友圈
- 图片搜索
-

图片上传

- ◆ 两种方式实现文件上传

- 方式一：不使用wtf实现
- 方式二：使用FileField并添加类型验证

- ◆ 文件名称格式化

- `werkzeug.utils.secure_filename`

图片上传：不使用wtf

- ◆ 第一步：设置`<form>`的`enctype`

`enctype="multipart/form-data"`

- ◆ 第二步：在视图函数中获取文件对象

`request.files`

- ◆ 第三步：保存文件

`f.save(file_path)`

图片上传：验证

- ◆ FileRequired
文件必须上传验证
- ◆ FileAllowed
文件类型验证

使用扩展：Flask-Uploads

- ◆ 常用文件类型验证
- ◆ 指定文件上传的目录

课程总结

◆ 知识点总结

“踩坑”指南

◆ 在实际开发中会有哪些坑呢？