浙江理工大学
ZHEJIANG SCI-TECH UNIVERSITY

Lab3

《Web应用开发》 / 任课教师：罗志一

计算机科学与技术学院

在视图函数里操作数据库

- 在视图函数里操作数据库的方式和我们在Python Shell中的练习大致相同，只不过需要一些额外的工作。比如把查询结果作为参数传入模板渲染出来，或是获取表单的字段值作为提交到数据库的数据。
- 本节我们将把上一节课学习的所有数据库操作知识运用到一个简单的笔记程序中。这个程序可以让你创建、编辑和删除笔记，并在主页列出所有保存后的笔记。

添加笔记

Create

- 为了支持输入笔记内容，我们先创建一个用于填写新笔记的表单：

```
from flask_wtf import FlaskForm
from wtforms import TextAreaField, SubmitField
from wtforms.validators import DataRequired

class NewNoteForm(FlaskForm):
    body = TextAreaField('Body', validators=[DataRequired()])
    submit = SubmitField('Save')
```

Create

- 我们创建一个new_note视图，这个视图负责渲染创建笔记的模板，并处理表单的提交。

```
@app.route('/new', methods=['GET', 'POST'])
def new_note():
    form = NewNoteForm()
    if form.validate_on_submit():
        body = form.body.data
        note = Note(body=body)
        db.session.add(note)
        db.session.commit()
        flash('Your note is saved.')
        return redirect(url_for('index'))
    return render_template('new_note.html', form=form)
```

Create

- 在templates/macros.html中定义form_field宏渲染表单字段

```
{% macro form_field(field) %}
    {{ field.label }}<br>
    {% if field.flags.required %}
        {{ field(required='required', **kwargs) }}<br>
    {% else %}
        {{ field(**kwargs) }}<br>
    {% endif %}
    {% if field.errors %}
        {% for error in field.errors %}
            <small class="error">{{ error }}</small><br>
        {% endfor %}
    {% endif %}
{% endmacro %}
```

Create

- 在templates/new_note.html模板中渲染表单（传入rows和cols来定制<textarea>输入框的大小）

```
{% from 'macros.html' import form_field %}

{% block content %}
<h2>New Note</h2>
<form method="post">
    {{ form.csrf_token }}
    {{ form_field(form.body, rows=5, cols=50) }}
    {{ form.submit }}<br>
</form>
{% endblock %}
```

Create

- index视图用来显式主页，渲染主页对应的模板：

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

- 在对应的index.html模板中，我们添加一个指向创建新笔记页面的链接：

```
<h1>Notebook</h1>
<a href="{{ url_for('new_note') }}">New Note</a>
```

展示笔记

Read

- 我们实现了添加笔记的功能，当你在创建笔记的页面单击保存后，程序会重定向到主页，提示的消息告诉你刚刚提交的笔记已经成功保存了，可是你却无法看到创建后的笔记。为了在主页列出所有保存的笔记，我们需要修改index视图。

```
@app.route('/')
def index():
    notes = Note.query.all()
    return render_template('index.html', notes=notes)
```

- 下一步，我们将在模板中将笔记们显式出来。

Read

- 修改index.html模板

```
<h1>Notebook</h1>
<a href="{{ url_for('new_note') }}">New Note</a>

<h4>{{ notes|length }} notes:</h4>
{% for note in notes %}
    <div class="note">
        <p>{{ note.body }}</p>
    </div>
{% endfor %}
```

更新笔记

Update

- 更新一条笔记和创建一条笔记的实现代码几乎完全相同，首先是编辑笔记的表单：

```
class EditNoteForm(FlaskForm):
    body = TextAreaField('Body', validators=[DataRequired()])
    submit = SubmitField('Update')
```

Update

- 用于渲染更新笔记页面和处理更新表单提交的edit_note视图：

```
@app.route('/edit/<int:note_id>', methods=['GET', 'POST'])
def edit_note(note_id):
    form = EditNoteForm()
    note = Note.query.get(note_id)
    if form.validate_on_submit():
        note.body = form.body.data
        db.session.commit()
        flash('Your note is updated.')
        return redirect(url_for('index’))
    form.body.data = note.body
    return render_template('edit_note.html', form=form)
```

Update

- 在笔记列表中的每个笔记内容下添加一个编辑按钮，用来访问编辑页面：

```
{% for note in notes %}
    <div class="note">
        <p>{{ note.body }}</p>
        <a class='btn' href="{{ url_for('edit_note',note_id=note.id) }}">Edit</a>
    </div>
{% endfor %}
```

课后练习：删除笔记