web表单使用及第三方登录页面实现

web表单使用及第三方登录页面实现

1. config配置文件创建及配置
2. config.py配置文件加载
3. 创建用户登录表单
4. 编写表单模板
5. 处理OpenIDs
    5.1. 申请测试用openid
    5.2. 配置OpenID提供者列表
    5.3. 修改登录视图函数
    5.4. 注册登录视图函数
    5.5. 登录模板中渲染OpenID提供者
6. 启动服务查看页面
7. 代码提交版本库

1. config配置文件创建及配置

为了实现web表单功能，就要使用到flask的扩展库`FLASK-WTF`。

```
pip install flask-wtf
```

各种flask扩展库都需要配置一些参数，所以创建一个配置文件`app/config.py`用于存放这些配置信息。

对于扩展库`FLASK-WTF`主要需要以下两个配置信息：

- CSRF_ENABLED = True：激活`跨站点请求伪造`保护；
- SECRET_KEY = 'you-will-never-guess'：CSRF被激活时，需要该参数用于令牌加密，并实现表单数据的验证。

```
# ----------FLASK-WTF扩展库配置--------- #
# 激活跨站点请求伪造保护
CSRF_ENABLED = True
# CSRF被激活时，用于令牌加密，表单验证
SECRET_KEY = 'you-will-never-guess'
```

注意：SECRET_KEY参数尽量使用较为复杂的密钥，可以使用`os.urandom(16)`的方式获取随机密钥。

2. config.py配置文件加载

将`app/config.py`配置文件中的内容加载到flask应用程序中，修改`app/__init__.py`文件。

```
import os
from flask import Flask
```

```
def create_app():
    """应用工厂函数"""
    app = Flask(__name__)
    # 加载config配置文件
    app.config.from_pyfile('config.py', silent=True)

    # 注册hello视图URL
    from app.hello import HelloWorld
    app.add_url_rule('/hello', view_func=HelloWorld.as_view('hello'))

    try:
        # 确保 app.instance_path 存在
        os.makedirs(app.instance_path)
    except OSError:
        pass

    return app
```

3. 创建用户登录表单

创建并编写用户登录表单文件`app/forms.py`：

```
from flask_wtf import FlaskForm
from wtforms import StringField, BooleanField
from wtforms.validators import DataRequired


class LoginForm(FlaskForm):
    # DataRequired: 数据不可为空的验证器
    openid = StringField('openid', validators=[DataRequired()])
    remember_me = BooleanField('remember_me', default=False)
```

4. 编写表单模板

新建并编写表单模板文件`app/templates/login/login.html`。

```
{% extends 'base.html' %}

{% block content %}
<h1>登录</h1>
<form action="" method="post" name="login">
    {{ form.hidden_tag() }}
    <p>
        请录入你的OpenID: <br>
        {{ form.openid(size=80) }}<br>
        {% for error in form.openid.errors %}
            <span style="color: red;">[{{ error }}]</span>
        {% endfor %}
```

```
    </p>
    <p>{{ form.remember_me }} 记住我</p>
    <p><input type="submit" value="登录"></p>
</form>
{% endblock %}
```

5. 处理OpenIDs

5.1. 申请测试用openid

通过网站http://www.openid.org.cn 注册自己的测试用OpenID。

5.2. 配置OpenID提供者列表

在配置文件`app/config.py`中配置OpenID提供者列表（由于国内需要翻墙访问其他网址，所以主要是通过新注册的OpenID进行测试）。

```
# OpenID提供者列表
OPENID_PROVIDERS = [
    {'name': 'OpenID', 'url': 'https://www.openid.com'},
    {'name': 'Google', 'url': 'https://www.google.com/accounts/o8/id'},
    {'name': 'Yahoo', 'url': 'https://me.yahoo.com'},
    {'name': 'AOL', 'url': 'http://openid.aol.com/<username>'},
    {'name': 'Flickr', 'url': 'http://www.flickr.com/<username>'}
]
```

5.3. 修改登录视图函数

编写登录视图函数`app/login.py`

```
from flask.views import View
from flask import flash, redirect, render_template, current_app

from app.forms import LoginForm


class LoginView(View):
    methods = ['GET', 'POST']

    def dispatch_request(self):
        form = LoginForm()
        if form.validate_on_submit():
            flash("登录请求，登录OpenID：{0}，是否记住我：{1}".format(form.openid.data,
form.remember_me.data))
            return redirect('/index')
        return render_template('login/login.html',
                               title='登录',
                               form=form,
                               providers=current_app.config['OPENID_PROVIDERS'])
```

5.4. 注册登录视图函数

编辑`app/__init__.py`，注册登录视图

```
import os
from flask import Flask


def create_app():
    """应用工厂函数"""
    app = Flask(__name__)
    # 加载config配置文件
    app.config.from_pyfile('config.py', silent=True)

    # 注册hello视图URL
    from app.hello import HelloWorld
    app.add_url_rule('/hello', view_func=HelloWorld.as_view('hello'))
```

```
    # 注册Login登录视图URL
    from app.login import LoginView
    app.add_url_rule('/login', view_func=LoginView.as_view('login'))

    try:
        # 确保 app.instance_path 存在
        os.makedirs(app.instance_path)
    except OSError:
        pass
```

5.5. 登录模板中渲染OpenID提供者

修改`app/templates/login/login.html`代码。其中，JS代码用于判断加载的OpenID内容中是否存在`<username>`，若存在，则弹窗让登录则录入用户名，然后自动将登录者录入的用户名替代掉`<username>`字符串，来作为最终的OpenID。

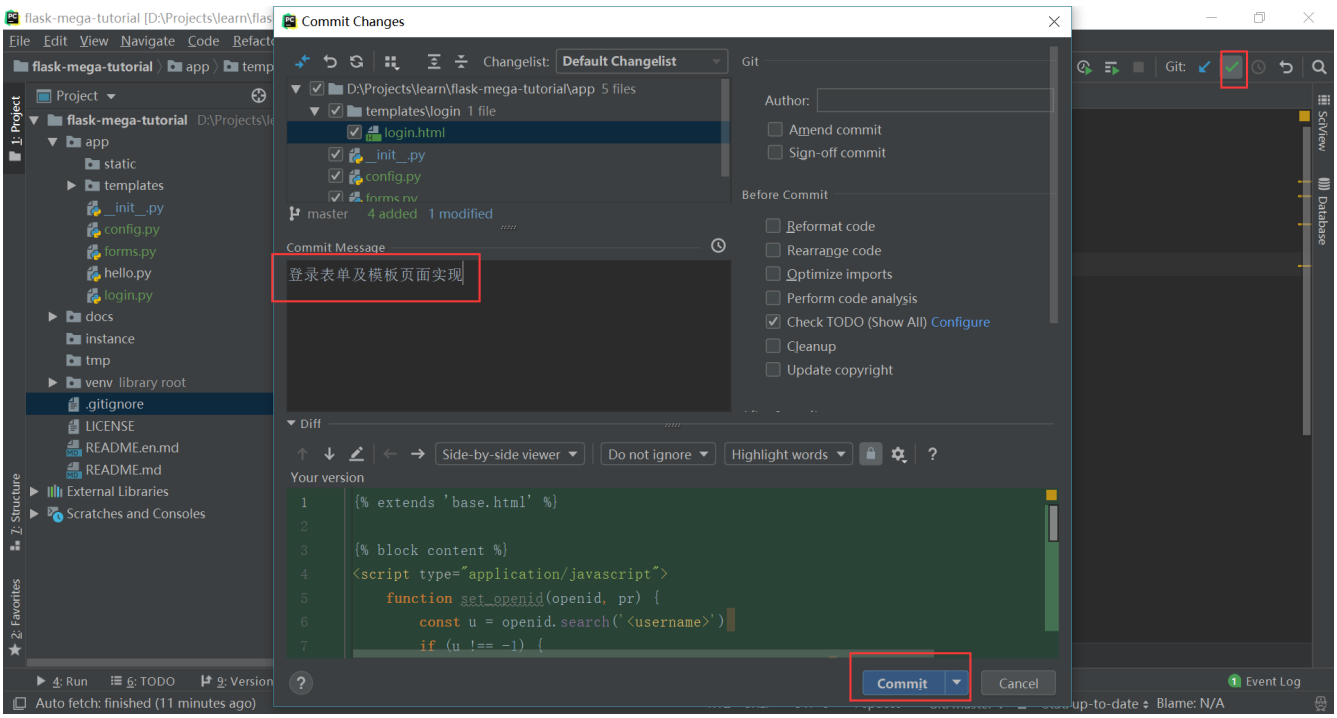
```
{% extends 'base.html' %}

{% block content %}
<script type="application/javascript">
    function set_openid(openid, pr) {
        const u = openid.search('<username>')
        if (u !== -1) {
            const user = prompt(`请输入您的${pr}用户名：`)
            openid = openid.substr(0, u) + user
        }
        const form = document.forms['login']
        form.elements['openid'].value = openid
    }
</script>
<h1>登录</h1>
<form action="" method="post" name="login">
    {{ form.hidden_tag() }}
    <p>
        请录入你的OpenID，或者选择下面一个提供商：<br>
        {{ form.openid(size=80) }}<br>
        {% for error in form.openid.errors %}
            <span style="color: red;">[{{ error }}]</span>
        {% endfor %}<br>
        |{% for pr in providers %}
            <a href="javascript:set_openid('{{ pr.url }}', '{{ pr.name }}');">{{ pr.name }}
</a>
        {% endfor %}
    </p>
    <p>{{ form.remember_me }} 记住我</p>
    <p><input type="submit" value="登录"></p>
</form>
{% endblock %}
```

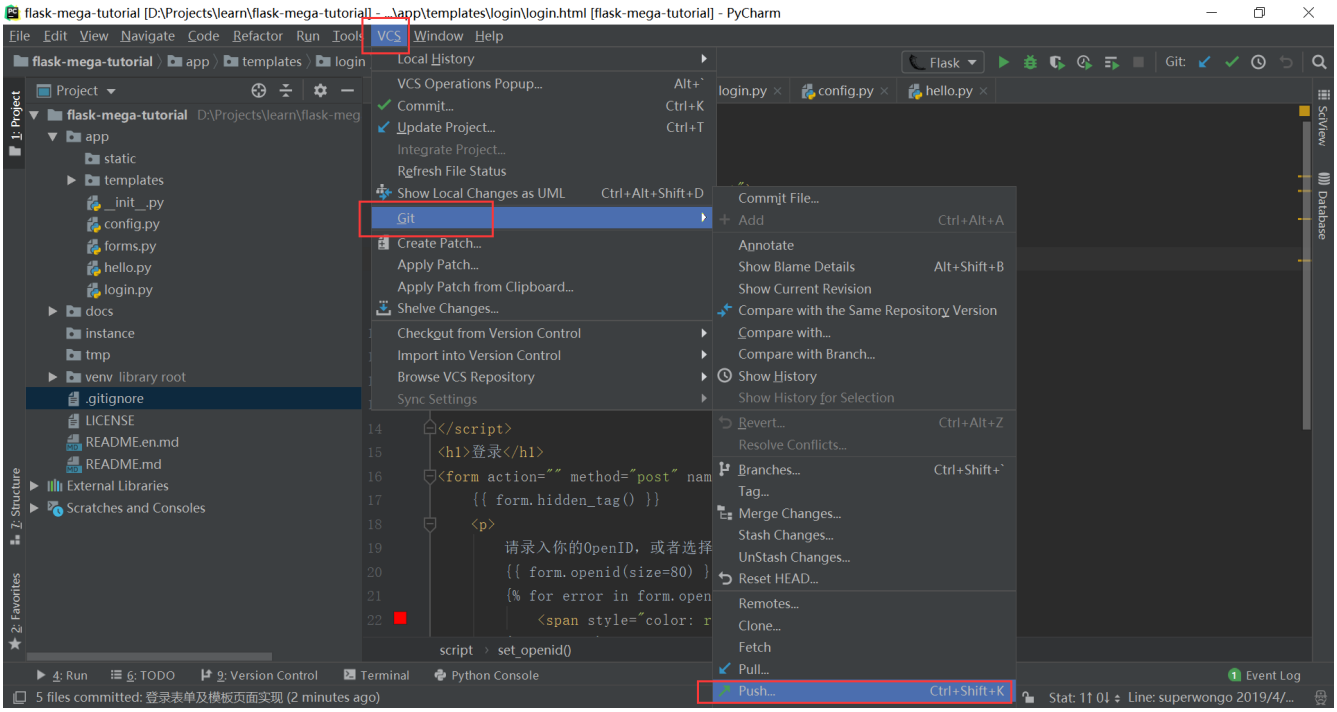
6. 启动服务查看页面

启动服务，查看`/login`页面，以下即为完成的页面部分，但是还没用编写具体的登录逻辑，后续会进行添加。

7. 代码提交版本库

通过pycharm进行版本提交




flask-mega-tutorial [D:\Projects\learn\flask-mega-tutorial] - ...\app\templates\login\login.html [flask-mega-tutorial] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Local History
VCS Operations Popup... Alt+`
Commit... Ctrl+K
Update Project... Ctrl+T
Integrate Project...
Refresh File Status
Show Local Changes as UML Ctrl+Alt+Shift+D
Git
Create Patch...
Apply Patch...
Apply Patch from Clipboard...
Shelve Changes...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
Commit File...
Add Ctrl+Alt+A
Annotate
Show Blame Details Alt+Shift+B
Show Current Revision
Compare with the Same Repository Version
Compare with...
Compare with Branch...
Show History
Show History for Selection
Revert... Ctrl+Alt+Z
Resolve Conflicts...
Branches... Ctrl+Shift+`
Tag...
Merge Changes...
Stash Changes...
UnStash Changes...
Reset HEAD...
Remotes...
Clone...
Fetch
Pull...
Push... Ctrl+Shift+K
Project
flask-mega-tutorial D:\Projects\learn\flask-meg
app
static
templates
__init__.py
config.py
forms.py
hello.py
login.py
docs
instance
tmp
venv library root
.gitignore
LICENSE
README.en.md
README.md
External Libraries
Scratches and Consoles
login.py
config.py
hello.py
script set_openid()
4: Run 6: TODO 9: Version Control Terminal Python Console
Event Log
5 files committed: 登录表单及模板页面实现 (2 minutes ago)
Stat: 1↑ 0↓ Line: superwongo 2019/4/...