

1.渲染密码修改页面

web/views/user/User.py

```
# -*- encoding: utf-8 -*-
"""
File      : User.py
teaching   :
    用户模块 -- 登录
"""

from flask import Blueprint, render_template, views, request, jsonify, make_response, url_for, redirect, g
from common.lib.UrlManager import UrlManager
from web.models.User import User # 模型类导出
from common.lib.UserService import UserService
from application import app, db
import json

# 构建蓝图对象
route_user = Blueprint('user_page', __name__)

class ReSetPwd(views.MethodView):
    def get(self):
        # 1.建立渲染数据字典
        context = {}
        # 2.数据为空, 添加用户对象数据
        if not context:
            context['current_user'] = g.current_user
        return render_template('user/reset_pwd.html', **context)

    # 使用add_url_rule方法来构造与视图关联的请求url
    # 继承自模板的as_view方法可以将类转换为可以为路由注册的视图函数
    # as_view('login') --- 传入的是站点名 (标准操作是视图名小写 Login -- login)
    #           <Rule '/user/login' (GET, HEAD, OPTIONS) -> user_page.login>] user_page.login中的login就是
    # as_view指明的站点名
    route_user.add_url_rule('/reset-pwd', view_func=ReSetPwd.as_view('reset-pwd'))
    route_user.add_url_rule('/edit', view_func=EditInfo.as_view('edit'))
    route_user.add_url_rule('/login', view_func=Login.as_view('login'))
    route_user.add_url_rule('/logout', view_func=Logout.as_view('logout'))
```

调整模板

1

1 fipro > web > templates > user > edit.html

Project User.py reset_pwd.html edit.js AuthInterceptor.py edit.html

```
1  {% extends "common/layout_main.html" %}\n2  {% block content %}\n3      <div class="row border-bottom">\n4          <div class="col-lg-12">\n5              <div class="tab_title">\n6                  <ul class="nav nav-pills">\n7                      <li class="current">\n8                          <a href="{{ buildUrl('/user/edit') }}">>信息编辑</a>\n9                      </li>\n10                     <li>\n11                         <a href="{{ buildUrl('/user/reset-pwd') }}">>修改密码</a>\n12                     </li>\n13                 </ul>\n14             </div>\n15         </div>\n16     </div class="row m-t user_edit_wrap">\n17         <div class="col-lg-12">\n18             <h2 class="text-center">账号信息编辑</h2>\n19             <div class="form-horizontal m-t m-b">\n20                 <div class="form-group">\n21                     <label class="col-lg-2 control-label">手机:</label>\n22                     <div class="col-lg-10">\n23                         <input type="text" name="mobile" class="form-control" placeholder="请输入手机号码" value="13800138000">\n24                     </div>\n25                 </div>\n26             </div>\n27         </div>
```

Structure Bookmarks

2

1 fipro > web > templates > user > reset_pwd.html

Project User.py reset_pwd.html edit.js AuthInterceptor.py edit.html

```
1  <div class="col-lg-12">\n2      <div class="tab_title">\n3          <ul class="nav nav-pills">\n4              <li>\n5                  <a href="{{ buildUrl('/user/edit') }}">>信息编辑</a>\n6              </li>\n7              <li class="current">\n8                  <a href="{{ buildUrl('/user/reset-pwd') }}">>修改密码</a>\n9              </li>\n10             </ul>\n11         </div>\n12     </div>\n13     </div class="row m-t user_reset_pwd_wrap">\n14         <div class="col-lg-12">\n15             <h2 class="text-center">修改密码</h2>\n16             <div class="form-horizontal m-t m-b">\n17                 <div class="form-group">\n18                     <label class="col-lg-2 control-label">账号:</label>\n19                     <div class="col-lg-10">\n20                         <label class="control-label">编程浪子2</label>\n21                     </div>\n22                 </div>\n23             </div>\n24         </div>\n25     </div>
```

Structure Bookmarks

3

```
21 <div class="form-group">
22     <label class="col-lg-2 control-label">账号:</label>
23     <div class="col-lg-10">
24         <label class="control-label">{{current_user.login_name}}</label>
25     </div>
26 </div>
27 <div class="hr-line-dashed"></div>
28 <div class="form-group">
29     <label class="col-lg-2 control-label">手机:</label>
30     <div class="col-lg-10">
31         <label class="control-label">{{current_user.mobile}}</label>
32     </div>
33 </div>
34 <div class="hr-line-dashed"></div>
35
36 <div class="form-group">
37     <label class="col-lg-2 control-label">原密码:</label>
38     <div class="col-lg-10">
39         <input type="password" id="old_password" class="form-control" value="">
40     </div>
41 </div>
42 <div class="hr-line-dashed"></div>
43
44 <div class="form-group">
```

```
<div class="col-lg-10">
    <input type="password" id="new_password" class="form-control" value="">
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
    <div class="col-lg-4 col-lg-offset-2">
        <button class="btn btn-w-m btn-outline btn-primary" id="save">保存</button>
    </div>
</div>
</div>
</div>
{% endblock %}

{% block js %}
<script src="{{ buildStaticUrl('/js/user/reset_pwd.js') }}></script>
{% endblock %}
```

```
{% extends "common/layout_main.html" %}
{% block content %}
<div class="row border-bottom">
```

```

<div class="col-lg-12">
    <div class="tab_title">
        <ul class="nav nav-pills">
            <li>
                <a href="{{ buildUrl('/user/edit') }}>信息编辑</a>
            </li>
            <li class="current">
                <a href="{{ buildUrl('/user/reset-pwd') }}>修改密码</a>
            </li>
        </ul>
    </div>
</div>
<div class="row m-t user_reset_pwd_wrap">
    <div class="col-lg-12">
        <h2 class="text-center">修改密码</h2>
        <div class="form-horizontal m-t m-b">
            <div class="form-group">
                <label class="col-lg-2 control-label">账号:</label>
                <div class="col-lg-10">
                    <label class="control-label">{{current_user.login_name}}</label>
                </div>
            </div>
            <div class="hr-line-dashed"></div>
            <div class="form-group">
                <label class="col-lg-2 control-label">手机:</label>
                <div class="col-lg-10">
                    <label class="control-label">{{current_user.mobile}}</label>
                </div>
            </div>
            <div class="hr-line-dashed"></div>

            <div class="form-group">
                <label class="col-lg-2 control-label">原密码:</label>
                <div class="col-lg-10">
                    <input type="password" id="old_password" class="form-control" value="">
                </div>
            </div>
            <div class="hr-line-dashed"></div>

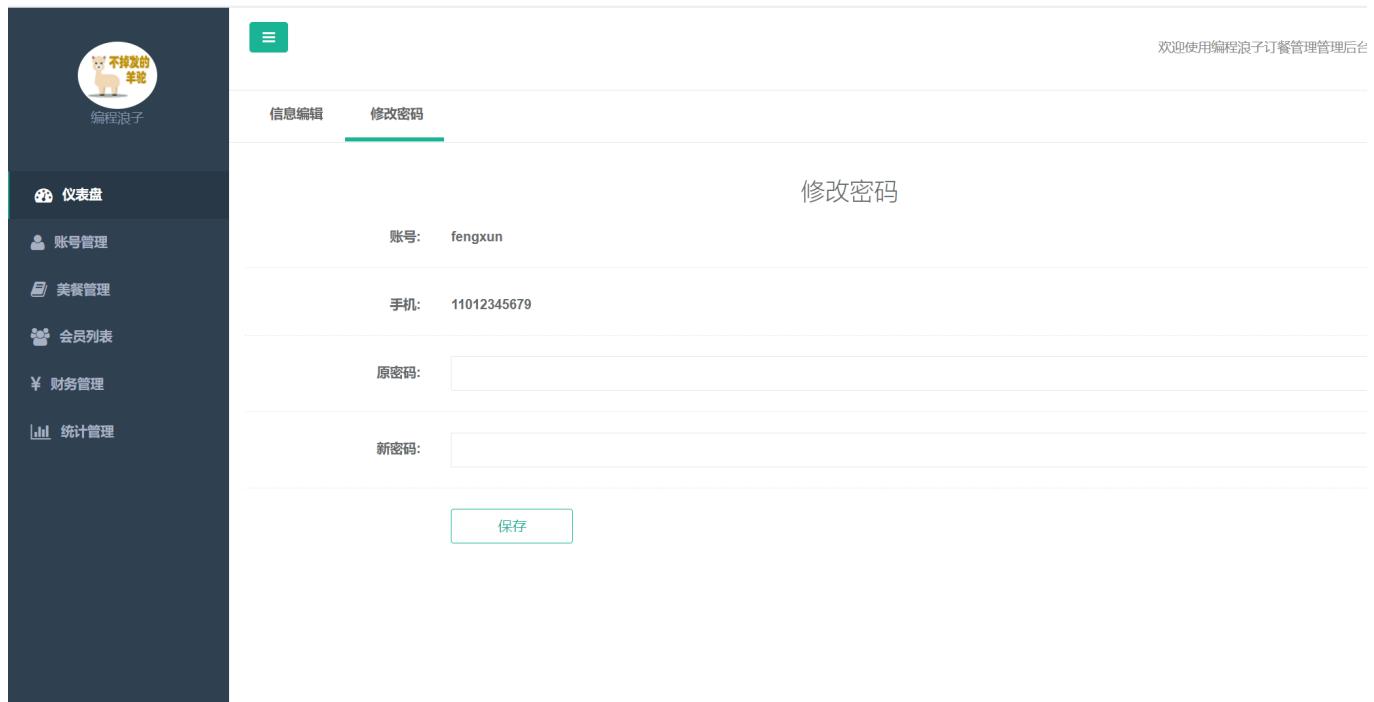
            <div class="form-group">
                <label class="col-lg-2 control-label">新密码:</label>
                <div class="col-lg-10">
                    <input type="password" id="new_password" class="form-control" value="">
                </div>
            </div>
            <div class="hr-line-dashed"></div>
            <div class="form-group">
                <div class="col-lg-4 col-lg-offset-2">
                    <button class="btn btn-w-m btn-outline btn-primary" id="save">保存</button>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}

{%block js %}
<script src="{{ buildStaticUrl('/js/user/reset_pwd.js') }}></script>

```

```
{% endblock %}
```

启动访问测试 <http://127.0.0.1:8888/user/reset-pwd>



2. 实现密码修改

点击保存后，触发js，通过验证 发送ajax请求

The screenshot shows a code editor with several files open. On the left is a file tree for a project named 'fipro' with sub-directories like 'common', 'config', 'docs', 'job', 'web', 'interceptors', 'models', 'statics', and 'js'. In the center, the 'reset_pwd.js' file is being edited. A red arrow points from the sidebar's 'js' folder to the 'reset_pwd.js' file in the editor. The code in 'reset_pwd.js' is as follows:

```
old_password: old_password,
new_password: new_password
};

$.ajax( url: {
    url:common_ops.buildUrl( path: "/user/reset-pwd" ),
    type:'POST',
    data:data,
    dataType:'json',
    success:function( res ){
        btn_target.removeClass( value: "disabled" );
        var callback = null;
        if( res.code == 200 ){
            callback = function(){
                window.location.href = window.location.href;
            }
        }
        common_ops.alert( res.msg,callback );
    }
});
```

视图处理post请求

```
# -*- encoding: utf-8 -*-
"""
File      : User.py
teaching   :
    用户模块 -- 登录
"""

from flask import Blueprint, render_template, request, jsonify, make_response, url_for, redirect, g
from common.lib.UrlManager import UrlManager
from web.models.User import User # 模型类导出
from common.lib.UserService import UserService
from application import app, db
import json

# 构建蓝图对象
route_user = Blueprint('user_page', __name__)

class ReSetPwd(views.MethodView):
    def get(self):
        # 1.建立渲染数据字典
        context = {}
        # 2.数据为空, 添加用户对象数据
        if not context:
            context['current_user'] = g.current_user
        return render_template('user/reset_pwd.html', **context)

    def post(self):
        """
        修改密码
        :return:
        """
        # 1.构建响应数据
        resp = {"code": 200, "msg": "修改密码成功", "data": {}}

        # 2.获取post请求提交的数据
        req = request.values

        # 3.如果存在响应数据键 则获取对应数据
        old_password = req["old_password"] if "old_password" in req else "" # 旧密码
        new_password = req["new_password"] if "new_password" in req else "" # 新密码
        confirm_password = req["confirm_password"] if "confirm_password" in req else "" # 确认密码

        if old_password == "" or new_password == "" or confirm_password == "":
            resp["code"] = 400
            resp["msg"] = "参数不完整"
            return jsonify(resp)

        if new_password != confirm_password:
            resp["code"] = 400
            resp["msg"] = "新密码和确认密码不一致"
            return jsonify(resp)

        user = User.query.filter_by(id=g.current_user.id).first()
        if user:
            user.old_password = old_password
            user.new_password = new_password
            user.confirm_password = confirm_password
            db.session.commit()
            resp["code"] = 200
            resp["msg"] = "修改密码成功"
            return jsonify(resp)
        else:
            resp["code"] = 400
            resp["msg"] = "用户不存在"
            return jsonify(resp)
```

```
# -*- encoding: utf-8 -*-
"""
File      : User.py
teaching   :
    用户模块 -- 登录
"""

from flask import Blueprint, render_template, request, jsonify, make_response, url_for, redirect, g
from common.lib.UrlManager import UrlManager
from web.models.User import User # 模型类导出
from common.lib.UserService import UserService
from application import app, db
import json

# 构建蓝图对象
route_user = Blueprint('user_page', __name__)
```

```
class ReSetPwd(views.MethodView):
    def get(self):
        # 1.建立渲染数据字典
        context = {}
        # 2.数据为空, 添加用户对象数据
        if not context:
            context['current_user'] = g.current_user
        return render_template('user/reset_pwd.html', **context)

    def post(self):
        """
        修改密码
        :return:
        """
        # 1.构建响应数据
        resp = {"code": 200, "msg": "修改密码成功", "data": {}}

        # 2.获取post请求提交的数据
        req = request.values

        # 3.如果存在响应数据键 则获取对应数据
        old_password = req["old_password"] if "old_password" in req else "" # 旧密码
        new_password = req["new_password"] if "new_password" in req else "" # 新密码
        confirm_password = req["confirm_password"] if "confirm_password" in req else "" # 确认密码

        if old_password == "" or new_password == "" or confirm_password == "":
            resp["code"] = 400
            resp["msg"] = "参数不完整"
            return jsonify(resp)

        if new_password != confirm_password:
            resp["code"] = 400
            resp["msg"] = "新密码和确认密码不一致"
            return jsonify(resp)

        user = User.query.filter_by(id=g.current_user.id).first()
        if user:
            user.old_password = old_password
            user.new_password = new_password
            user.confirm_password = confirm_password
            db.session.commit()
            resp["code"] = 200
            resp["msg"] = "修改密码成功"
            return jsonify(resp)
        else:
            resp["code"] = 400
            resp["msg"] = "用户不存在"
            return jsonify(resp)
```

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

```
修改密码
:return:
"""

# 1.构建响应数据
resp = {"code": 200, "msg": "修改密码成功", "data": {}}

# 2.获取post请求提交的数据
req = request.values

# 3. 如果存在响应数据键 则获取对应数据
old_password = req["old_password"] if "old_password" in req else "" # 旧密码
new_password = req["new_password"] if "new_password" in req else "" # 新密码

# 4.密码是否存在数据, 及 符合规则
if (not old_password) or len(old_password) < 1:
    resp["code"] = -1
    resp["msg"] = "请输入正确的原密码"

    return jsonify(resp)
if (not new_password) or len(new_password) < 6:
    resp["code"] = -1
    resp["msg"] = "请输入不少于6位的新密码"
    return jsonify(resp)

# 5.判断新密码与旧密码是否一致
if old_password == new_password:
    resp["code"] = -1
    resp["msg"] = "请重新输入一个密码, 新密码不要和旧密码一样哦~"
    return jsonify(resp)

# 6.获取用户对象数据
user_info = g.current_user

# 7.获取用户密码加密盐
salt = user_info.login_salt

# 8.将传入的旧密码加密
gen_pwd = UserService.genPwd(old_password, salt) # 跟注册时相同的加密机制

# 9.判断传入的旧密码 是否与 用户对象原密码是否一致 -- 不一致就不能修改
if user_info.login_pwd != gen_pwd:
    resp["code"] = -1
    resp["msg"] = "原密码不对~"
    return jsonify(resp)

# 10.更改密码
user_info.login_pwd = UserService.genPwd(new_password, salt)
# 提交到数据库
db.session.add(user_info)
db.session.commit()

# 11.构建json响应对象
response = make_response(json.dumps(resp))

# 12.携带新的cookie, 更新用户cookie数据
# 加密cookie数据并构建 完整cookie
cokie_str = UserService.genAuthCode(user_info) + "#" + str(user_info.uid)
# 设置 UTH_COOKIE_NAME cookie, 同时设置过期时间 --- 注意在登录视图 post处理里面 cookie设置也去
设置一个过期时间 max_age=7 * 24 * 60 * 60
```

```
response.set_cookie(app.config.get("AUTH_COOKIE_NAME"), cokie_str,
max_age=7 * 24 * 60 * 60)

# 13.响应
return response

# 使用add_url_rule方法来构造与视图关联的请求url
# 继承自模板的as_view方法可以将类转换为可以为路由注册的视图函数
# as_view('login') --- 传入的是站点名 (标准操作是视图名小写 Login -- login)
#     <Rule '/user/login' (GET, HEAD, OPTIONS) -> user_page.login>] user_page.login中的login就是
as_view指明的站点名
route_user.add_url_rule('/reset-pwd', view_func=ReSetPwd.as_view('reset-pwd'))
route_user.add_url_rule('/edit', view_func=EditInfo.as_view('edit'))
route_user.add_url_rule('/login', view_func=Login.as_view('login'))
route_user.add_url_rule('/logout', view_func=Logout.as_view('logout'))
```

访问 <http://127.0.0.1:8888/user/reset-pwd> 测试 修改密码 (1234567) 保存

