

# Python 로그인 설명

Python 로그인 설명

## 메인 프로그램에서 "/" 유입

세션 여부를 판단해서 로그인 페이지나 메인 페이지로 이동

```
@app.route('/', methods=['GET', 'POST'])
def mainpage():
    userid = session.get('userid', None)
    form = LoginForm() #로그인폼
    if userid is None:
        return render_template('login.html', form=form)
    else:
        msgList = postgresql.getMsg()
        return render_template("index.html", msgList=msgList, userid=userid)
```

## 로그인 페이지

```
<html>
  <head>
    <meta charset='utf-8' />
    <meta name='viewport' content="width=device-width, initial-scale=1, shrink-to-fit=no" />

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
      integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/
X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">
    </script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
      integrity="sha384-
U02eT0CpHqdSjQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous">
    </script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
      integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous">
    </script>
  </head>
```

```

<body>

  <div class="container">
    <div class="row mt-5">
      <h1>로그인</h1>
    </div>
    <div class="row mt-5">
      <div class="col-12">
        <form method="POST" action="/login">
          {{form.csrf_token}}
          <div class="form-group">

            {{form.userid.label("아이디")}}
            {{form.userid(class="form-control", placeholder="아이디")}}
          </div>

          <div class="form-group">

            {{form.password.label("비밀번호")}}
            {%if form.password.errors%}
            <!-- 로그인 실패시 errors/ password필드의 모든 에러를 갖고있다. -->
            {{form.password.errors.0}}
            <!-- 0 : 첫번째 에러메시지를 출력 하겠다 라는 뜻 -->
            {%endif%}
            {{form.password(class="form-control", placeholder="비밀번호")}}
            <p>
              {{message}}
            </p>
          </div>

          <button type="submit" class="btn btn-primary">로그인</button>
        </form>
      </div>
    </div>
  </div>
</body>
</html>

```

## 메인 페이지 로그인 처리

```

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm() #로그인폼
    try:
        if form.validate_on_submit(): #유효성 검사
            logger.debug('{}가 로그인 했습니다'.format(form.data.get('userid')))
            session['userid']=form.data.get('userid') #form에서 가져온 userid를 세션에 저장
            msgList = postgresql.getMsg()

```

```

        return render_template("index.html", msgList=msgList,
userid=form.data.get('userid')) #성공하면 main.html로
    except ValueError as ve:
        logger.debug('로그인 실패' + str(ve))
        return render_template('login.html', form=form, message=str(ve))

```

## Form

```

from flask_wtf import FlaskForm
from wtforms import StringField
from wtforms import PasswordField
from wtforms.validators import DataRequired, EqualTo
from models import User #models.py 가져옴

class RegisterForm(FlaskForm):
    userid = StringField('userid', validators=[DataRequired()])
    username = StringField('username', validators=[DataRequired()])
    email = StringField('email', validators=[DataRequired()])
    password = PasswordField('password', validators=[DataRequired(),
EqualTo('re_password')]) #비밀번호 확인
    re_password = PasswordField('re_password', validators=[DataRequired()])

class LoginForm(FlaskForm):
    class UserPassword(object):
        def __init__(self, message=None):
            self.message = message
        def __call__(self, form, field):
            userid = form['userid'].data
            password = field.data
            usertable = User.query.filter_by(userid=userid).first()
            if usertable is None or usertable.password != password:
                raise ValueError('로그인 실패 비밀번호 틀림')

    userid = StringField('userid', validators=[DataRequired('ID를 입력하여 주세요!')])
    password = PasswordField('password', validators=[DataRequired('패스워드를 입력해 주세요'), UserPassword()])

```

## 객체

```

from flask_sqlalchemy import SQLAlchemy

db = SQLAlchemy() #SQLAlchemy를 사용해 데이터베이스 저장

class User(db.Model): #데이터 모델을 나타내는 객체 선언
    __tablename__ = 'user_table' #테이블 이름

    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(32), unique=True, nullable=False)
    email = db.Column(db.String(32), unique=True, nullable=False)

```

```

userid = db.Column(db.String(32), unique=True, nullable=False)
password = db.Column(db.String(8), nullable=False)

# def __init__(self, email, password):
#     self.email = email
#     self.set_password(password)
#

def set_password(self, password):
    self.password = generate_password_hash(password)

def check_password(self, password):
    return check_password_hash(self.password, password)

```

## 사용자 등록 페이지

```

<html>
  <head>
    <meta charset='utf-8' />
    <meta name='viewport' content="width=device-width, initial-scale=1, shrink-to-
fit=no" />

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css"
      integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/
X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">
    </script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/
popper.min.js"
      integrity="sha384-
U02eT0CpHqdSfJQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous">
    </script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js"
      integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous">
    </script>
  </head>

  <body>
    <div class="container">
      <div class="row mt-5">
        <h1>회원가입</h1>
      </div>
      <div class="row mt-5">
        <div class="col-12">

```

```

<form method="POST" > <!--액션 삭제해야됨-->
    {{form.csrf_token}}
    <div class="form-group">
        <!-- <label for="userid">아이디</label>
        <input type="text" class="form-control" id="userid"
placeholder="아이디" name="userid" /> -->
        {{form.userid.label("아이디")}}
        {{form.userid(class="form-control", placeholder="아이디")}}
    </div>
    <div class="form-group">
        <!-- <label for="username">이름</label>
        <input type="text" class="form-control" id="username"
placeholder="사용자이름" name="username" /> -->
        {{form.username.label("사용자 이름")}}
        {{form.username(class="form-control", placeholder="사용자 이
름")}}
    </div>
    <div class="form-group">
        <!-- <label for="username">이메일</label>
        <input type="text" class="form-control" id="email"
placeholder="사용자메일" name="email" /> -->
        {{form.email.label("사용자 이메일")}}
        {{form.email(class="form-control", placeholder="사용자 이메일")}}
    </div>
    <div class="form-group">
        <!-- <label for="password">비밀번호</label>
        <input type="password" class="form-control" id="password"
placeholder="비밀번호" name="password" /> -->
        {{form.password.label("비밀번호")}}
        {{form.password(class="form-control", placeholder="비밀번호")}}
    </div>
    <div class="form-group">
        <!-- <label for="re-password">비밀번호 확인</label>
        <input type="password" class="form-control" id="re_password"
placeholder="비밀번호 확인" name="re-password" /> -->
        {{form.re_password.label("비밀번호 확인")}}
        {{form.re_password(class="form-control", placeholder="비밀번호 확
인")}}
    </div>
    <button type="submit" class="btn btn-primary">제출</button>
</form>
</div>
</div>
</body>
</html>

```

## 메인 페이지 사용자 등록

```
@app.route('/register', methods=['GET', 'POST']) #GET, 포스트 메소드 둘다 사용
def register(): #get 요청 단순히 페이지 표시 post요청 회원가입-등록을 눌렀을때 정보 가져오는것
    form = RegisterForm()

    if form.validate_on_submit(): # POST검사의 유효성검사가 정상적으로 되었는지 확인할 수 있다. 입력 안
한것들이 있는지 확인됨.
        #비밀번호 = 비밀번호 확인 -> EqulaTo

        user = User() #models.py에 있는 user
        user.userid = form.data.get('userid')
        user.username = form.data.get('username')
        user.email = form.data.get('email')
        user.password = form.data.get('password')

        print(user.userid,user.password) #회원가입 요청시 콘솔창에 ID만 출력 (확인용, 딱히 필요없음)
        db.session.add(user) # id, name 변수에 넣은 회원정보 DB에 저장
        db.session.commit() #커밋

        return "가입 완료" #post요청일시는 '/'주소로 이동. (회원가입 완료시 화면이동)

return render_template('register.html', form=form)
```

---

🕒Revision #1

★Created 2023-06-10 12:59:36 UTC by Hyeon Su Ryu

✎Updated 2023-06-10 13:24:51 UTC by Hyeon Su Ryu