

# Lab06:

## 보충설명

ESM2014-41

객체지향프로그래밍 및 실습

SKKU 시스템경영공학과

조영일

# Model & View & Template(Django MVT)

- Model : Django 로 만든 웹 서비스에서 영속적인 데이터 저장을 위해 정의 하여야 하는 Class(정의 이후 `manage.py makemigrations` / `manage.py migrate` 명령을 통해 Database에 반영 => Class로 부터 Instance를 만들고 `.save()` method 호출 시 데이터 베이스에 저장)
- View : HTTP Request에 따른 적절한 HTTP Response를 생성하는 Logic을 구현하는 부분. 만들고자 하는 모든 페이지당 1개 씩 구현되어야 하며, 페이지가 아니더라도 Form을 Submit 받아 처리하는 경우에도 모든 Submit 요청에 대해 구현되어야 함
- Template : View 에서 처리된 결과에 대해 실제 사용자에게 표시될 웹페이지의 형식을 결정하는 부분 View 1개당 보통 1개 이상이 작성되어야 하며, Form을 Submit 받아 처리하는 경우 구현하지 않을 수도 있다.

# 회원 가입 기능 구현 예시

- 2개의 view 구현 필요
  - `views.py > def signup_form` : 회원가입 양식을 보여주는 view
  - `views.py > def signup_complete` : 회원가입 form으로 부터 전송 받은 데이터를 실제로 Model에 저장하고 회원가입 완료 화면을 보여주는 view
- 2개의 template 구현 필요
  - `templates>signup_form.html` : 회원가입 Form을 포함하는 template
  - `templates>signup_complete.html` : 회원가입 완료 화면을 위한 template

# Django User Authentication System

- Django User Authentication System : Django에 미리 탑재되어 있는 회원가입, 로그인, 로그아웃 등 사용자 인증 관련 시스템. Django User Authentication System을 이용하면 대부분의 웹사이트에서 필요로 하는 회원 관리 관련 기능을 손쉽게 구현 할 수 있다.
- <https://docs.djangoproject.com/ko/2.2/topics/auth/>

# 구현 예시 - 회원가입

- 회원 관련 처리를 위한 “member” application을 생성했다고 가정
- member>urls.py

```
urlpatterns = [  
    url('signup/', views.signup, name='signup'),  
]
```

# 구현 예시 - 회원가입

- member>views.py

```
def signup(request):  
    context = {}  
    return render(request, "member/signup.html", context)
```

# 구현 예시 - 회원가입

- templates/member/signup.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1>회원가입</h1>
  {% if error %}
    <p>{{ error }}</p>
  {% endif %}
  <form id="signup-form" action="{% url 'member:signup_complete' %}" method="post">
    {% csrf_token %}
    <label for="id">아이디</label>
    <input type="text" id="id" name="id">
    <label for="password">비밀번호</label>
    <input type="password" id="password" name="password">
    <label for="last_name">성</label>
    <input type="text" id="last_name" name="last_name" class="form-control">
    <label for="first_name">이름</label>
    <input type="text" id="first_name" name="first_name" class="form-control">
    <div class="col-xs-12 text-center">
      <button type="submit" class="btn btn-primary btn-block btn-flat">등록</button>
    </div>
  </form>
</body>
</html>
```

# 구현 예시 - 회원가입

- member>views.py

```
def signup_complete(request):  
    context = {}  
  
    try:  
        user = create_user(username=request.POST.get('id'), password=request.POST.get('password'))  
    except IntegrityError:  
        context.update({"error": "이미 가입된 계정입니다"})  
        return render(request, "member/signup.html", context)  
  
    user.first_name = request.POST.get("first_name")  
    user.last_name = request.POST.get("last_name")  
    user.save()  
  
    context.update({"user": user})  
  
    return render(request, "member/signup_complete.html", context)
```



# 구현 예시 - 회원가입

- templates/member/signup\_complete.html

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1>회원가입완료</h1>
  <p>{{ user.last_name }}{{ user.first_name }}님, 회원가입을 축하합니다.</p>
  <p>가입 ID : {{ user.user_name }}</p>
</body>
</html>
```