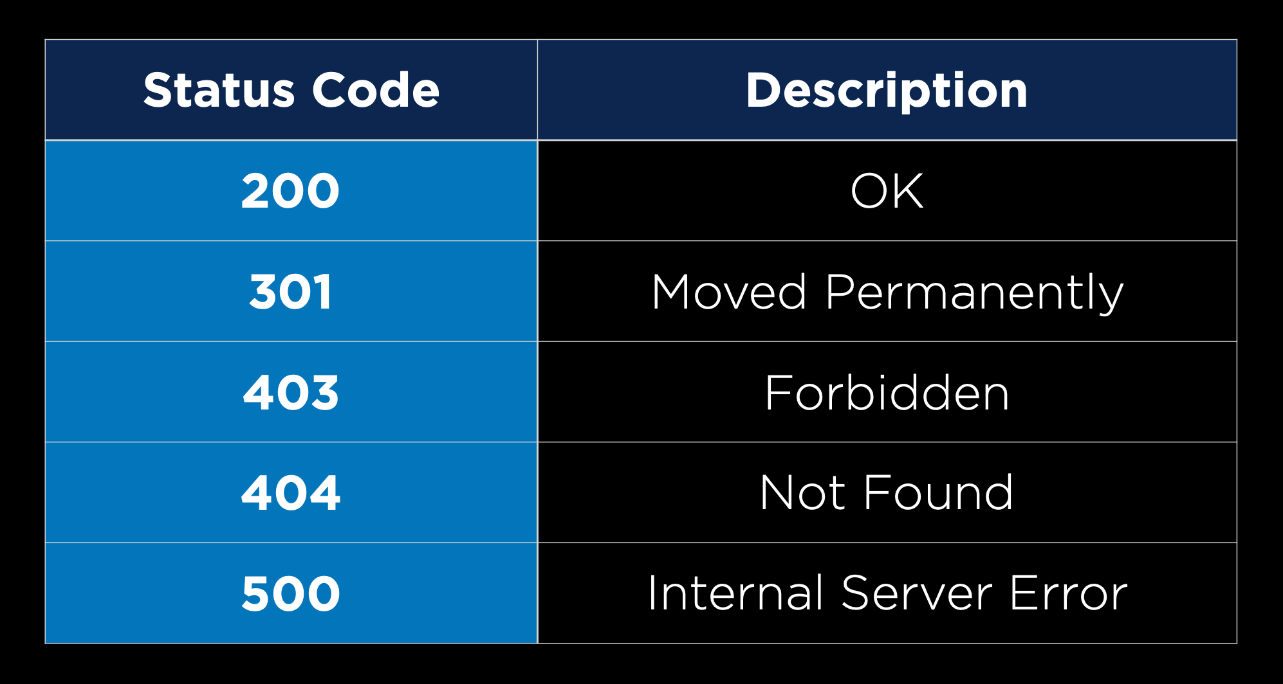
**<lecture3. Django>**

**HTTP Status Codes**



**installation**

- pip 설치 -> Django 설치

**make a project with Django**

- Django-admin startproject <directory name>

- \* project folder 내 manage.py 건들면 안 됨!

**run local server**

- python manage.py runserver

**create app for Django project**

- Django project는 여러 개의 app들이 모여 하나의 app을 구성함(component 형식 like react)

- python manage.py startapp <app name>: 새로운 app 생성

# my project\settings.py\INSTALLED\_APPS에 <app name> 추가

텍스트이(가) 표시된 사진

자동 생성된 설명

# my project\urls.py\urlpatterns

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**How to do something with new app**

#hello/views.py

텍스트이(가) 표시된 사진

자동 생성된 설명

#hello/urls.py

텍스트, 화면, 스크린샷, 은색이(가) 표시된 사진

자동 생성된 설명

**How to apply functions in Django html file**

# newyear/index.html

텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

<!-- you can input any function into {%%}. the end of the functions you shold write {% endif %} that is the endpoint of if functions -->

# newyear/views.py

텍스트이(가) 표시된 사진

자동 생성된 설명

- dictionary 안의 key "newyear"의 value "True"를 위의 html 내 {% keyname %} 형식으로 받음.

**inheritance of Django html fifles**

# templates/tasks/layout.html

텍스트이(가) 표시된 사진

자동 생성된 설명

# templates/tasks/index.html

텍스트이(가) 표시된 사진

자동 생성된 설명

# templates/tasks/index.html

텍스트이(가) 표시된 사진

자동 생성된 설명

{% extends "tasks/layout.html" %}을 입력하여 해당 html file 내 {% block body %} {% endblock%}의 태그들이 layout.html에 inherit됨.

**How to prevent to be duplicated url names**

# tasks/urls.py

텍스트이(가) 표시된 사진

자동 생성된 설명

app\_name = "{app name}"을 추가하여 해당 app의 교유한 url name들로 지정.



a태그 내 url name 앞에 app name을 추가해준다.

**Form security**

Django html 내 form 태그 쓸 때 {% csrf\_token %} 넣어서 보안 처리하지 않으면 CSRF(Cross-site request forgery)에 취약하기 때문에 권한이 없다며 403 error 뜸.

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 같이 {% csrf\_token %} 추가.

**Add task to tasks page and redirect to tasks page whenever add task**

# tasks/views.py

텍스트이(가) 표시된 사진

자동 생성된 설명

허나 위처럼 만들면 다른 사람이 접속해도 똑같은 tasks를 볼 수 있음.

아래처럼 session 기능을 추가하여 각 user의 고유한 tasks로 만들어줘야 함.

텍스트, 스크린샷, 화면, 은색이(가) 표시된 사진

자동 생성된 설명

tasks가 request.session 내에 있는지 여부 확인하여 없으면 empty list 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

위 코드만 입력하면 error뜨는데 migrate해줘야 함.

$ python manage.py migrate

위 커맨드는 default tables를 Django database 안에 생성할 수 있게 해준다.(자세한 건 추후 배움)

**project 1**

* **Specification**
  + Entry Page: Visiting /wiki/TITLE, where TITLE is the title of an encyclopedia entry, should render a page that displays the contents of that encyclopedia entry.
    - The view should get the content of the encyclopedia entry by calling the appropriate util function.
    - If an entry is requested that does not exist, the user should be presented with an error page indicating that their requested page was not found.
    - If the entry does exist, the user should be presented with a page that displays the content of the entry. The title of the page should include the name of the entry.
  + Index Page: Update index.html such that, instead of merely listing the names of all pages in the encyclopedia, user can click on any entry name to be taken directly to that entry page.
  + Search: Allow the user to type a query into the search box in the sidebar to search for an encyclopedia entry.
    - If the query matches the name of an encyclopedia entry, the user should be redirected to that entry’s page.
    - If the query does not match the name of an encyclopedia entry, the user should instead be taken to a search results page that displays a list of all encyclopedia entries that have the query as a substring. For example, if the search query were ytho, then Python should appear in the search results.
    - Clicking on any of the entry names on the search results page should take the user to that entry’s page.
  + New Page: Clicking “Create New Page” in the sidebar should take the user to a page where they can create a new encyclopedia entry.
    - Users should be able to enter a title for the page and, in a textarea, should be able to enter the Markdown content for the page.
    - Users should be able to click a button to save their new page.
    - When the page is saved, if an encyclopedia entry already exists with the provided title, the user should be presented with an error message.
    - Otherwise, the encyclopedia entry should be saved to disk, and the user should be taken to the new entry’s page.
  + Edit Page: On each entry page, the user should be able to click a link to be taken to a page where the user can edit that entry’s Markdown content in a textarea.
    - The textarea should be pre-populated with the existing Markdown content of the page. (i.e., the existing content should be the initial value of the textarea).
    - The user should be able to click a button to save the changes made to the entry.
    - Once the entry is saved, the user should be redirected back to that entry’s page.
  + Random Page: Clicking “Random Page” in the sidebar should take user to a random encyclopedia entry.
  + Markdown to HTML Conversion: On each entry’s page, any Markdown content in the entry file should be converted to HTML before being displayed to the user. You may use the python-markdown2 package to perform this conversion, installable via pip3 install markdown2.
    - Challenge for those more comfortable: If you’re feeling more comfortable, try implementing the Markdown to HTML conversion without using any external libraries, supporting headings, boldface text, unordered lists, links, and paragraphs. You may find using regular expressions in Python helpful.
* **Hints**
  + By default, when substituting a value in a Django template, Django HTML-escapes the value to avoid outputting unintended HTML. If you want to allow for an HTML string to be outputted, you can do so with the safe filter (as by adding |safe after the variable name you’re substituting).