ПРОГРАМУВАННЯ В ІНТЕРНЕТ

МАРШРУТИЗАЦІЯ. ЗАПИТИ І ВІДПОВІДІ СЕРВЕРА

Лекція 2

ОБРОБКА ЗАПИТУ DJANGO

1. Django визначає який кореневий модуль UrlConf використовувати. Зазвичай це настройка ROOT_URLCONF
2. Django завантажує модуль конфігурації URL і шукає зміну urlpatterns. Це повинен бути список екземплярів Django.conf.urls.url() або path+repath 2.0 version
3. Path вказує рядок запиту однозначно, а repath – як RegEx
4. Django перебирає кожен шаблон URL по порядку і зупиняється при першому співпадінні із шаблоном.
5. Якщо один із регулярних виразів співпадає з URL, Django імпортує і викликає відповідне представлення, що є функцією Python
6. Якщо жоден вираз не співпадає або виникла помилка на будь-якому етапі, Django викликає відповідний обробник помилок.

- **Кореневий файл маршрутизації вказується у параметрах файлу settings.py**

```
ROOT_URLCONF = 'mysite.urls'
```

СИНТАКСИС РЕГУЛЯРНИХ ВИРАЗІВ

Перелік символів ([abc]) abc[xyz]t = > abcxt, abcyt, abczt

- **[az-], [-az], [a-z] – 26 малих латинських букв**
- **Метасимволи**
  - **^ - початок рядка,**
  - **$ - кінець рядка,**
  - **\A – початок тексту**
  - **\Z – кінець тексту**
  - **. Будь-який символ в рядку**

Набір символів у дужках називається символьним класом і дозволяє вказати інтерпретатору, що на даному місці рядка може стояти один із перечислених символів

СИНТАКСИС РЕГУЛЯРНИХ ВИРАЗІВ

- Метасимволи – стандартні переліки символів
- \w – буквено-цифровий символ або “_”
- \W – не \w
- \d – цифровий символ
- \D – не \d
- \s – будь-який “пробільний” символ (по замовчуванню - \t\n\r\f)
- \S – не \s

Метасимволи-варіанти:

- Test(qwe|abc) знаходить ‘testqwe’ або ‘testabc’

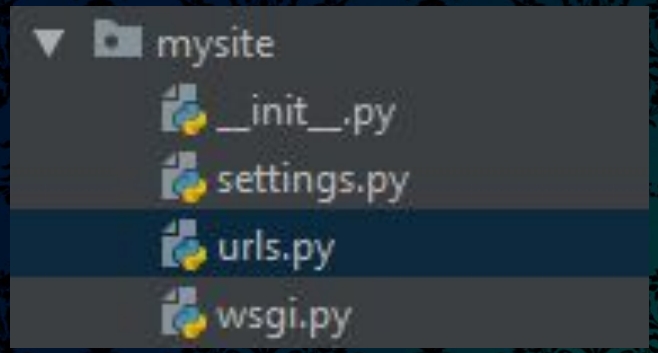
СИНТАКСИС РЕГУЛЯРНИХ ВИРАЗІВ

- Метасимволи – повтори
- * нуль або більше раз
- + один або більше раз
- ? нуль або один раз
- {n} точно n раз
- {n,} не менше n раз
- {n,m} не менше n але не більше m раз

ЖАДІБНІСТЬ РЕГУЛЯРНИХ ВИРАЗІВ

- b+ як і b* поверне з abbbbc-> bbbb
- b+? поверне перше входження b
- b*? –поверне пустий рядок, оскільки нижня межа запиту 0
- b{2,3}? –знайде два символи bb
- b{2,3} без «?» жадібний і знайде 3 входження bbb

МАРШРУТИЗАЦІЯ




```python
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('', include('myapp.urls')),
    path('myapp/', include('mysecondapp.urls')),
```

```python
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
]
```

```python
from django.http import HttpResponse


def index(request):
    return HttpResponse("Hello, world. Myapp")
```

МАРШРУТИЗАЦІЯ З ПАРАМЕТРАМИ

Синтаксис Python іменованих груп: (?P<name>pattern)

Url(r'^item/(?P<year>[\d]{4})/(?P<month>[0-9]{2}/(?P<day>[\d]{2})$', views.day_archive , name = "day_archive")

Запит до /item/2019/02/20/ викликатиме функцію views.day_archive(request, year='2019', month='02', day = '20'), замість views.day_archive(request, '2019', '02' , '20').

Алгоритм відповідності/групування

Якщо існує іменований аргумент, то він буде використаний замість позиційного аргумента. Інакше всі неіменовані параметри будуть передані як іменовані аргументи.

ЗНАЙДЕНІ АРГУМЕНТИ ЗАВЖДИ РЯДКИ

- Кожен знайдений аргумент передається в представлення як рядок, незалежно від того, яке “співпадіння” визначено в регулярному виразі

- Значення по замовчуванню можна встановити присвоївши аргументу конкретне значення. В такому випадку аргумент стає необов'язковим.

```
def page(request , num  = "1"):
    if num == "1":
```

- **В регулярних виразах пробіл вважається символом, неправильне використання пробілів призводитиме до помилок**
- **Хоча регулярні вирази можуть відбирати тільки числові значення та перейдуть вони у функцію завжди у текстовому вигляді**
- **Кожний вираз з urlpatterns буде скомпільований при першому використанні, що робить систему швидкою.**

КОМБІНУВАННЯ URLCONFS

- Додавання додаткових URL-шаблонів з допомогою списку екземплярів
- Вказування спільного префікса один раз і групування різних суфіксів
- Вкладені аргументи
- Передача додаткових аргументів в представленні

- При пошуку аргументів Django шукатиме спочатку зовнішній аргумент.
- При вкладених виразах зовнішній вираз буде братись першим аргументом, а вкладений другим.
- Запис ?: замість ?P дозволить ігнорувати параметр і відповідно у функцію не потрібно включати даний параметр як аргумент.

- **Функції регулярних виразів також можуть приймати третій необов’язковий параметр**
- **Даний параметр буде приймати вигляд колекції(словника)**

```
re_path(r'^optional-args/(?P<year>[0-9]{4})/$', views.optional_args,
{'foo': 'bar'})
```

DJANGO SHORTCUT

Django.shortcuts – методи відповідей

- **HttpResponce – відповідь сервера - рядок**
- **Redirect – перенаправлення на вказану адресу**
- **Render – повертає шаблон з контекстними даними**

```
(request, template_name,  context=None, content_type=None, status=None, using=None)
```