

Маршрутизация. Запросы и ответы сервера

№ урока: 2 **Курс:** Python&Django

Средства обучения: Персональный компьютер с установленным PyCharm , Python и Django.

Обзор, цель и назначение урока

Целью данного урока является познакомить учащегося с системой привязки интернет-адресов в фреймворке. Познакомить с регулярными выражениями. Научить составлять шаблоны url и привязывать их к функциям-обработчикам. Продемонстрировать написание регулярного выражения. Познакомить с понятием HTTP, рассмотреть запросы и ответы сервера.

Изучив материал данного занятия, учащийся сможет:

- Создавать шаблоны интернет-адресов и привязывать к ним отображения(view).
- Понимать, что такое HTTP, GET и POST запросы.
- Писать ответы сервера.

Содержание урока

1. Что такое регулярное выражение?
2. Как привязывать интернет-адрес к функции отображения.
3. HTTP и методы протокола.
4. Рассмотрение GET и POST запросов.
5. Различие GET и POST запросов на стороне сервера.

Резюме

- **URI** (*Uniform Resource Identifier*, унифицированный идентификатор ресурса) — путь до конкретного ресурса, над которым необходимо осуществить операцию (например, в случае использования метода GET подразумевается получение ресурса).
- **Регулярные выражения** — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец (*pattern*), состоящая из символов и метасимволов.

- Метасимволы - стандартные перечни символов:

<code>\w</code>	буквенно-цифровой символ или "_"
<code>\W</code>	не <code>\w</code>
<code>\d</code>	цифровой символ
<code>\D</code>	не <code>\d</code>
<code>\s</code>	любой "пробельный" символ (по умолчанию - [<code>\t\n\r\f</code>])
<code>\S</code>	не <code>\s</code>

- Метасимволы - разделители строк

<code>^</code>	начало строки
<code>\$</code>	конец строки
<code>\A</code>	начало текста
<code>\Z</code>	конец текста
<code>.</code>	любой символ в строке

- Метасимволы-повторения

<code>*</code>	ноль или более раз ("жадный"), то же что <code>{0,}</code>
<code>+</code>	один или более раз ("жадный"), то же что <code>{1,}</code>
<code>?</code>	ноль или один раз ("жадный"), то же что <code>{0,1}</code>
<code>{n}</code>	точно n раз ("жадный")
<code>{n,}</code>	не менее n раз ("жадный")
<code>{n,m}</code>	не менее n но не более m раз ("жадный")
<code>*?</code>	ноль или более раз ("не жадный"), то же что <code>{0,}? </code>
<code>+</code>	один или более раз ("не жадный"), то же что <code>{1,}? </code>
<code>??</code>	ноль или один раз ("не жадный"), то же что <code>{0,1}? </code>
<code>{n}? </code>	точно n раз ("не жадный")
<code>{n,}? </code>	не менее n раз ("не жадный")
<code>{n,m}? </code>	не менее n но не более m раз ("не жадный")

- **HTTP** (протокол передачи гипертекста) — протокол прикладного уровня передачи данных. Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.
- **Метод** представляет собой последовательность из любых символов, кроме управляющих и разделителей, и определяет операцию, которую нужно осуществить с указанным ресурсом.

Закрепление материала

- Что такое регулярное выражение?
- Какие виды метасимволов вы знаете?
- Зачем нужны регулярные выражения в Django?
- Каким образом происходит подключение функции отображения к интернет-адресу?
- Что такое HTTP?
- Как работает клиент-серверная архитектура приложений?
- Какие основные методы HTTP вы знаете?

Дополнительное задание

Задание

Расширить проект с первого урока , создав в нем приложение `second_app`.

Самостоятельная деятельность учащегося

Задание №1

Выучите основные термины, рассмотренные на уроке.

Задание №2

Подключите `second_app` в проект, создайте несколько url-адресов, привяжите к ним свои функции обработки, которые будут использовать внутри себя разные методы ответов: `render`, `render_to_response`, `redirect`, `HttpResponse`.

Рекомендуемые ресурсы

Документация:

<https://docs.djangoproject.com/en/1.11/topics/http/shortcuts/>

Дополнительная информация:

<https://habrahabr.ru/post/50147/>