Автор курса

Антон Мазун

Full-stack Python developer
Тренер-консультант CBS

Python&Django

Тема

Маршрутизация. Запросы и ответы сервера

Как Django обрабатывает запрос

1. Django определяет какой корневой модуль URLconf использовать. Обычно, это значение настройки ROOT_URLCONF.
2. Django загружает модуль конфигурации URL и ищет переменную urlpatterns. Это должен быть список экземпляров django.conf.urls.url().
3. Django перебирает каждый URL-шаблон по порядку, и останавливается при первом совпадении с запрошенным URL-ом.
4. Если одно из регулярных выражений соответствует URL-у, Django импортирует и вызывает соответствующее представление, которое является просто функцией Python.
5. Если ни одно регулярное выражение не соответствует, или возникла ошибка на любом из этапов, Django вызывает соответствующий обработчик ошибок.


RegEx

Синтаксис регулярных выражений

Перечни символов ([abc]) abc[xyz]t = >abcxt , abcyt , abczt.

- [az-] , [-az] , [a-z]-все 26 малых латинских букв от 'a' до 'z'

Метасимволы

- ^ - начало строки ,
- $ - конец строки,
- \A - начало текста ,
- \Z -конец текста
- . любой символ в строке

Синтаксис регулярных выражений

Метасимволы - стандартные перечни символов:

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

Метасимволы-варианты:

- test(qwe|abc) находит ‘testqwe' или ‘testabc'.

Синтаксис регулярных выражений

Метасимволы - повторения

* ноль или более раз ("жадный")

\+ один или более раз ("жадный")

? ноль или один раз ("жадный"),

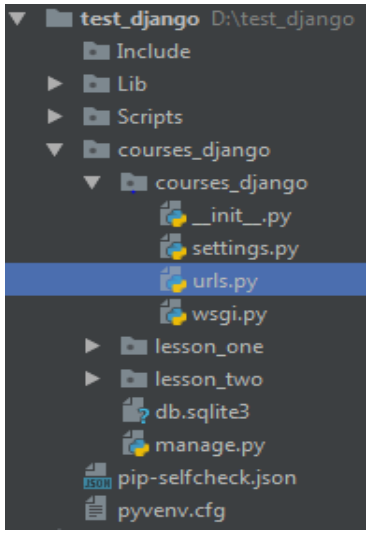
{n} точно n раз ("жадный")

{n,} не менее n раз ("жадный")

{n,m} не менее n но не более m раз ("жадный")

Маршрутизация

Маршрутизация внутри всего проекта



Внутри проекта courses_django -> urls.py задать ссылку на приложение

```
from lesson_two import views
```

Для регулярного выражения задать метод который будет обрабатывать запрос по ссылке с приложения

```
urlpatterns = [
    url(r'^$', views.home),
    url(r'^admin/', admin.site.urls),
]
```

Маршрутизация внутри приложения

```
urlpatterns = [
    url(r'', include('lesson_two.urls')),
    url(r'^admin/', admin.site.urls),
]
```

В файле urls.py проекта добавить ссылку на приложение

Маршрутизация с параметрами

Синтаксис Python именованных групп выглядит следующим образом : (?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")
```

Запрос к /item/2005/03/20/ вызовет функцию views.day_archive(request, year='2005', month='03‘ , day = ‘20’), вместо views.day_archive(request, '2005', '03‘ , ‘20’).

Алгоритм соответствия/группировки

Если существует именованный аргумент, он будет использован вместо позиционного аргумента.
Иначе все неименованные параметры будут переданы как позиционные аргументы.

Важно помнить

Найденные аргументы – всегда строки

Каждый найденный аргумент передается в представление как строка, независимо от того, какое “совпадение” определено в регулярном выражении. Например, URLconf содержит такую строку:

url(r'^item/(?P<year>[0-9]{4})/$', views.year_archive) аргумент year - строка!

Значения по умолчанию для аргументов представления

Book!

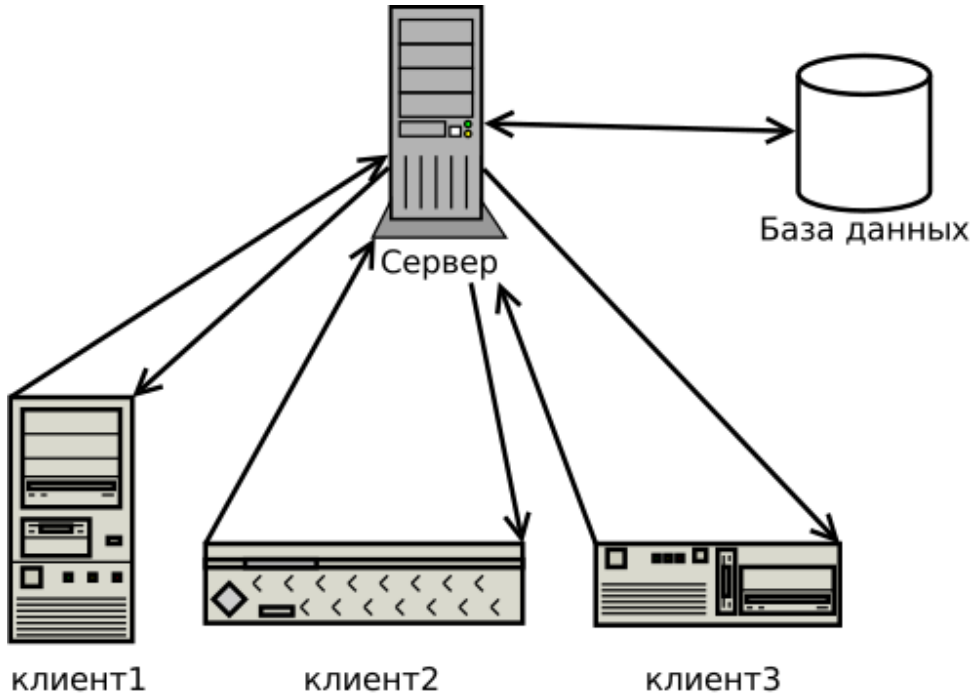
1. Page 1
2. Page 2
3. Page 3
4. Page 4
5. Page 5

Комбинирование URLconfs

- Добавление дополнительных URL-шаблонов с помощью списка экземпляров url()
- Указание общего префикса только один раз и сгруппировав различающиеся суффиксы.
- Вложенные аргументы
- Передача дополнительных аргументов в представлении

Запросы и ответы сервера

HTTP (HyperText Transfer Protocol) - протокол прикладного уровня передачи данных.



История развития HTTP

- **HTTP 0.9** (1991 Тим Бернерс-Ли)
- **HTTP 1.0** (1996)
- **HTTP 1.1** (1999)
- **HTTP 2.0** (2015)

HTTP методы

Метод	Описание
OPTIONS	Используется для определения возможностей веб-сервера или параметров соединения для конкретного ресурса. Также в заголовке ответа может включаться информация о поддерживаемых расширениях.
GET	Используется для запроса содержимого указанного ресурса. С помощью метода GET можно также начать какой-либо процесс
HEAD	Аналогичен методу GET, за исключением того, что в ответе сервера отсутствует тело.
POST	Применяется для передачи пользовательских данных заданному ресурсу.
PUT	Применяется для загрузки содержимого запроса на указанный в запросе URI
PATCH	Аналогично PUT, но применяется только к фрагменту ресурса.
DELETE	Удаляет указанный ресурс.
TRACE	Возвращает полученный запрос так, что клиент может увидеть, какую информацию промежуточные серверы добавляют или изменяют в запросе.
LINK	Устанавливает связь указанного ресурса с другими.
UNLINK	Убирает связь указанного ресурса с другими.

Код состояния

1xx Informational	100 Continue (Продолжать) 101 Switching Protocols (Переключение протоколов) 102 Processing (Идёт обработка)
2xx Success	200 OK (Успешно). 201 Created (Создано) 202 Accepted (Принято) 204 No Content (Нет содержимого) 206 Partial Content (Частичное содержимое)
3xx Redirection	300 Multiple Choices (Множественный выбор) 301 Moved Permanently (Перемещено навсегда) 304 Not Modified (Не изменялось)
4xx Client Error	401 Unauthorized (Неавторизован) 402 Payment Required (Требуется оплата) 403 Forbidden (Запрещено) 404 Not Found (Не найдено) 405 Method Not Allowed (Метод не поддерживается) 406 Not Acceptable (Не приемлемо) 407 Proxy Authentication Required (Требуется аутентификация прокси)
5xx Server Error	500 Internal Server Error (Внутренняя ошибка сервера) 502 Bad Gateway (Плохой шлюз) 503 Service Unavailable (Сервис недоступен) 504 Gateway Timeout (Шлюз не отвечает)

Структура запроса

Строка запроса — указан метод запроса (HTTP-метод), URI, версия протокола;
Заголовки — характеризуют тело сообщения, параметры передачи и прочие сведения;
Тело сообщения — данные сообщения.

Строка запроса

GET http://www.google.com/ HTTP/1.1

Django shortcut

django.shortcuts - методы ответов

HttpResponse - ответ сервера –строка

redirect - перенаправление на указаный адрес (абсолютный)

render – возвращает шаблон с контекстными данными
render(*request*, *template_name*, *context=None*, *content_type=None*, *status=None*, *using=None*)

render_to_response(*template_name*, *context=None*, *content_type=None*, *status=None*, *using=None*)

Спасибо за внимание!

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association