Разработка программного обеспечения на языке Python

Лекция 2. Библиотека Django

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

Параметры представлений в Django

В предыдущей теме мы рассмотрели инструменты для добавления навигации в приложение django: представления и маршрутизатор url. Далее опишем варианты передачи параметров от пользователя на серверную часть приложения.

Важным моментом для работы веб-приложения является возможность для пользователя отправлять данные приложению, для их обработки или записи в базу данных. В джанго существуют различные варианты передачи данных от клиентской части приложения - в параметрах представлений, строки запроса или заполняя формы. Сейчас рассмотрим первые два варианта, третий вариант - в одной из следующих тем.

Функции-представления могут принимать параметры, через которые могут передаваться различные данные. Подобные параметры передаются в адресе URL и могут быть связанны с параметрами функции-представления через систему маршрутизации. **Рассмотрим возможность передачи параметров на примере.**

views.py

```python
from django.shortcuts import render
from django.http import HttpResponse


def hello(request, name=""):
    return HttpResponse(f"<h2>Hello, {name}</h2>")

```

Определим в приложении в файле **views.py** функцию hello, которая кроме параметра request, принимает также параметр name - имя. Отправляемый пользователю ответ содержит значение этого параметра.

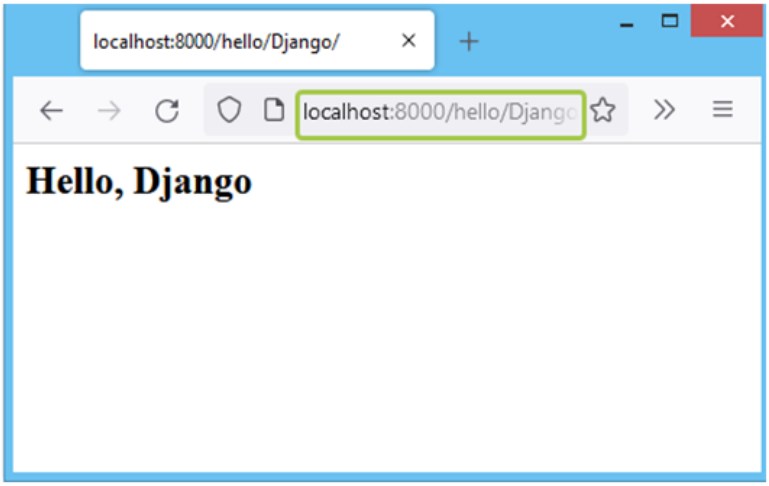
Теперь изменим файл **urls.py**, чтобы он мог сопоставить данные функции с запросами. Параметры заключаются в угловые скобки в формате <спецификатор:название_параметра> и определяют формат параметра, например, что он представляет собой строку.

urls.py

```python
from django.urls import path
from .views import *

urlpatterns = [
    path('', index),
    path('about', about),
    path('contact', contact),
    path('hello/<str:name>/', hello),
]
```

В этом случае при обращении к маршруту hello задание параметра является обязательным.



От параметров, которые передаются через адрес URL, следует отличать параметры, которые передаются через строку запроса. Параметры строки запроса указывается в браузере в адресной строке после маршрута и вопросительного знака ?. Каждый параметр предсталяет пару ключ-значение, например, в id=3 : id - название или ключ параметра, а 3 - его значение. Для получения параметров из строки запроса применяется метод **request.GET.get()**.

Например, определим в файле **views.py** функцию info.

views.py

```
from django.shortcuts import render
from django.http import HttpResponse


def info(request):
    # получение параметров name, age
    # "unknown", "-" значения параметров по умолчанию
    name = request.GET.get("name", "unknown")
    age = request.GET.get("age", "-")
    if name == 'unknown' and age == '-':
        res = "<h2>Hello</h2>"
    else:
        res = f"<h2>Hello, {name}, {age} years </h2>"
    return HttpResponse(res)
```

В ней из строки запроса получаем значения параметров имя и возраст. В методах request.GET.get("name", "") - первый аргумент - это название параметра строки запроса, значение которого надо извлечь, а второй аргумент - значение по умолчанию, если вдруг в строке запроса не оказалось подобного параметра.

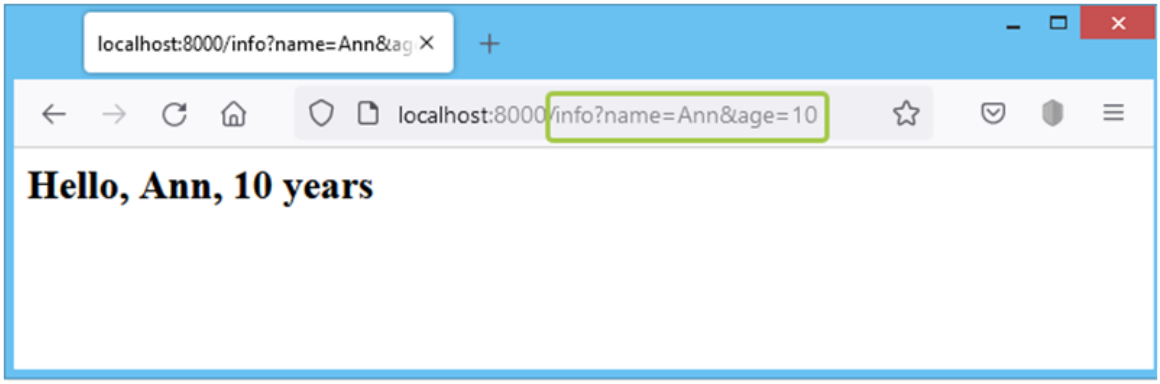
В файле **urls.py** определим маршрут для этой функции.

urls.py

```
from django.urls import path
from .views import *

urlpatterns = [
    path('', index),
    path('about', about),
    path('contact', contact),
    path('hello/<str:name>/', hello),
    path('info', info),
]
```

При переходе по маршруту info можно ввести в адресной строке браузера параметры имени и возраста, которые затем увидим в ответе от сервиса. В другом случае увидим просто приветствие.



При перемещении документа с одного адреса на другой мы можем воспользоваться механизмом переадресации, чтобы указать пользователям, что документу теперь доступен по новому адресу. Переадресация бывает временная и постоянная. Для создания временной переадресации применяется класс **HttpResponseRedirect**, а для постоянной - класс **HttpResponsePermanentRedirect**. В качестве примера определим в файле **views.py** следующий код. При обращении к функции contact она будет перенаправлять на маршрут "about", который будет обрабатываться функцией about.

```
from django.http import HttpResponse, HttpResponseRedirect


def index(request):
    return HttpResponse("Index")


def about(request):
    return HttpResponse("About")


def contact(request):
    return HttpResponseRedirect("/about")
```

Подведем итоги, были рассмотрены некоторые варианты передачи пользователем данных на сервер. Ответы на поступившие от пользователей запросы часто возвращаются ему в виде HTML-страниц. В свою очередь, эти страницы формируются на основе шаблонов, которые изучим далее.

Вопросы

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◄ Задание 2. Создание flask приложения

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

Задание 3. Создание своего первого сайта на Django ►

Контакты +7(391) 206-27-05
info-ms@sfu-kras.ru