## Получение данных запроса. HttpRequest

Функции-представления в качестве обязательного параметра получают объект **HttpRequest**, который хранит информацию о запросе. HttpRequest определяет ряд атрибутов, которые хранят информацию о запросе. Выделим следующие из них:

**scheme**: схема запроса (http или https)

**body**: представляет тело запроса в виде строки байтов

**path**: представляет путь запроса

**method**: метод запроса (GET, POST, PUT и т.д.)

**encoding**: кодировка

**content\_type**: тип содержимого запроса (значение заголовка CONTENT\_TYPE)

**GET**: объект в виде словаря, который содержит параметры запроса GET

**POST**: объект в виде словаря, который содержит параметры запроса POST

**COOKIES**: отправленные клиентом куки

**FILES**: отправленные клиентом файлы

**META**: хранит все доступные заголовки http в виде словаря. Набор заголовков зависит от клиента и сервера, некоторые из них:

* + CONTENT\_LENGTH: длина содержимого.
  + CONTENT\_TYPE: MIME-тип запроса.
  + HTTP\_ACCEPT: типы ответа, которые принимает клиент.
  + HTTP\_ACCEPT\_ENCODING: кодировка, в которой клиент принимает ответ.
  + HTTP\_ACCEPT\_LANGUAGE: язык ответа, который принимает клиент.
  + HTTP\_HOST: хост сервера.
  + HTTP\_REFERER: страница, с которой клиент отправил запрос (при ее наличии).
  + HTTP\_USER\_AGENT: юзер-агент или информация о браузере клиента.
  + QUERY\_STRING: строка запроса.
  + REMOTE\_ADDR: IP-адрес клиента.
  + REMOTE\_HOST: имя хоста клиента.
  + REMOTE\_USER: аутентификационные данные клиента (при наличии)
  + REQUEST\_METHOD: тип запроса (GET, POST).
  + SERVER\_NAME: имя хоста сервера.
  + SERVER\_PORT: порт сервера.

**headers**: заголовки запроса в виде словаря

Также HttpRequest определяет ряд методов. Отметим следующие из них:

**get\_full\_path()**: возвращает полный путь запроса, включая строку запроса

**get\_host()**: возвращает хост клиента, для этого используется значения заголовков HTTP\_X\_FORWARDED\_HOST (если включена опция USE\_X\_FORWARDED\_HOST) и HTTP\_HOST

**get\_port()**: возвращает номер порта

Например, получим некоторую информацию о запросе. Для этого в файле **views.py**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | from django.http import HttpResponse    def index(request):      host = request.META["HTTP\_HOST"] # получаем адрес сервера      user\_agent = request.META["HTTP\_USER\_AGENT"]    # получаем данные бразера      path = request.path     # получаем запрошенный путь        return HttpResponse(f"""          <p>Host: {host}</p>          <p>Path: {path}</p>          <p>User-agent: {user\_agent}</p>      """) |

В данном случае получаем два заголовка "HTTP\_HOST" и "HTTP\_USER\_AGENT" и запрошенный путь.

В файле **urls.py** зарегистрируем данную функцию:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | from django.urls import path  from hello import views    urlpatterns = [      path("index", views.index),  ] |

Результат работы:

