## HttpResponse и отправка ответа

Для отправки ответа клиенту в Django применяется класс **HttpResponse** из пакета **django.http**. В общем случае для отправки некоторых данных достаточно эти данные передать в конструктор HttpResponse. Например, пусть в файле **views.py** имеется простейшая функция-представление, которая отправляет ответ клиенту:

|  |  |
| --- | --- |
| 1  2  3  4 | from django.http import HttpResponse    def index(request):      return HttpResponse("Hello") |

И в файле **urls.py** эта функция соотносится с некоторым маршрутом:

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

Подобная функция просто передает в HttpResponse некоторый текст, который пользователь затем увидит в браузере. Однако подобной функциональностью HttpResponse не ограничивается. Так, функция инициализации класса определяет несколько параметров:

|  |  |
| --- | --- |
| 1 | HttpResponse.\_\_init\_\_(content=b'', content\_type=None, status=200, reason=None, charset=None, headers=None) |

Параметры:

**content**: содержимое ответа в виде строки байтов. Если передается другое содержимое, то оно конвертируется в строку байтов

**content\_type**: MIME-тип ответа, устанавливает HTTP-заголовок Content-Type. Если этот параметр не установлен, то применяется mime-тип text/html и значение настройки DEFAULT\_CHARSET, то есть в итоге будет: "text/html; charset=utf-8".

**charset**: кодировка ответа в виде строки. По умолчанию django пытается установить кодировку из параметра content\_type, а в случае неудачи для установки кодировки применяется настройка DEFAULT\_CHARSET.

**status**: статусный код ответа. По умолчанию равно 200

**reason\_phrase**: сообщение, которое отправляется в вместе статусным кодом

**headers**: заголовки ответа в виде словаря

Для хранения отправляемых данных он определяет ряд атрибутов. Некоторые из них:

**content**: содержимое ответа в виде строки байтов

**headers**: отправляемые заголовки в виде словаря

**charset**: кодировка ответа в виде строки. По умолчанию django пытается установить кодировку из заголовка content\_type, а в случае неудачи для установки кодировки применяется настройка DEFAULT\_CHARSET.

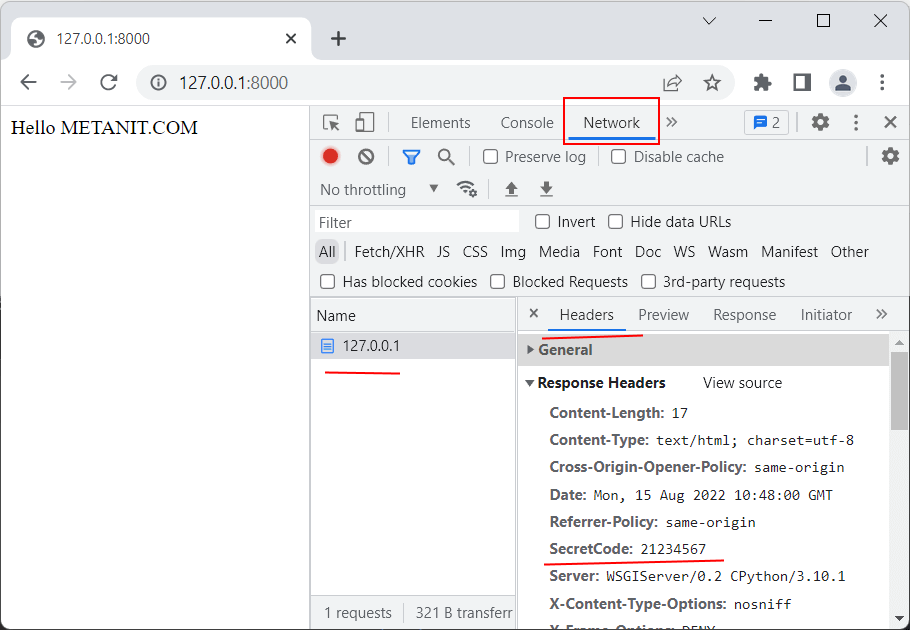
**status\_code**: статусный код ответа

**reason\_phrase**: сообщение, которое отправляется в вместе статусным кодом

Рассмотрим некоторые возможности. Например, изменим определение функции в файле **views.py**:

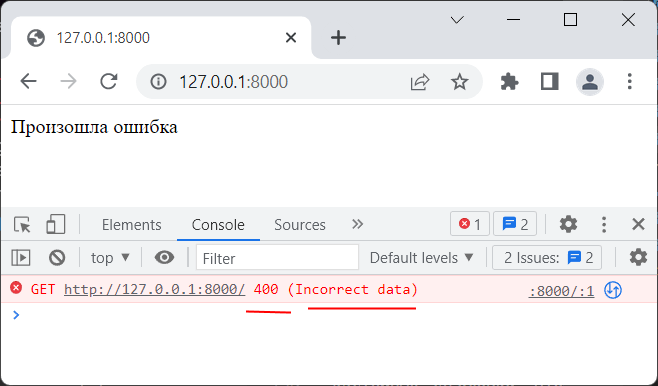
|  |  |
| --- | --- |
| 1  2  3  4 | from django.http import HttpResponse    def index(request):      return HttpResponse("Hello METANIT.COM", headers={"SecretCode": "21234567"}) |

В данном случае также устанавливается заголовок "SecretCode". Хотя в реальности в HTTP не существует такого заголовка, но мы можем определять кастомные заголовки, чтобы передать через них клиенту какую-нибудь информацию. Например, после обращения к функции мы можем в браузере через инструменты разработчика проинспектировать отправленные сервером заголовки и найти там в том числе заголовок "SecretCode".



Подобным образом можно установить другие параметры, например, статусный код и сообщение к нему:

|  |  |
| --- | --- |
| 1  2  3  4 | from django.http import HttpResponse    def index(request):      return HttpResponse("Произошла ошибка", status=400, reason="Incorrect data") |



Установка содержимого и кодировки:

|  |  |
| --- | --- |
| 1  2  3  4 | from django.http import HttpResponse    def index(request):      return HttpResponse("<h1>Hello</h1>", content\_type="text/plain", charset="utf-8") |

Хотя в содержимом ответа применяются теги html (<h1>), но браузере теперь будет рассматривать это содержимое как простой текст, потому что установлен заголовок "text/plain":

