Сообщить об ошибке.

ХОЧУ ПОМОЧЬ ПРОЕКТУ

Модуль http.server в Python



Sberbusiness.live РЕКЛАМА

Секреты заработка на сапсерфинге в городах России

Лайфхаки • Инструкции • Статьи • Личный опыт

Узнать больше

Бесплатное Модуль http.server в Python занятие английским ерверов в Яндекс лассы для реализации НТТР-серверов (веб-серверов). Практикуме er не рекомендуется для использования в продакшне, так как он выполняет только Полноценное занятие с преподавателем, а не презентация курсов вляется подклассом <u>socketserver.TCPServer()</u>. Он создает и прослушивает HTTP-сокет, Устный тест на уровень языка ера выглядит так: Практика английского r, handler_class=BaseHTTPRequestHandler): Узнать больше r_address, handler_class) httpd.serve_forever()

Классы НТТР веб-серверов.

http.server.HTTPServer(server_address, RequestHandlerClass):

Класс HTTPServer основан на классе TCPServer, сохраняя адрес сервера в виде переменных экземпляра с именами server_name и server_port.

Сервер доступен обработчику, обычно через переменную экземпляра сервера обработчика RequestHandlerClass.

http.server.ThreadingHTTPServer(server_address, RequestHandlerClass):

Knacc ThreadingHTTPServer идентичен <u>HTTPServer</u>, но использует потоки для обработки запросов с помощью миксина <u>ThreadingMixIn</u>.

Такое поведение полезно для обработки запросов веб-браузеров, предварительно открывающих сокеты, на которых HTTPServer будет ждать бесконечно.

Классы <u>HTTPServer</u> и <u>ThreadingHTTPServer</u> должны получить в качестве аргумента RequestHandlerClass при создании экземпляра, из которых этот модуль предоставляет три различных варианта.

Обработчики НТТР-запросов:

- 1. http:server.BaseHTTPRequestHandler() класс используется для обработки HTTP-запросов, поступающих на сервер;
- 2. http.server.SimpleHTTPRequestHandler() обслуживает файлы из текущего каталога, напрямую сопоставляя структуру каталогов с HTTP-запросами;
- 3. http.server.CGIHTTPRequestHandler() обслуживает файлы или вывод сценариев CGI из текущего каталога.

Примеры создания простых HTTP web-серверов

Пример HTTP web-сервера, обрабатывающего статические файлы.

```
python3 -m http.server 8000 --bind 127.0.0.1
```

Вверх

Если запустить HTTP сервер таким способом, то можно создать, например, фиктивный API путем загрузки на этот сервер статических файлов json или/и xml. Структура ресурсов файлов, организованных в подпапки, будет обеспечивать RESTful-подобные URL-адреса. Например, /users/all.json может содержать фиктивные записи пользователей. Этот подход еще быстрее, чем создание, например, приложения <u>Flask</u>. База данных не требуется, работает везде. Можно применять

ОР

е трудности с командой терминала scp для загрузки данных. Решение - можно запустить ине и загрузить необходимое содержимое через HTTP.

ра, с поддержкой SSL.

ервера создайте следующий модуль:

```
п practicum.yandex.ru
Бесплатное
занятие
английским
в Яндекс
Практикуме
```

Полноценное занятие с преподавателем, а не презентация курсов

Устный тест на уровень языка

Практика английского

Узнать больше

```
rver, BaseHTTPRequestHandler

', 4443), BaseHTTPRequestHandler)

(httpd.socket,
em",
.pem', server_side=True)
```

ртификатов с помощью OpenSSL используйте следующую команду

```
2048 -keyout key.pem -out cert.pem -days 365
```

ba, обрабатывающего запросы GET и POST.

_________epвера, который отвечает Hello, world! на запросы. <u>Обратите внимание</u>, что self.send_response(200) и self.end_headers() являются обязательными, в противном случае ответ не будет считаться действительным. Проверить работу сервера можно при помощи утилиты терминала curl.

```
from http.server import HTTPServer, BaseHTTPRequestHandler

class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):

# определяем метод `do_GET`

def do_GET(self):
    self.send_response(200)
    self.end_headers()
    self.wfile.write(b'Hello, world!')

httpd = HTTPServer(('localhost', 8000), SimpleHTTPRequestHandler)
httpd.serve_forever()
```

<u>Обратите внимание</u>, что self.wfile является файлоподобным объектом, поэтому ожидает <u>байтоподобных объектов</u> для функции записи. Другой способ передать информацию в self.wfile - это использование объекта <u>io.BytesIO</u>.

```
from http.server import HTTPServer, BaseHTTPRequestHandler

from io import BytesIO

class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):

# определяем метод 'do_GET'
def do_GET(self):
    self.send_response(200)
    self.send_header('Content-type', 'text/html')
    self.end_headers()
    self.wfile.write(b'Hello, world!')

BBEPX еделяем метод 'do_POST'
def do_POST(self):
```

```
content_length = int(self.headers['Content-Length'])
body = self.rfile.read(content_length)
self.send_response(200)
self.send_header('Content-type', 'text/html')

PEKJAMA-18-9

is POST request. ')
ved: ')
nse.getvalue())

practicum.yandex.ru

practicum.yandex.ru

becплатное

занятие

cotyп через self.rfile. Это BufferedReader, поэтому для получения содержимого должен
```

ьесплатное занятие английским в Яндекс Практикуме

Полноценное занятие с преподавателем, а не презентация курсов

Устный тест на уровень языка

Практика английского

Узнать больше

оступ через self.rfile. Это BufferedReader, поэтому для получения содержимого должен]). <u>Обратите внимание</u>, что функции должен быть явно передан размер, иначе запрос чится.

ngth необходимо. Он может быть извлечен через self.headers и преобразован в целое ает обратно все, что он получает.

сервера Python?

нтерпретатор Python, например python3 myserver.py, просто нажмите Crtl + C.

ть его с помощью кода, то надо создать какое-либо условие, или точку, или вать команду на выключение. Например можете добавить блок и вызвать httpd.shutdown() является подклассом <u>SocketServer.TCPServer</u>. В свою очередь класс подклассом <u>SocketServer.BaseServer</u> у которого есть метод BaseServer.shutdown().

Из вышесказанного можно заключить то, что класс HttpServer так же будет имеет метод .shutdown(), как унаследованный от класса SocketServer.BaseServer.

```
import os
from http.server import SimpleHTTPRequestHandler, HTTPServer

os.chdir('c:/users/owner/desktop/pyserver')
server_address = ('', 8000)
try:
    httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
    httpd.serve_forever()
except Exception:
    httpd.shutdown()
```

Meтog httpd.shutdown() говорит серверу, что бы цикл httpd.serve_forever() остановился, при этом дождался завершения обработки запросов подключенных клиентов.



Содержание раздела:

• <u>КРАТКИЙ ОБЗОР МАТЕРИАЛА.</u>



DOCS-Python.ru™, 2023 Γ.

practicum.yanuex.ru

Бесплатное занятие английским в Яндекс Практикуме

Полноценное занятие с преподавателем, а не презентация курсов

Устный тест на уровень языка

Практика английского

Узнать больше

inogyna http.server
index http.server
index http.server
index http.server
index http.server

(Внимание! При копировании материала ссылка на источник обязательна)

@docs_python_ru

