

Пишем echo-сервер

В этом уроке необходимо будет создать свой echo-сервер. Приставка "echo" означает, что этот веб-сервер на любой запрос должен отвечать ответом, содержащим детали запроса.

Пример 1. На GET-запрос, посланный по адресу 0.0.0.0:8888/index, ваш сервер должен возвращать такое:

```
{
  "method": "GET",
  "host": "0.0.0.0:8080",
  "path": "/index",
  "query_string": "",
  "query": {},
  "scheme": "http",
  "secure": false,
  "http_version": "1.1",
  "headers": {
    "Host": "0.0.0.0:8080",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.3 Safari/605.1.15",
    "Accept-Language": "en-us",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "keep-alive"
  },
  "cookies": {},
  "json": null
}
```

Пример 2. Для POST-запроса, посланном по адресу 0.0.0.0:8888/method?a=b&b=a, с заголовком "X-Header: value", кукой "xcookie=value" и json-нагрузкой {"data": "value"}, сервер должен возвращать примерно такое:

```
{
  "method": "POST",
  "host": "0.0.0.0:8080",
  "path": "/method",
  "query_string": "a=b&b=a",
  "query": {"a": "b", "b": "a"},
  "scheme": "http",
  "secure": false,
  "http_version": "1.1",
  "headers": {
    "Host": "0.0.0.0:8080",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.3 Safari/605.1.15",
    "Accept-Language": "en-us",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "keep-alive",
    "Content-Type": "application/json",
    "X-Header": "value"
  },
  "cookies": {
    "xcookie": "value"
  },
  "json": {
    "data": "value"
  }
}
```

Важные моменты

1. Необходимо, чтобы сервер мог ответить на запросы с любым URI. Например такие: /, /index, /query?a=b. Этого можно достичь с помощью [динамических путей](#), но не только.
2. Необходимо, чтобы сервер мог ответить на запросы с любым HTTP-методом из этих: HEAD, GET, POST, DELETE, PUT, PATCH и OPTIONS. Этого можно достичь с помощью [class-based view](#).
3. Необходимо возвращать данные о запросе в json-формате. В этом вам поможет функция [json_response](#).

Ссылка на сервер для выполнения находится в следующей карточке.

Задание находится в папке `echo_server`.

Запустить сервер можно командой в терминале `./run_echo_server`, а запустить тесты для этого задания командой `./run_echo_tests`

Внимание! Меркури-задание будет считаться выполненным, когда все тесты для всех задач будут пройдены. Это произойдет только после выполнения всех заданий этой главы. Не волнуйтесь об этом.