

Streaming

[документация](#)

Примеры из документации aiohttp

```
# пример 1
with open('massive-body', 'rb') as f:
    await session.post('http://httpbin.org/post', data=f)

# пример 2
async def file_sender(file_name=None):
    async with aiofiles.open(file_name, 'rb') as f:
        chunk = await f.read(64*1024)
        while chunk:
            yield chunk
            chunk = await f.read(64*1024)

async with session.post('http://httpbin.org/post', data=file_sender(file_name='huge_file')) as resp:
    print(await resp.text())
```

Библиотека aiohttp предоставляет удобный интерфейс для отправки данных по частям (чанкам). Функция **session.post** (параметр data) умеет принимать в качестве аргумента:

- итераторы (пример 1)
- асинхронные итераторы (пример 2)

Давайте рассмотрим пример 2 детальнее:

1. [библиотека aiofiles](#) позволяет асинхронно, частями, читать с диска данные: **chunk = await f.read(64*1024)**
2. читаем, пока **f.read** возвращает данные
3. прочтенный чанк передаем в `yield`. В этот момент управление вернется обратно в функцию `session.post`
4. внутри корутина `session.post` получает `chunk`, который был передан через `yield`, и сразу же отправляет эту часть на сервер, при этом не храня ее у себя в памяти
5. так происходит до тех пор, пока не будет передан весь файл

В результате нужная для отправки файла память равна размеру чанка, что существенно меньше размера файла.

На заметку — `aiofiles` нечестная асинхронная библиотека. Она использует синхронный `open` и `run_in_executor`. Исходный код можно посмотреть [тут](#).