

Aiohttp



КИРИЛЛ ТАБЕЛЬСКИЙ



КИРИЛЛ ТАБЕЛЬСКИЙ

Lightmap



План занятия

1. [Что такое Aiohttp?](#)
2. [Установка Aiohttp](#)
3. [Middleware](#)
4. [Дополнительные материалы](#)
5. [Домашнее задание](#)

Что такое aiohttp

- Асинхронный http-клиент. Асинхронная замена requests.
- Асинхронный http-сервер. Асинхронная замена flask.



Установка aiohttp

```
pip install aiohttp  
pip install cchardet #необязательно, но крайне желательно  
pip install aiodns   #необязательно, но крайне желательно
```

Пример из документации

```
async with aiohttp.ClientSession() as session: # создаем сессию
    async with session.get('http://httpbin.org/get') as resp: # отправляем запрос
        await resp.text() # ожидаем текстовое представление ответа
```

Aiohttp сервер прост как flask

```
from aiohttp import web

routes = web.RouteTableDef()

@routes.get('/')
async def hello(request):
    return web.Response(text="Hello, world")

@routes.get('/{name}')
async def hello_name(request):
    name = request.match_info['name']
    return web.Response(text=f"Hello, {name}")

@routes.get(r'/1/{name:\d+}')
async def hello_name_regexp(request):
    name = request.match_info['name']
    return web.Response(text=f"buy, {name}")

app = web.Application()
app.add_routes(routes)
web.run_app(app)
```

Альтернативный способ добавления роутов

```
app.add_routes([web.get('/path1', get_1),  
                web.post('/path1', post_1),  
                web.get('/path2', get_2),  
                web.post('/path2', post_2)])
```

Вью может быть классом

```
@routes.view('/{name}')
class NameView(web.View):
    async def get(self):
        name = self.request.match_info['name']
        return web.Response(text=f"Hello, {name}")
```

Middleware

Пример из документации.

```
@web.middleware
async def error_middleware(request, handler):
    try:
        response = await handler(request)
        if response.status != 404:
            return response
        message = response.message
    except web.HTTPException as ex:
        if ex.status != 404:
            raise
        message = ex.reason
    return web.json_response({'error': message})
```

```
app = web.Application(middlewares=[error_middleware])
```



Дополнительные материалы

- <https://aiohttp.readthedocs.io/en/stable/>
- [Пример боевого проекта](#)



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаём в чате !
- Задачи можно сдавать по частям.
- Зачёт по домашней работе проставляется после того, как приняты **все задачи**.

 нетология

Задавайте вопросы и напишите отзыв о лекции!

КИРИЛЛ ТАБЕЛЬСКИЙ