

# ЭНДПОИНТЫ И ПАРАМЕТРЫ URL

## ВВЕДЕНИЕ

### Задачи урока

- 01** Узнать, что такое эндпоинт и зачем он нужен при работе с веб-сервисами и API
- 02** Рассмотреть примеры эндпоинтов для различных функциональных возможностей, таких как получение данных, создание ресурсов и обновление информации

### Эндпоинт endpoint

это конечная точка веб-сервиса или API (интерфейса программирования приложений), доступная для отправки запросов и получения ответов. Эндпоинты определяются на стороне сервера и предоставляют определенные функциональные возможности, которые клиентские приложения могут использовать для взаимодействия с сервером

Каждый эндпоинт имеет уникальный URL-адрес и обычно связан с определенным HTTP-методом, таким как GET, POST, PUT или DELETE. Клиентские приложения отправляют запросы на указанные URL-адреса эндпоинтов и ожидают ответа от сервера. В ответе, сервер может возвращать данные, изменять состояние сервера или выполнять другие операции в соответствии с запросом

Обычно эндпоинты используются для организации логических группировок функциональных возможностей или ресурсов, доступных через API. Например, веб-сервис, предоставляющий информацию о пользователях, может иметь эндпоинты для получения списка пользователей, создания нового пользователя, обновления информации о пользователе и т. д

Все взаимодействие между клиентом и сервером происходит через эндпоинты, поэтому понимание концепции эндпоинтов важно при разработке и использовании веб-сервисов и API

В FastAPI вы можете принимать запросы типа GET с помощью декоратора `app.get()` или `app.get("/путь")`, где `app` - экземпляр FastAPI, а путь - URL-адрес, по которому будет доступен эндпоинт

Для примера, давайте создадим простой эндпоинт для обработки запросов GET:

```
from fastapi import FastAPI

# Создаем экземпляр FastAPI
app = FastAPI()

# Определяем GET-метод для эндпоинта "/"
@app.get("/")
def read_root():
    return {"message": "Hello, World!"}

@app.get("/user/{user_id}")
def get_user(user_id: int):
    return {"user_id": user_id}
```

В этом примере мы добавили переменную `user_id` в URL-адресе, обозначенную фигурными скобками `{}`. Затем мы объявили эту переменную как аргумент функции `get_user()` с ожидаемым типом данных `int`. Когда мы делаем запрос по адресу `/user/123`, FastAPI автоматически приводит `user_id` к типу `int` и передает его в функцию

Также можно описать принимаемые параметры которые необходимо передать в теле запроса:

```
class Hello(BaseModel):
    Name: str
    Age: int

app = FastAPI()

@app.get("/hello")
def read_root(hello: Hello):
    return {"message": f"Hello, {hello.Name}!"}
```

## ИТОГИ

- ✓ Узнали, что такое эндпоинт и как он используется при работе с веб-сервисами и API
- ✓ Научились создавать эндпоинты для обработки запросов типа GET в FastAPI с помощью соответствующих декораторов