УНИВЕРСИТЕТ СИНЕРГИЯ

РАБОТА С API

Продвинутый Python

Андрей Вильмов

СОДЕРЖАНИЕ ТЕМЫ

01 Как кэшировать используя Redis

02 Как запускать фоновые задачи

03 Как подключаться к Redis

Определение понятия

КЭШИРОВАНИЕ

это техника, позволяющая сохранять результаты выполнения запроса для повторного использования и увеличения производительности системы

ПОДКЛЮЧЕНИЕ К REDIS

Для подключения используйте **RedisBackend**

```
...
@app.on_event("startup")
async def startup_event():
    redis = aioredis.from_url("redis://localhost",
    encoding="utf8", decode_responses=True)
    FastAPICache.init(RedisBackend(redis),
    prefix="fastapi-cache")
```

@CACHE

это **декоратор**, указывающий на кэширование результата функции на определенное время для повторного использования без повторного выполнения

СОЗДАНИЕ КЭША

Создания кэширования на 30 секунд

```
...
@router.get('/')
@cache(expire=30)
def get_all():
 …
```

CELERY

это распределенная система очередей задач для выполнения асинхронных и отложенных операций, часто используемая веб-приложениями для обработки задач в фоновом режиме

НАСТРОЙКА CELERY

Для создания фоновых задач необходимо настроить **Celery**

```
...
celery = Celery(
     'tasks', broker='redis://localhost:6379',
     broker_connection_retry_on_startup=True
)
```

СОЗДАНИЕ ФОНОВОЙ ЗАДАЧИ


```
...
@celery.task
def send_task(user_id: str):
    bot = telebot.TeleBot('Токен')
    bot.send_message(user_id, "Привет")
```

КАК РАБОТАЕТ CELERY

- Когда клиент отправляет задачу, она помещается в очередь брокера
- Затем исполнитель извлекает задачу из очереди и выполняет ее
- После завершения задачи результат может быть возвращен клиенту или сохранен для дальнейшего использования

КАК РАБОТАЕТ CELERY

Celery

Celery также предоставляет возможность масштабирования горизонтально путем запуска нескольких исполнителей, которые могут обрабатывать задачи параллельно

Celery обеспечивает асинхронность, распределение задач и масштабируемость, что делает его мощным инструментом для обработки задач в фоновом режиме в веб-приложениях

ЗАПУСК ФОНОВЫХ ЗАДАЧ

Для запуска сервера **Celery** который будет обрабатывать задачи в фоне выполним в консоли следующую строку:

```
...
celery -A tasks.send_message:celery worker --loglevel=INFO --pool=solo
```

ПОДВЕДЁМ ИТОГИ

- Подключение к Redis для кэширование делается при запуске сервиса
- У кэша есть время жизни и его можно задавать для каждого метода отдельно
- Для создания фоновых задач используем Celery который будет выполнять все переданные ему задачи в фоне