Лабораторная работа №8: Основы работы с планировщиками

ФИО: Ляхова Елизавета Олеговна
Группа: РИМ-150950

Цель работы: научиться создавать, запускать и управлять асинхронными и отложенными задачами с помощью TaskIQ.

Ход работы

1. Подготовка окружения

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS          PORTS
                                                       NAMES
d91bd522cdf6   postgres:15            "docker-entrypoint.s…"   52 minutes ago   Up 52 minutes   0.0.0.0:5432->5432/tcp, [::]:5432->5
432/tcp                                                lab8-postgres-1
de4d5918b05f   rabbitmq:3-management  "docker-entrypoint.s…"   52 minutes ago   Up 52 minutes   0.0.0.0:5672->5672/tcp, [::]:5672->5
672/tcp, 0.0.0.0:15672->15672/tcp, [::]:15672->15672/tcp   lab8-rabbitmq-1

C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>
```

2. Настройка TaskIQ

2.1 taskiq worker app.tasks:broker

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>taskiq worker app.broker:broker
[2025-12-14 21:18:16,959][taskiq.worker][INFO   ][MainProcess] Pid of a main process: 33136
[2025-12-14 21:18:16,960][taskiq.worker][INFO   ][MainProcess] Starting 2 worker processes.
[2025-12-14 21:18:16,981][taskiq.process-manager][INFO   ][MainProcess] Started process worker-0 with pid 30684
[2025-12-14 21:18:16,998][taskiq.process-manager][INFO   ][MainProcess] Started process worker-1 with pid 32596
[2025-12-14 21:18:19,681][taskiq.receiver.receiver][INFO   ][worker-0] Listening started.
[2025-12-14 21:18:19,681][taskiq.receiver.receiver][INFO   ][worker-1] Listening started.
```

2.2 tasks scheduler tasks.app:scheduler –skip-first-run

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>taskiq worker app.broker:broker
[2025-12-14 21:18:16,959][taskiq.worker][INFO   ][MainProcess] Pid of a main process: 33136
[2025-12-14 21:18:16,960][taskiq.worker][INFO   ][MainProcess] Starting 2 worker processes.
[2025-12-14 21:18:16,981][taskiq.process-manager][INFO   ][MainProcess] Started process worker-0 with pid 30684
[2025-12-14 21:18:16,998][taskiq.process-manager][INFO   ][MainProcess] Started process worker-1 with pid 32596
[2025-12-14 21:18:19,681][taskiq.receiver.receiver][INFO   ][worker-0] Listening started.
[2025-12-14 21:18:19,681][taskiq.receiver.receiver][INFO   ][worker-1] Listening started.
```

3. Создание планировщика

```python
from taskiq_aio_pika import AioPikaBroker
from taskiq import TaskiqScheduler
from taskiq.schedule_sources import LabelScheduleSource

# 1. Создаём брокер
```

```python
broker = AioPikaBroker(
    "amqp://guest:guest@localhost:5672/",
    exchange_name="report",
    queue_name="cmd_order"
)

# 2. Создаём планировщик
scheduler = TaskiqScheduler(
    broker=broker,
    sources=[LabelScheduleSource(broker)],
)

# 3. Периодическая задача (каждую минуту)
@broker.task(
    schedule=[
        {
            "cron": "*/1 * * * *",
            "args": [{"report_at": "2024-12-14", "order_id": 1, "count_product": 5}],
            "schedule_id": "generate_report_every_minute",
        }
    ]
)
async def generate_report(data: dict):
    """Задача, выполняемая каждую минуту для формирования отчёта"""
    import datetime
    print(f"⏰ [{datetime.datetime.now()}] Задача generate_report выполнена")
    print(f"📊 Данные: {data}")
    return {"status": "success", "timestamp": str(datetime.datetime.now())}

# 4. Простая тестовая задача
@broker.task
async def test_task(name: str) -> str:
```

```
    return f"Hello, {name}!"
```

4. Логи выполнения задачи

4.1 taskiq worker app.tasks:broker

```
C:\Users\lyaho>cd "C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8"

C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>taskiq worker app.tasks:broker
[2025-12-14 22:49:19,367][taskiq.worker][INFO   ][MainProcess] Pid of a main process: 34036
[2025-12-14 22:49:19,367][taskiq.worker][INFO   ][MainProcess] Starting 2 worker processes.
[2025-12-14 22:49:19,372][taskiq.process-manager][INFO   ][MainProcess] Started process worker-0 with pid 30996
[2025-12-14 22:49:19,376][taskiq.process-manager][INFO   ][MainProcess] Started process worker-1 with pid 16864
[2025-12-14 22:49:20,304][taskiq.receiver.receiver][INFO   ][worker-0] Listening started.
[2025-12-14 22:49:20,304][taskiq.receiver.receiver][INFO   ][worker-1] Listening started.
[2025-12-14 22:49:20,315][taskiq.receiver.receiver][INFO   ][worker-1] Executing task app.tasks:generate_report with ID:
 030693488875462 78abc411c93756c07
[2025-12-14 22:49:20,315][taskiq.receiver.receiver][INFO   ][worker-0] Executing task app.tasks:generate_report with ID:
 458addc1766b41e6a722b6081cfe36b6
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
[2025-12-14 22:49:20,320][taskiq.receiver.receiver][INFO   ][worker-1] Executing task app.tasks:generate_report with ID:
 baab7d841cd742a49572715793f37f70
[2025-12-14 22:49:20,320][taskiq.receiver.receiver][INFO   ][worker-0] Executing task app.tasks:generate_report with ID:
 64180700e6384f948ea71e4a82280e6f
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
[2025-12-14 22:49:20,323][taskiq.receiver.receiver][INFO   ][worker-1] Executing task app.tasks:generate_report with ID:
 6eb58204bd5f4033b5e4632d04fda808
[2025-12-14 22:49:20,323][taskiq.receiver.receiver][INFO   ][worker-0] Executing task app.tasks:generate_report with ID:
 d53ea64df00e4e5c948af71b1b1f527e
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
📍 Создание отчёта: {'report_at': '2024-12-14', 'order_id': 1, 'count_product': 5}
[2025-12-14 22:49:20,325][taskiq.receiver.receiver][INFO   ][worker-1] Executing task app.tasks:generate_report with ID:
```

4.2 tasks scheduler tasks.app:scheduler –skip-first-run

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>taskiq scheduler app.tasks:scheduler --skip-first-run
[2025-12-14 22:49:34,503][INFO   ][run:run_scheduler:404] Starting scheduler.
[2025-12-14 22:49:34,561][INFO   ][run:run_scheduler:406] Startup completed.
[2025-12-14 22:50:00,003][INFO   ][run:send:168] Sending task app.tasks:generate_report with schedule_id generate_report
_every_minute.
[2025-12-14 22:51:00,005][INFO   ][run:send:168] Sending task app.tasks:generate_report with schedule_id generate_report
_every_minute.
[2025-12-14 22:52:00,004][INFO   ][run:send:168] Sending task app.tasks:generate_report with schedule_id generate_report
_every_minute.
[2025-12-14 22:53:00,003][INFO   ][run:send:168] Sending task app.tasks:generate_report with schedule_id generate_report
_every_minute.
```

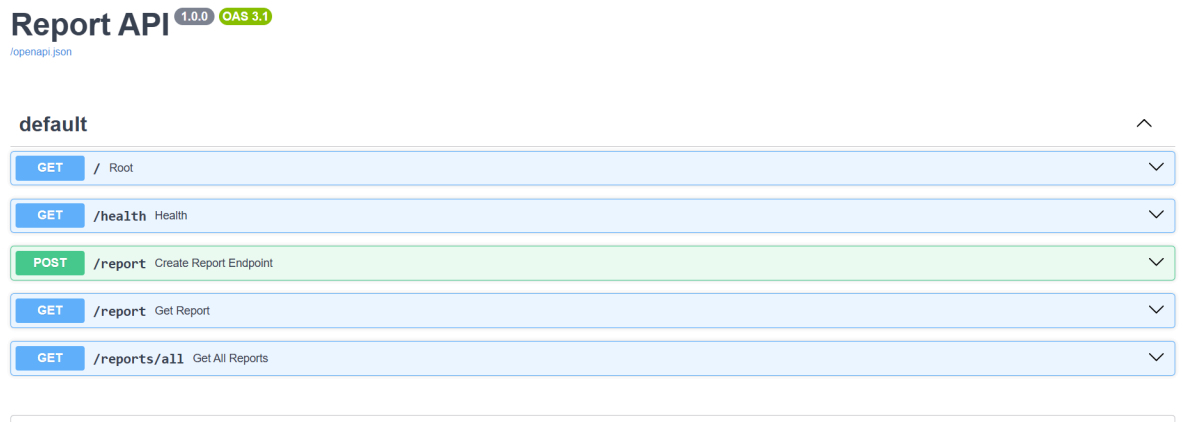
5. Создание таблицы

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>docker exec lab8-postgres-1 psql -U user -d reports_db -c "SELECT * FR
OM reports;"
 id | report_at | order_id | count_product | created_at
----+-----------+----------+---------------+------------
(0 rows)
```

5.1 Наполненная данным бд

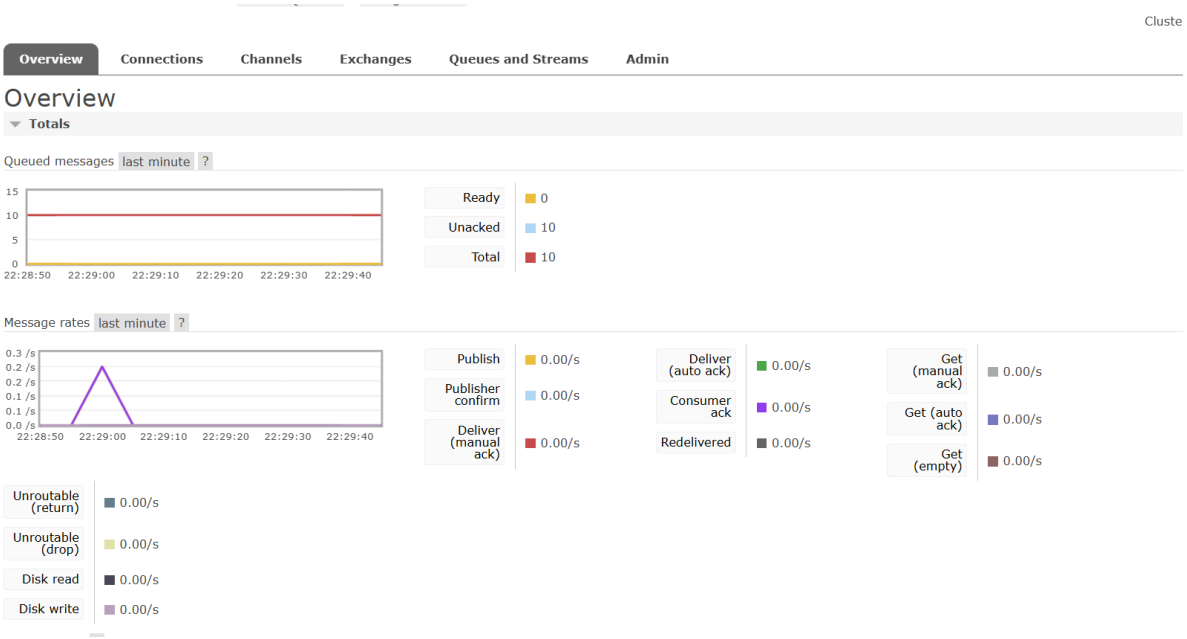
```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>docker exec lab8-postgres-1 psql -U user -d reports_db -c "SELECT * FR
OM reports;"
 id | report_at  | order_id | count_product |         created_at
----+------------+----------+---------------+----------------------------
  1 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.525436
  6 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.531279
  7 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.532646
  8 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.534412
  9 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.535663
  3 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.527313
 20 | 2025-12-14 |        1 |             5 | 2025-12-14 17:50:00.033355
  5 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.529761
  2 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.526395
 10 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.546629
 21 | 2025-12-14 |        1 |             5 | 2025-12-14 17:51:00.031609
 11 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.549642
 12 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.548054
 13 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.55046
 14 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.551504
 15 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.552782
 16 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.555027
 17 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.554098
 18 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.556805
 19 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.558561
  4 | 2025-12-14 |        1 |             5 | 2025-12-14 17:49:20.528853
(21 rows)
```

6. Документация REST API

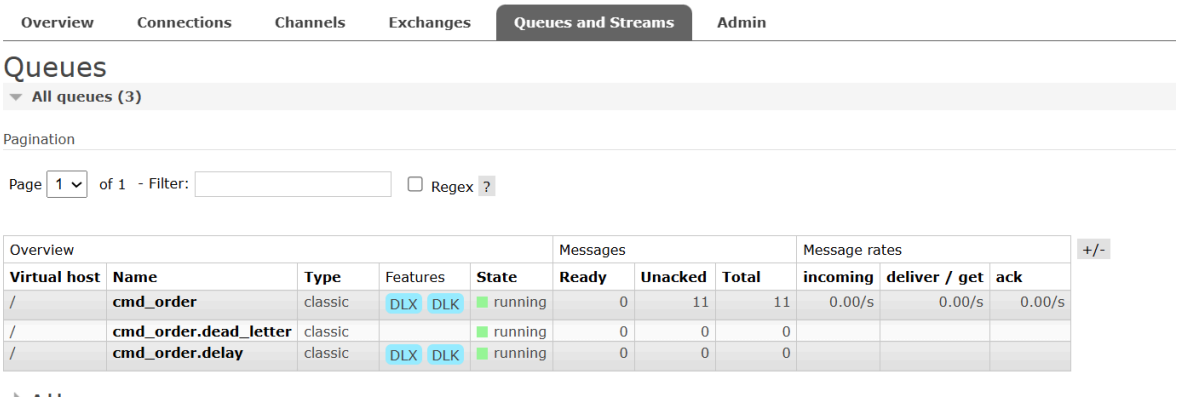


7. RABBITMQ панель управления

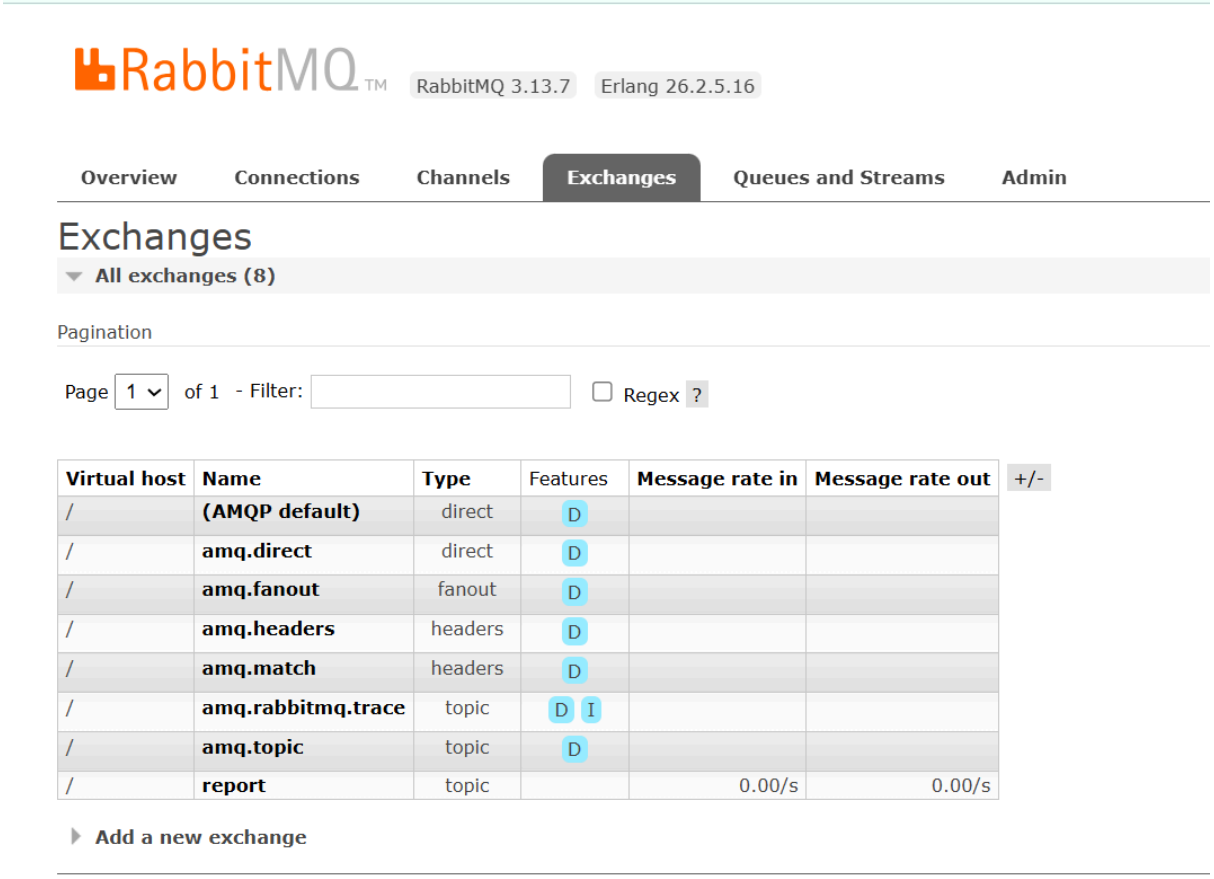
7.1 Overview



7.2 Queues



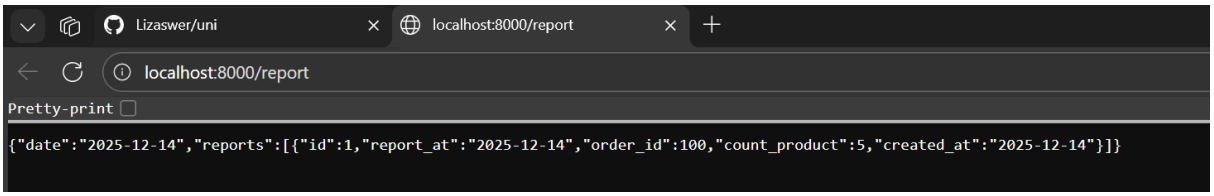
7.3 Exchanges



8. Тестирование REST API

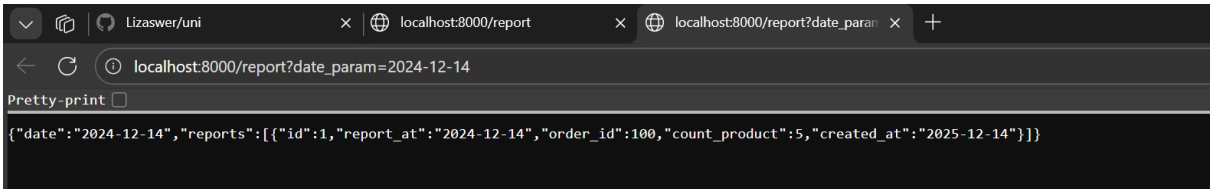
8.1 POST запрос

```
curl -X POST "http://localhost:8000/report" ^
  -H "Content-Type: application/json" ^
  -d "{\"report_at\": \"2024-12-14\", \"order_id\": 9999, \"count_product\": 99}"
```



8.2 GET запрос

```
curl "http://localhost:8000/report?date_param=2024-12-14"
```



9. Все компоненты в Docker

docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"

```
C:\Users\lyaho\OneDrive\Документы\GitHub\uni\lab8>docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"
NAMES             STATUS       PORTS
lab8-postgres-1   Up 2 hours   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
lab8-rabbitmq-1   Up 2 hours   0.0.0.0:5672->5672/tcp, [::]:5672->5672/tcp, 0.0.0.0:15672->15672/tcp, [::]:15672->15672/
tcp
```

Ответы на вопросы:

1. Что делает Брокер (Broker) в системе TaskIQ?

Брокер отвечает за передачу задач между клиентом и воркерами. Он определяет, куда отправлять задачи (например, в очередь RabbitMQ, Redis и т.д.) и обеспечивает асинхронную коммуникацию.

2. Что делает Планировщик (Scheduler)?

Планировщик отвечает за выполнение задач по расписанию (например, через cron). Он запускает задачи в указанное время и может работать с несколькими источниками расписаний.

3. Проблема с выполнением задачи каждые 5 секунд, если она выполняется 10 секунд

Это приведёт к накоплению задач, перегрузке очереди и возможному падению воркеров.
Решение:

- Увеличить интервал выполнения.
- Использовать блокировку (lock) или семафоры.
- Настроить масштабирование воркеров.

4. Cron-выражение для будних дней в 9:00 MSK

0 6 * * 1-5

Если сервер в UTC, то 9:00 MSK = 6:00 UTC. Нужно учитывать смещение через параметр cron_offset или настройку таймзоны сервера.

5. Масштабирование и отказоустойчивость

- Масштабирование: Запустить несколько воркеров, использовать балансировку очередей.
- Отказ воркера: Задачи останутся в очереди и будут обработаны другими воркерами.
- Отказ планировщика: Задачи по расписанию не будут запускаться. Решение: дублирование планировщика или использование внешнего планировщика (например, systemd/cron).

Вывод

В ходе лабораторной работы была успешно реализована распределённая система для выполнения периодических задач с использованием TaskIQ. Основные результаты:

1. Настроена инфраструктура: PostgreSQL для хранения данных, RabbitMQ в качестве брокера сообщений.
2. Реализован планировщик задач: TaskIQ Scheduler настроен на выполнение задачи generate_report каждую минуту (cron: */1 * * * *).
3. Созданы обработчики: TaskIQ Workers (2 процесса) асинхронно выполняют задачи из очереди.
4. Реализован REST API: FastAPI с эндпоинтами для управления отчётами (/report, /reports/all).

5. Обеспечена интеграция: Все компоненты связаны через RabbitMQ, данные сохраняются в PostgreSQL.

Система демонстрирует принципы работы с асинхронными задачами, планировщиками и очередями сообщений. Все компоненты работают корректно и взаимодействуют между собой.