

Worker

bot/worker.py — микросервис, который читает очередь и реализует бизнес-логику бота.

Шаги выполнения

1. создать класс-наследник *BotWorker* на основе написанного *Worker* в предыдущем уроке по RabbitMQ
2. переопределить *handler* — внутри него нужно вызвать метод *handle_update*, в котором будет реализована бизнес-логика бота
3. подключиться к PostgreSQL и реализовать необходимые запросы для бота
4. реализовать бизнес-логику бота

Как проверить руками

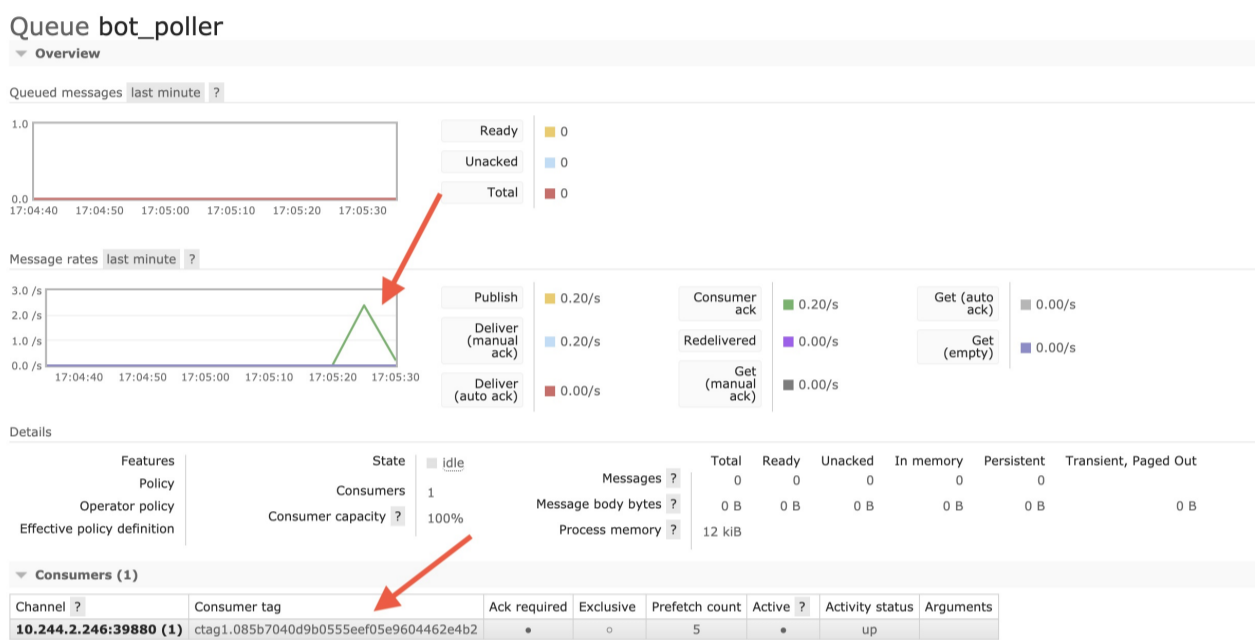
Запустить poller и worker в разных вкладках терминала:

```
export BOT_TOKEN=<token>
python3 run_bot.py poller
python3 run_bot.py worker
```

Далее:

- написать что-то своему боту в telegram
- *BotWorker.handler* должен начать вызываться при отправке сообщений боту
- в админке RabbitMQ в разделе queues в очереди "bot_poller" должна появиться информация о том, что задачи начали выполняться

Пример отображения корректной работы воркера:



В message rates должен появиться всплеск. Это означает, что задачи начали поступать в очередь. В consumers должен появиться процесс, которые получает события из очереди.

Как проверить тестами

```
pytest tests/bot/test_worker.py -vv -s
```