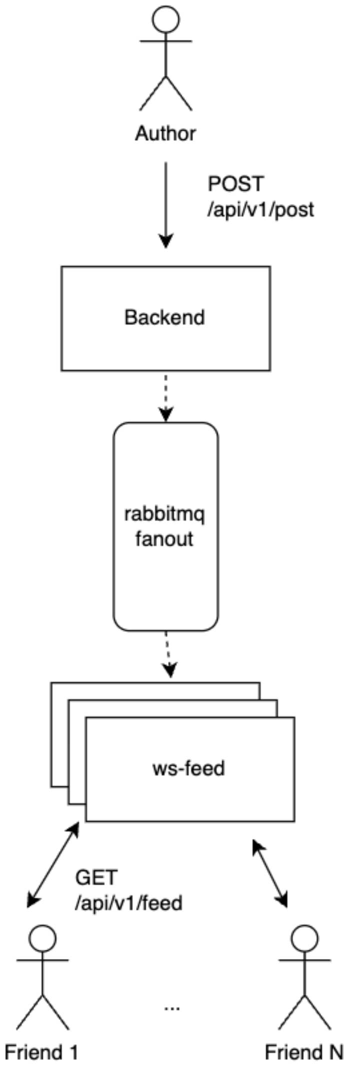
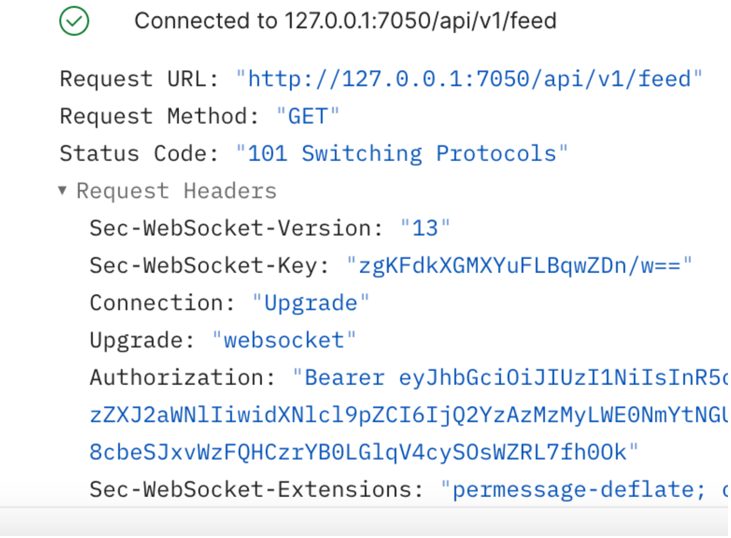
РЕАЛИЗАЦИЯ

Для реализации веб-сокетов напишем новый сервис, для удобного масштабирования, будем использовать тип exchange – fanout, для горизонтального масштабирования.

Справа схема взаимодействия.



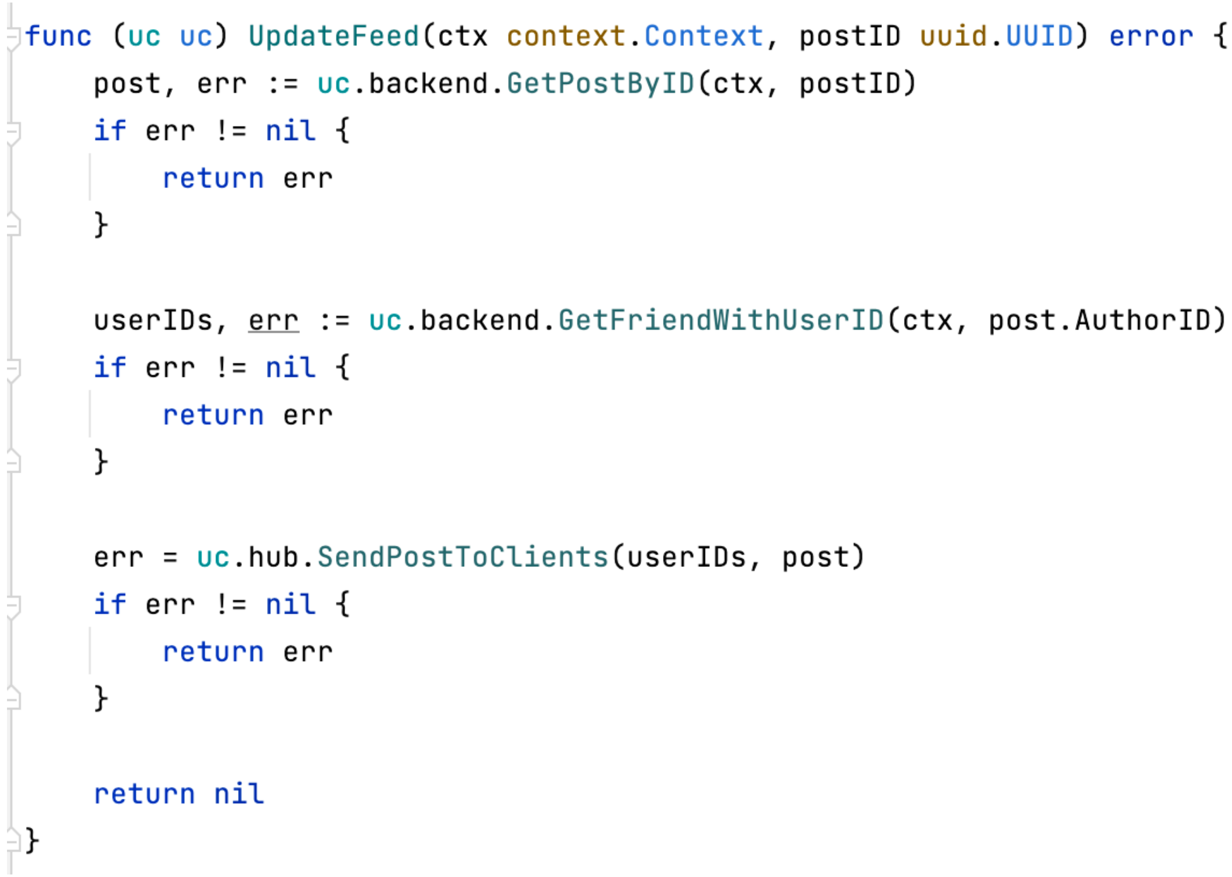
Запрос на создание канала происходит с помощью GET запроса. Там же прокидываем заголовок Authorization с jwt-токеном.



```
Connected to 127.0.0.1:7050/api/v1/feed

Request URL: "http://127.0.0.1:7050/api/v1/feed"
Request Method: "GET"
Status Code: "101 Switching Protocols"
▾ Request Headers
   Sec-WebSocket-Version: "13"
   Sec-WebSocket-Key: "zgKFdkXGMXYuFLBqwZDn/w=="
   Connection: "Upgrade"
   Upgrade: "websocket"
   Authorization: "Bearer eyJhbGciOiJIUzI1NiIsInR5c
   zZXJ2aWNlIiwidXNlcl9pZCI6IjQ2YzAzMzMyLWE0NmYtNG
   8cbeSJxvWzFQHCzrYB0LGlqV4cySOsWZRL7fh0Ok"
   Sec-WebSocket-Extensions: "permessage-deflate;
```

USECASE



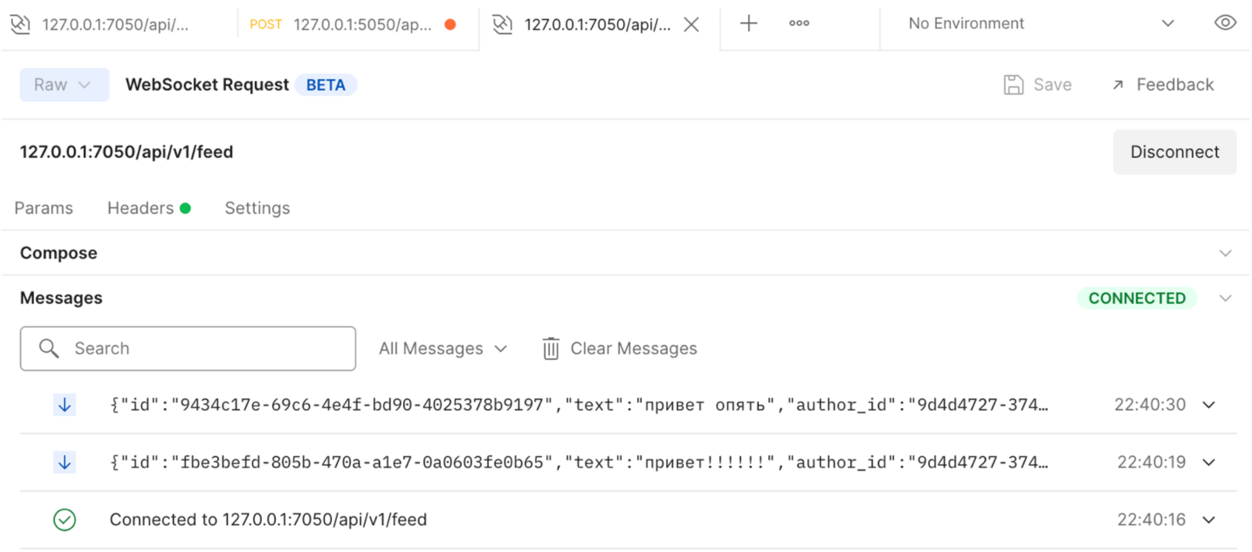
```go
func (uc uc) UpdateFeed(ctx context.Context, postID uuid.UUID) error {
	post, err := uc.backend.GetPostByID(ctx, postID)
	if err != nil {
		return err
	}

	userIDs, err := uc.backend.GetFriendWithUserID(ctx, post.AuthorID)
	if err != nil {
		return err
	}

	err = uc.hub.SendPostToClients(userIDs, post)
	if err != nil {
		return err
	}

	return nil
}
```

POSTMAN



127.0.0.1:7050/api/v1/feed

```
↓ {"id":"9434c17e-69c6-4e4f-bd90-4025378b9197","text":"привет опять","author_id":"9d4d4727-374…   22:40:30
↓ {"id":"fbe3befd-805b-470a-a1e7-0a0603fe0b65","text":"привет!!!!!!","author_id":"9d4d4727-374…   22:40:19
✓ Connected to 127.0.0.1:7050/api/v1/feed                                                          22:40:16
```

docker-compose

```yaml
version: "3.9"
services:
  db:
    image: postgres:14
    environment:
      POSTGRES_DB: "default-db"
      POSTGRES_USER: "someuser"
      POSTGRES_PASSWORD: "p@ssw0rD"
    ports:
      - "5432:5432"
    healthcheck:
      test: [ "CMD-SHELL", "pg_isready -U someuser -d default-db" ]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
  migrations:
    build:
      context: ./backend
      dockerfile: ./migrate.Dockerfile
    environment:
      - POSTGRES_HOST=db
      - POSTGRES_PORT=5432
      - POSTGRES_USER=someuser
      - POSTGRES_PASSWORD=p@ssw0rD
      - POSTGRES_DB=default-db
    depends_on:
      db:
        condition: service_healthy

  rabbitmq:
    image: rabbitmq:management
    ports:
      - 5672:5672
      - 15672:15672
    healthcheck:
      test: [ "CMD-SHELL", "rabbitmqctl status" ]
      interval: 5s
      timeout: 5s
      retries: 10

  backend:
    build:
      context: ./backend
      dockerfile: ./Dockerfile
    ports:
      - "5050:5050"
    environment:
      - POSTGRES_HOST=db
      - POSTGRES_PORT=5432
      - POSTGRES_USER=someuser
      - POSTGRES_PASSWORD=p@ssw0rD
      - POSTGRES_DB=default-db
      - LOG_LEVEL=debug
      - ADDR=5050
      - JWT_EXPIRATION=10000
      - JWT_KEY=kek
      - RABBITMQ_DSN=amqp://guest:guest@rabbitmq:5672/
      - RABBITMQ_QUEUE=worker
      - FEEDWORKER_HOST=http://feed-worker:6050
    depends_on:
      migrations:
        condition: service_completed_successfully
      rabbitmq:
        condition: service_healthy

  redis:
    image: redis:latest
    ports:
      - "6379:6379"
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 5s
      timeout: 30s
      retries: 50

  feed-worker:
    build:
      context: ./feed-worker
      dockerfile: ./Dockerfile
    ports:
      - "6050:6050"
    environment:
      - RABBITMQ_DSN=amqp://guest:guest@rabbitmq:5672/
      - RABBITMQ_QUEUE=worker
      - LOG_LEVEL=debug
      - ADDR=6050
      - BACKEND_HOST=http://backend:5050
      - REDIS_HOSTS=redis:6379
    depends_on:
      redis:
        condition: service_healthy
      rabbitmq:
        condition: service_healthy

  ws-feed:
    build:
      context: ./ws-feed
      dockerfile: ./Dockerfile
    ports:
      - "7050:7050"
    environment:
      - RABBITMQ_DSN=amqp://guest:guest@rabbitmq:5672/
      - LOG_LEVEL=debug
      - ADDR=7050
      - BACKEND_HOST=http://backend:5050
      - JWT_KEY=kek
    depends_on:
      rabbitmq:
        condition: service_healthy
      backend:
        condition: service_started
    restart: always
```