

Документация базы данных Reflect

Общая информация

База данных проекта развернута в **Docker-контейнере** с использованием PostgreSQL 13. Основные характеристики:

- Контейнер: `reflect_postgres`
- Порт: 5432 (внутри контейнера) / 5438 (проброшен на хост)
- Хост БД: `reflect_postgres` (сервис в docker-compose)

Защита от SQL-инъекций

Система обеспечивает защиту следующими способами:

1. **Django ORM** - все запросы используют параметризованные запросы
2. **Валидация входных данных** на уровне сериализаторов

Примеры CRUD-операций

1. Create (Создание)

```
# Создание нового пользователя
from api.authReflect.models import User

user = User.objects.create(
    username='new_user',
    email='user@example.com',
    password='securepassword123'
)
```

2. Read (Чтение)

```
# Получение пользователя
user = User.objects.get(id=1)

# Фильтрация
active_users = User.objects.filter(is_active=True)

# Получение эмоциональных состояний пользователя
from api.emotions.models import UserState
states = UserState.objects.filter(user=request.user)
```

3. Update (Обновление)

```
# Обновление одного поля
```

```
user = User.objects.get(id=1)
user.username = 'updated_username'
user.save()

# Массовое обновление
User.objects.filter(is_active=False).update(is_active=True)
```

4. Delete (Удаление)

```
# Удаление пользователя
user = User.objects.get(id=1)
user.delete()

# Массовое удаление
UserState.objects.filter(created_at__lt=timezone.now()-timedelta(days=365)).delete()
```

Подключение к БД извне

Параметры подключения:

- Хост: 185.185.71.233
- Порт: 5438
- Пользователь: your_user (указывается в .env)
- БД: your_db

Мониторинг

Для проверки состояния БД:

bash

Copy

Download

```
docker exec reflect_postgres psql -U your_user -c "SELECT now(), version();"
```

Подключенная база данных в DBeaver:

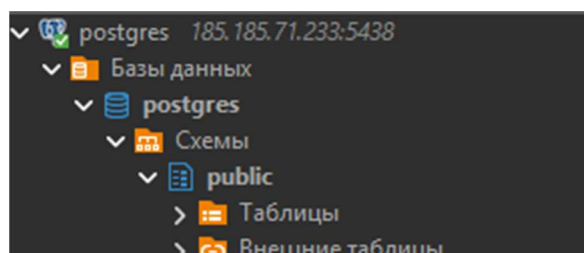


Диаграмма развёрнутой базы данных:

