

# Как написать расширение PostgreSQL?

Илья Евдокимов, «Тантор Лабс»

# \* Обо мне

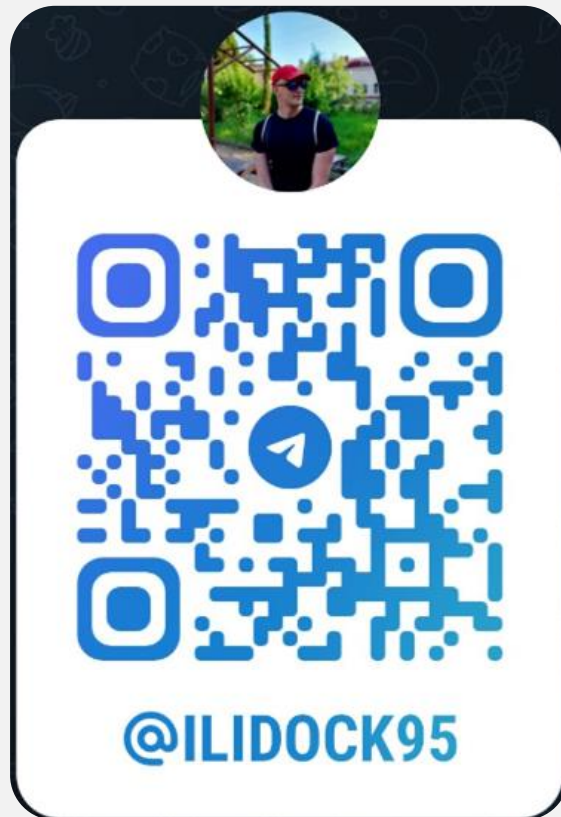


## Образование:

- Магистратура ВМК МГУ им. Ломоносова

## Опыт работы:

- Auriga (USB-hub 3.0)
- InfoTeCS (ПАК)
- Tantor(PostgreSQL)



# \* Основное достоинство PostgreSQL



**Расширяемость.**

**Плюсы расширений**

- Гибкость
- Удобное сопровождение (переносимость, апгрейд, ...)

**Что можно реализовать в расширениях?**

- Тип данных
- Функции и операторы по работе с данными
- Индексные и табличные методы доступа
- Интеграции с другими системами
- *Вообще произвольный функционал...*

## \* Из чего состоит расширение?

- Makefile # файл для сборки
- my\_extension.control # управляющий файл
- my\_extension--1.0.sql # скрипт после загрузки расширения
- my\_extension.c # исходник на C
- 
- # Регрессионные тесты (а для сложных сценариев TAP тесты)
- sql/my\_extension.sql # регрессионный тест
- data/my\_extension.data # данные для регрессионных тестов
- expected/my\_extension.out # ожидаемые результаты тестов
- 
- README # документация

## \* Системные составляющие расширений



my\_extension.control

```
comment = 'generate ID'
default_version = '1.0'
module_pathname = '$libdir/my_extension'
relocatable = true
```

my\_extension-1.0.sql

```
CREATE OR REPLACE FUNCTION generate_check_digit(text)
RETURNS smallint
AS 'MODULE_PATHNAME', 'generate_check_digit'
IMMUTABLE STRICT LANGUAGE C PARALLEL SAFE;
```

## \* Шаг 1: Makefile



Обязательные переменные

```
MODULE_big = my_extension      # my_extension.so
OBJS = my_extension.o
HEADERS = my_extension.h
EXTENSION = my_extension      # название расширения
DATA = my_extension-1.0.sql
```

## \* Шаг 2: Makefile



### Инфраструктура PGXS

```
ifdef USE_PGXS
PG_CONFIG = /usr/local/pgsql/bin/pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs )
include $(PGXS)
else
subdir = contrib/my_extension
top_builddir = ../..
include $(top_builddir)/src/Makefile.global
include $(top_srcdir)/contrib/contrib-global.mk
endif
```

## \* Шаг 3: Makefile



Если для регрессионных тестов необходимы дополнительные параметры в postgresql.conf

```
shared_reload_libraries = 'pg_example'
```

Добавим параметры в postgresql.conf, который будет использован для запуска тестов:

```
# регрессионные тесты
REGRESS = my_extension

# Параметры регрессионных тестов
REGRESS_OPTS = --temp-configure=my_extension.conf
```



## \* my\_extension.c



```
#include "postgres.h"
#include "fmgr.h"

PG_MODULE_MAGIC;
PG_FUNCTION_INFO_V1(generate_check_digit);

Datum
check_digit(PG_FUNCTION_ARGS)
{
    Text *name = PG_GETARG_TEXT_PP(0);
    uint64_t id = atol(name->vl_dat);
    uint8_t check_digit = 0;
    if(id > 99999999) /* 8 length only */
    {
        return -1;
    }

    return id;
}
```



## Компиляция, установка и управление расширениями



Если собираем через PGXS

```
USE_PGXS=1 make  
USE_PGXS=1 make install  
USE_PGXS=1 make installcheck
```

Если собираем в contrib/

```
make  
sudo make install  
make check
```

Включаем, выключаем, обновляем

```
CREATE EXTENSION [IF NOT EXISTS] my_extension;  
DROP EXTENSION [IF EXISTS] my_extension;  
ALTER EXTENSION my_extension UPDATE [TO version];
```

**Спасибо за внимание!**

