

docker-compose.yml

```
version: '3.9'

services:

  server:
    image: postgres/postgres:v8.0.0
    ports:
      - "3000:3000"
    links:
      - db:db
    environment:
      PGRST_DB_URI:
postgres://sample_user:sample_password@db:5432/sample_db
      PGRST_DB_SCHEMA: public
      PGRST_DB_ANON_ROLE: sample_user
      PGRST_SERVER_PROXY_URI: "http://127.0.0.1:3000"
    depends_on:
      - db

  db:
    image: postgres:13.3-alpine
    restart: always
    ports:
      - "9000:5432"
    environment:
      POSTGRES_DB: sample_db
      POSTGRES_USER: sample_user
      POSTGRES_PASSWORD: sample_password
    volumes:
      - "./pgdata:/var/lib/postgresql/data"
```

Заныч:

```
docker-compose up
```

Проверка работоспособности

Создадим и заполним таблицу:

```
user@domain:~$ docker-compose exec db psql -U sample_user -d sample_db
```

```
psql (13.3)
Type "help" for help.

sample_db=# create table if not exists sample_table (
sample_db(#      id serial primary key,
sample_db(#      value int not null default 0
sample_db(# );
CREATE TABLE
sample_db=# insert into sample_table (value)
sample_db-# values
sample_db-#      (10),
sample_db-#      (9),
sample_db-#      (8),
sample_db-#      (7),
sample_db-#      (6);
INSERT 0 5
```

Выполним выборку:

```
user@domain:~$ curl localhost:3000/sample_table
[{"id":1,"value":10},
 {"id":2,"value":9},
 {"id":3,"value":8},
 {"id":4,"value":7},
 {"id":5,"value":6}]
```

Добавим новую строку:

```
user@domain:~$ curl localhost:3000/sample_table -X POST \
> -H "Content-Type: application/json" \
> -d '{"value": 100500}'
```

Выполним выборку еще раз - строка успешно добавилась:

```
user@domain:~$ curl localhost:3000/sample_table
[{"id":1,"value":10},
 {"id":2,"value":9},
 {"id":3,"value":8},
 {"id":4,"value":7},
 {"id":5,"value":6},
 {"id":6,"value":100500}]
```