
Iks Documentation

Release latest

Jul 30, 2021

Contents

1	Project Install from repo for developing	3
2	Start in Docker	5
3	Start for developing locale with postgres, redis in docker	7
4	Git flow	9

- This is project, blog and shop with the most modern technologies, also for testing new version of python and django.
- This is just a pet project backend, this is not a boxed solution, this is just an API for my project.

CHAPTER 1

Project Install from repo for developing

Clone project:

```
git clone -b develop https://github.com/63phc/lks.git
```

- There are two ways to start a project, all in docker or only pg, redis in docker
- You can not use docker, then you should have pg and redis in local

CHAPTER 2

Start in Docker

- Install Docker: [instructions](<https://docs.docker.com/install/linux/docker-ce/ubuntu/#supported-storage-drivers>)
- edit `.docker/dev/.env` file with your params:

```
cp .env.example .docker/dev/.env
docker-compose -f .docker/docker-compose.yml build
docker-compose -f .docker/docker-compose.yml run backend python manage.py_
↳makemigrations
docker-compose -f .docker/docker-compose.yml run backend python manage.py migrate
docker-compose -f .docker/docker-compose.yml run backend python manage.py_
↳loaddata src/fixtures/*.json
docker-compose -f .docker/docker-compose.yml up
```

- Pycharm Setup: <https://www.jetbrains.com/help/pycharm/docker.html>

CHAPTER 3

Start for developing locale with postgres, redis in docker

- in file `.env`: need update `POSTGRES_HOST=localhost`:

```
docker-compose -f .docker/docker-compose.local.yml up postgresql redis
```

- Create virtual env:

```
python3 -m venv Venv
source Venv/bin/activate
pip3 install -r src/requirements/development.txt
```

- Or through pipenv:

```
pip3 install pipenv
pipenv install
pipenv shell
```

- Env File

-
- edit `.env.example` file with your params:

```
cp .env.example .env
```

- Prepare project:

```
python manage.py makemigration
python manage.py migrate
python manage.py createsuperuser
python manage.py loaddata src/fixtures/*.json
python manage.py runserver
```


CHAPTER 4

Git flow

- Easy git flow:

```
git checkout develop
git pull develop
git checkout -b <your branch>
# when complete task
git add .
git commit -m '#<number task> commit messages'
git push origin <your branch>
```

- Git flow healthy person <https://danielkummer.github.io/git-flow-cheatsheet/>
- Settings flake + pre-commit hook:

```
sudo pip3 install flake8
#(OUTPUT FILTERS -> $FILE_PATH$\: $LINE$\: $COLUMN$\: .* )
flake8 --install-hook git
git config --global --bool flake8.strict true
```

- Easy start -> ctrl + shift + a -> flake -> enter