Session 1:
Preparing the workspace

Xander Warszawski

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

0

Contents of this session

1. Introduction
2. Installing tools
3. Git and GitHub
4. Django setup
5. Docker setup

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

1

1. Introduction

2

Goals

- Be able to **create and maintain** a **Django webapp in production**
  - With **integrated payments** by **Mollie**
  - With **error monitoring** by **Sentry**
- Be able to **use Git** and **GitHub**
- Be able to **deploy** the webapp **using DigitalOcean**

3

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

3

Who am I?

- Xander Warszawski
- Master student Engineering Technology ICT
- IEEE SB KU Leuven Campus Bruges Chair
- GitHub Campus Expert

- Pythonista since +/- 2017
- Django for side projects
- Last year: worked on C# Web API
- For any questions: **xander@xdoubleu.com**

4

GitHub IEEE Student Branch KU Leuven Campus Brugge

4

Slides

- Can be fetched from https://xdoubleu.com/webdev-4-dummies-workshop/
- I can also provide printed slides on request
- Code is also available here

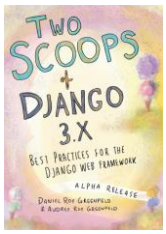
5

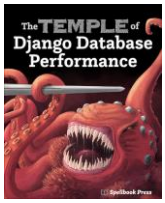
GitHub IEEE Student Branch KU Leuven Campus Brugge

5

Slides

- Slides are based on:
  - '**Django for Professionals 4.0**' by William S. Vincent
  - '**Speed Up Your Django Tests**' by Adam Johnson
  - '**Two Scoops of Django**' by Audrey R. Greenfeld and Danny R. Greenfeld
  - '**The Temple of Django Database Performance**' by Andrew Brookins





6

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

6

Scope of this workshop series

- Web applications / web apps:
  - App stored on remote server
  - Delivered over internet through browser interface
- Front-end / client-side:
  - HTML
  - CSS
  - JS
- Back-end / server-side:
  - In this case: Python
  - Other options:
    - JS/TS
    - Ruby
    - PHP
- I'll try to cover both, but back-end is my 'expertise'

7

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

7

What is Django?

- From the Django docs:
  - ***Django** is a **high-level Python web framework** that encourages **rapid development** and **clean, pragmatic design**. Built by experienced developers, it **takes care of much of the hassle of web development**, so **you can focus on writing your app without needing to reinvent the wheel**. It's **free and open source**.*
- Used by:
  - Instagram
  - Spotify
  - Pinterest
  - Bitbucket

GitHub  IEEE Student Branch KU Leuven Campus Brugge

Why use Django?

- Features (from Django docs):
  - **Ridiculously fast**: devs can take applications from concept to completion very fast
  - **Fully loaded**: handles lots of overhead out of the box (e.g. user authentication)
  - **Reassuringly secure**: lots of security features out of the box
  - **Exceedingly scalable**: can flexibly scale to meet the heaviest traffic demands
  - **Incredibly versatile**: lots of different use cases and uses

GitHub  IEEE Student Branch KU Leuven Campus Brugge

Python cheat sheet

- Django uses specific code that doesn't contain a lot of logic
  - Basic Python knowledge is sufficient
  - https://www.pythoncheatsheet.org/
- If you want to practice programming (in Python):
  - https://www.hackerrank.com/
  - https://leetcode.com/

GitHub IEEE Student Branch KU Leuven Campus Brugge

Timing

- 05/10: Introduction + setup
- 12/10: Basics of Django
- 19/10: Varia
- 26/10: Preparing for Deployment
- 09/11: Deployment
- 16/11: Practice everything on your own

GitHub IEEE Student Branch KU Leuven Campus Brugge

2. Installing tools

12

Python install

- https://www.python.org/downloads/
- Install now:
  - For all users
  - Add to PATH

GitHub IEEE Student Branch KU Leuven Campus Brugge

13

VSCode install

- https://code.visualstudio.com/
- Recommended extensions:
  - Bookmarks
  - Docker
  - GitLens
  - IntelliCode
  - Python
  - Django
  - SQLTools
    - + driver for PostGreSQL

GitHub IEEE Student Branch KU Leuven Campus Brugge

Git install

- https://git-scm.com/download
- Default everything

GitHub IEEE Student Branch KU Leuven Campus Brugge

Sourcetree install

- https://www.sourcetreeapp.com/
- Skip registration
- Install git
- SSH Key: No

GitHub IEEE Student Branch KU Leuven Campus Brugge

Docker install

- https://www.docker.com/

GitHub IEEE Student Branch KU Leuven Campus Brugge

3. Git and GitHub

18

Introduction

- **Git** is a **VCS** (**V**ersion **C**ontrol **S**ystem)
  - Allows developers to store all versions of their code as one (= repository)
  - *Branching* and *merging*
- **GitHub** is a **Hub**
  - Hub? **A hub** airport is an airport used by one or more airlines **to concentrate** passenger **traffic and** flight **operations**.
  - So? GitHub is a **concentration of Git repositories**
    - Also publishing & collaboration tool

19

GitHub IEEE Student Branch KU Leuven Campus Brugge

19

Terminology

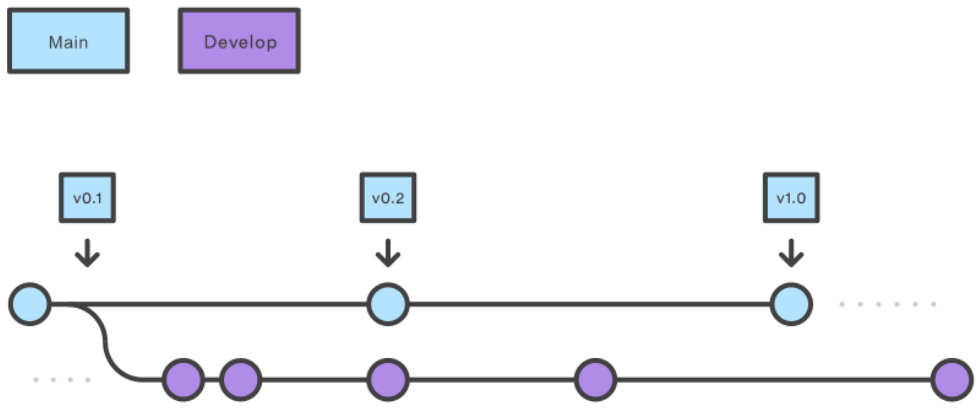
- **Repo**(sitory): project 'folder' (code, docs); has all versions
- **Branch**: parallel version in repo
- **Commit**: saving changes in branch
- **Fork**: 'copy' existing repo
- **Merge**: merge branches or repo
- **Pull Request** (GH) / Merge Request (GL): request to merge branch or fork
- **Clone**: copy of repo, mostly offline
- **Fetch**: getting last version of online repo
- **Pull**: 'download' latest commits from remote repo to clone
- **Push**: 'upload' committed changes to remote repo
- **Reverse**: undo changes made in commit(s), useful for when application breaks in production
- **Cherry pick**: pick a certain commit out of the change history

20

GitHub IEEE Student Branch KU Leuven Campus Brugge

20

GitFlow Workflow 1: main & dev

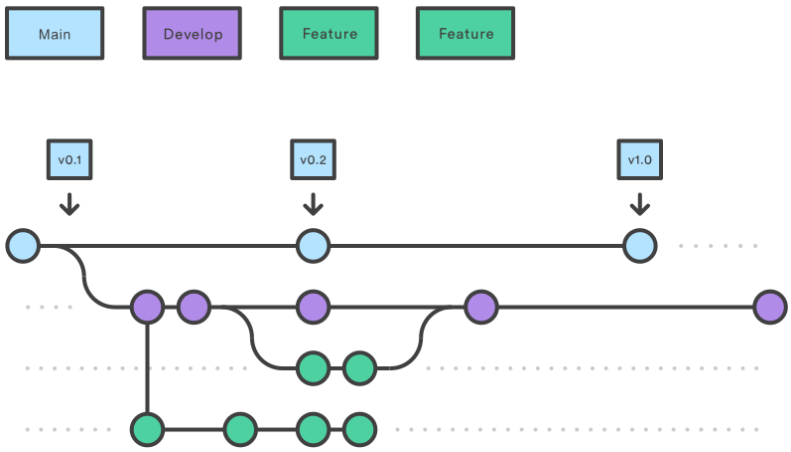


21

GitHub IEEE Student Branch KU Leuven Campus Brugge

21

GitFlow Workflow 2: feature



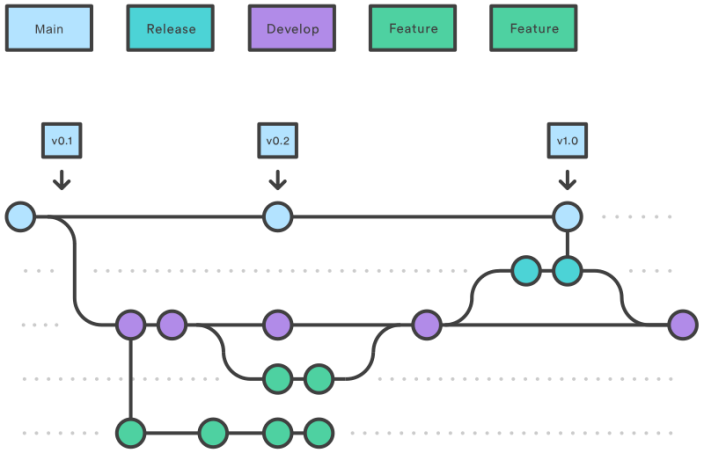
22

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

22

GitFlow Workflow 3: release



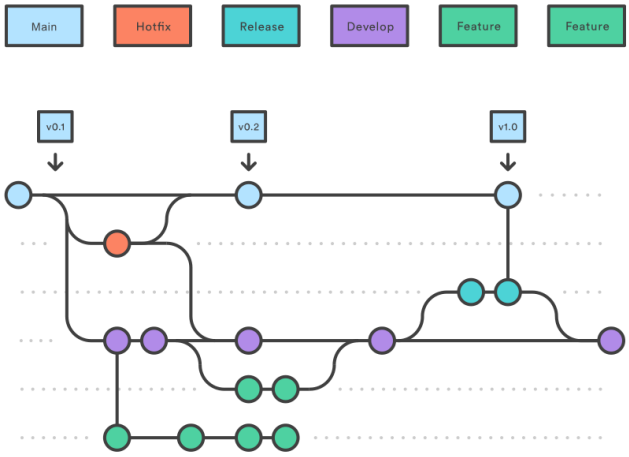
23

GitHub

IEEE Student Branch
KU Leuven Campus Brugge

23

GitFlow Workflow 4: hotfix



GitHub IEEE Student Branch KU Leuven Campus Brugge

GitFlow Workflow: overall flow

1. **Dev** branch from **main**
2. **Release** branch from **dev**
3. **Feature** branches from **dev**
4. **Feature** done? Merge in **dev**
5. **Release** done? Merge in **main** & **dev**
6. Issue in main? Create **hotfix** branch from **main**
7. **Hotfix** done? Merge in **main** & **dev**
8. Source: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

GitHub IEEE Student Branch KU Leuven Campus Brugge

Questions?

- If you want to practice git:
  - https://learngitbranching.js.org/

GitHub IEEE Student Branch KU Leuven Campus Brugge

Practice

1. Create GitHub account: https://github.com/signup
2. Create repo:
   1. '+'-icon right upper corner > new repo;
   2. public or private up to you
   3. no readme
   4. no gitignore
3. Automatically delete head branches (Settings > General > Pull Requests)
4. Go to actions:
   1. Search for 'Django', click configure
   2. Remove run on push, add 'dev' to on PR branches
   3. Make sure python versions are: 3.8, 3.9, 3.10
   4. Start commit

GitHub IEEE Student Branch KU Leuven Campus Brugge

Practice

1. Setup dev branch:
    1. Click main > Type dev in “find or create a branch” > create dev branch from main
    2. Settings > branches
    3. Add branch protection rule for main and dev
        1. Require PR
        2. Disable require approvals
2. Clone using SourceTree
3. Add file:
    1. Checkout dev branch from remotes
    2. Create feature branch from dev
    3. Add file, commit file
    4. Create PR to dev
4. Django action will fail, merge anyways

GitHub IEEE Student Branch KU Leuven Campus Brugge

4. Django setup

Git checkpoint

1. Create new feature branch

GitHub IEEE Student Branch KU Leuven Campus Brugge

Django installation

1. Make sure you have a GitHub repo on your pc and you're on a feature branch
2. Open the folder in VSCode, open a new terminal
3. **python –m venv .venv**
4. **.venv\Scripts\activate**
5. **pip install django**
6. Optional: **pip install --upgrade pip**
7. **pip freeze > requirements.txt**
8. **django-admin startproject config . (!! Don't forget dot at the end)**
9. **python manage.py migrate**
10. **python manage.py runserver**

GitHub IEEE Student Branch KU Leuven Campus Brugge

Git checkpoint

1. Add gitignore (https://www.toptal.com/developers/gitignore)
2. Commit changes + do a PR
3. Add require checks to branch protection
    1. ''build 3.10'' by example

GitHub IEEE Student Branch KU Leuven Campus Brugge

Questions?

GitHub IEEE Student Branch KU Leuven Campus Brugge

5. Docker setup

34

Introduction

- Docker:
  - Type of virtualization that only uses Linux containers
    - Containers are created from images
  - Everything above the OS is virtualized
- Virtual environments:
  - Can only isolate Python packages
- We will use Docker to facilitate local runs only, not in deployment

35

GitHub IEEE Student Branch KU Leuven Campus Brugge

35

Git checkpoint

1. Create new feature branch

GitHub IEEE Student Branch KU Leuven Campus Brugge

Add Docker

1. Create a **Dockerfile** next to **manage.py**

GitHub IEEE Student Branch KU Leuven Campus Brugge

Dockerfile (1)

```
# Pull base image
FROM python:3.10.4-slim-bullseye

# Set environment variables
ENV PIP_DISABLE_PIP_VERSION_CHECK 1
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1
```

38

GitHub IEEE Student Branch KU Leuven Campus Brugge

38

Dockerfile (2)

```
# Set work directory
WORKDIR /code

# Install dependencies
COPY ./requirements.txt .
RUN pip install -r requirements.txt

# Copy project
COPY . .
```

39

GitHub IEEE Student Branch KU Leuven Campus Brugge

39

Add Docker

1. Create a **Dockerfile** next to **manage.py**
2. Add **.dockerignore** (similar to .gitignore)

GitHub IEEE Student Branch KU Leuven Campus Brugge

.dockerignore

```
.venv
.git
.gitignore
```


GitHub IEEE Student Branch KU Leuven Campus Brugge

Add Docker

1. Create a **Dockerfile** next to **manage.py**
2. Add **.dockerignore** (similar to .gitignore)
3. Add **docker-compose.yml**

GitHub IEEE Student Branch KU Leuven Campus Brugge

docker-compose.yml

```
version: "3.9"

services:
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
```


GitHub IEEE Student Branch KU Leuven Campus Brugge

Add Docker

1. Create a **Dockerfile** next to **manage.py**
2. Add **.dockerignore** (similar to .gitignore)
3. Add **docker-compose.yml**
4. Execute: **docker-compose up -d --build**
5. To stop**:**
    1. **docker-compose down**

GitHub IEEE Student Branch KU Leuven Campus Brugge

Database

- Django has built-in support for five databases:
    - PostgreSQL
    - MariaDB / MySQL
    - Oracle
    - SQLite
- Code is the same for each one, Django ORM & drivers handle the differences

GitHub IEEE Student Branch KU Leuven Campus Brugge

Add database

1. Pip install **psycopg2-binary** & freeze
2. Update **docker-compose.yml**

GitHub IEEE Student Branch KU Leuven Campus Brugge

docker-compose.yml (1)

```
version: "3.9"

services:
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
    depends_on:
      - db
```


GitHub IEEE Student Branch KU Leuven Campus Brugge

docker-compose.yml (2)

```
  db:
    image: postgres:13
    volumes:
      - postgres_data:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    environment:
      - "POSTGRES_HOST_AUTH_METHOD=trust"

volumes: #level of services
  postgres_data:
```


GitHub IEEE Student Branch KU Leuven Campus Brugge

Add database

1. Pip install **psycopg2-binary** & freeze
2. Update **docker-compose.yml**
3. Update **config/settings.py**

GitHub IEEE Student Branch KU Leuven Campus Brugge

config/settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'postgres',
        'USER': 'postgres',
        'PASSWORD': 'postgres',
        'HOST': 'db', # set in docker-compose.yml
        'PORT': 5432 # default port
    }
}
```


GitHub IEEE Student Branch KU Leuven Campus Brugge

Spin up containers

docker-compose up -d --build:

docker-compose exec web python manage.py migrate

docker-compose exec web python manage.py createsuperuser

docker-compose down

GitHub IEEE Student Branch KU Leuven Campus Brugge

Git checkpoint

1. Commit changes
2. Create PR

GitHub IEEE Student Branch KU Leuven Campus Brugge

Questions?

GitHub IEEE Student Branch KU Leuven Campus Brugge