Drups.io

Release 0.1

Drups.io

Apr 19, 2021

CONTENTS

CHAPTER ONE

LINKS

Documentation

- Drups.io Engine
- Drups.io Application
- Drups.io Docker

Repositories

- Drups.io Engine
- Drups.io Application
- Drups.io Docker

CHAPTER

TWO

CONTENTS

2.1 Whitepaper

2.1.1 Overview

Product's working name is Drups.io.

Drups.io is meant to be the open source project. It will be the collection of software pieces (front-end applications, back-end services and necessary devops scripts) which together assemble the build & deploy automation service for Drupal projects (similar to Netlify build, Vercel and others).

This document describes (1) the problem we face with build & deploy automation when developing drupal projects and (2) in-depth architectural and structural design of the solution along with picked programming languages, necessary tools, libraries and frameworks for implementation.

2.1.2 Goal and Challenge

Our goal is to create an instrument that improves the drupal project development process. Initially, Drups' purpose is to be the service that listens to every pull request and sets up respective installations with a unique subdomain, where the content and files will be maintained from the general branch. Central idea is that testers and developers can have the exact same instance, full with content for every pull request - even with tiny change.

Unlike most FE applications (that modern tools are working with), in Drupal cases we have different challenges. It is not possible to maintain application state only from the source code. In order to test pull requests properly, testers and developers need content and files, which is not part of the source code and hence can not be pulled automatically. Drups solves this issue and gives development teams necessary tools to control and manage initial databases and files for the project along with unique branches.

Drups is the tool (and service) makes it easy and comfortable for developers and testers to work on a Drupal project. It should not be thought of as a production environment for drupal projects.

2.2 Release Process

2.2.1 Branches

1. `main` branch for production
2. `develop` branch for active development
3. `feature/[feature-name]` branches for new features

4. `hotfix/[issue-name]` branches for hotfixes

2.2.2 Release

1. Create release branch from `develop` prefixed by `release/[version]`, for example: `release/1.0.0`
2. Bump version based on release
3. Generate a changelog file
4. Commit all change to release branch with `tag` on it
5. Create a pull request from the release branch to `master`: `release/1.0.0` -> `master`

2.2.3 HotFixing

Case 1: hotfix needed on `master` branch

1. Crete hotfix branch from `master` prefixed by `hotfix/[issu-name]`, for example: `hotfix/1-fix-user-creation`
2. Fix the issue
3. Bump patch version, so if the version is `1.0.0` it will be bumped as `1.0.1` and commit
4. Commit all change to the hotfix branch
5. Generate changelog file
6. Create a pull request from the `hotfix` branch to `master`: `hotfix/1-fix-user-creation` -> `master`
7. Create a pull request from the `hotfix` branch to `develop`: `hotfix/1-fix-user-creation` -> `develop`

Case 2: hotfix needed on `release/[version]` branch, for example: `release/1.0.0`

1. Pull the related `release` branch: `release/1.0.0`
2. Fix the issue and commit changes
3. Generate changelog file
4. Push changes
5. Create a pull request from the `release` branch to `develop`: `release/1.0.0` -> `develop`