



TP Déploiement d'une documentation utilisateur

Mise en place d'un déploiement automatisé via Gitlab

GitLab possède un outil de CI/CD qui permet un déploiement facile de MkDocs et Material avec GitLab pages et les Jobs. Un tel service permet de créer une documentation pour les utilisateurs d'une entreprise ou d'un logiciel.

Server

MkDocs

La documentation sera installée sur un serveur (par exemple Debian) sur lequel devront être présents : python, pip (un installateur de packages pour python) et le dossier dans lequel se trouvera le projet. A l'intérieur du dossier vous pourrez entrer la commande `pip install mkdocs` ce qui téléchargera et créera un fichier `mkdocs.yml` et un dossier "docs" dans lequel vous trouverez le fichier "index.md" qui vous aiguillera sur les ressources et la [documentation de MkDocs](https://www.mkdocs.org/) (attention cette documentation n'est qu'en anglais).

Le fichier `mkdocs.yml` vous permet de modifier le site à votre convenance. Vous trouverez la documentation de la configuration de ce fichier [ici](https://www.mkdocs.org/user-guide/configuration/).

Material

Material est un gestionnaire de thème pour MkDocs qui fonctionne en parallèle de celui-ci. Pour l'installer il vous faut exécuter la commande `pip install mkdocs-material` dans le dossier de votre documentation.

Material vous permet d'ajouter des fonctionnalités absentes de MkDocs à votre documentation. Vous trouverez les informations nécessaires à son utilisation [sur leur site](https://squidfunk.github.io/mkdocs-material/getting-started/).

Vous lancer le serveur à tout moment avec la commande `mkdocs serve` exécutée dans la racine de votre projet (le dossier du projet).

Git

Pour que ce projet soit visible et accessible aux utilisateurs, il est nécessaire de créer un dépôt (ici sur [Gitlab](https://about.gitlab.com/)) accessible en ligne ou pour l'organisation.

Initialisation

Créer un dépôt sur votre server gitlab sur lequel faire le versioning et le lier à la racine avec les commande :

```
```bash
```

```
git init
```

```
```
```

puis :

```
```bash
```

```
git add remote origin <url-dépôt>
```

```
```
```

et enfin les commandes

```
```bash
```

```
git add .
```

```
git commit -m "Initial commit"
```

```
git push origin main/master (selon votre branche principale)
```

```
```
```

Le dépôt sera présent avec le fichier mkdocs.yml qui permet de gérer les extensions et les thèmes, ainsi que le dossier docs dans lequel figurent les fichiers markdown.

Pages

Vous pouvez rendre accessible en ligne la documentation avec le CI de GitLab.

Pour celà il faut créer le fichier .gitlab-ci.yml à la racine du projet et y ajouter le contenu suivant :

```
pages:
```

```
  stage: deploy
```

```
  image: python:latest
```

```
  script:
```

```
    - pip install mkdocs-material
```

```
    - mkdocs build --site-dir public
```

Arthur Richert--Villain

cache:

key: \${CI_COMMIT_REF_SLUG}

paths:

- ~/.cache/

artifacts:

paths:

- public

rules:

- if: '\$CI_COMMIT_BRANCH == \$CI_DEFAULT_BRANCH'

Un commit vers la branche main de votre projet créera un Jobs et le CI vérifiera les erreurs et possibles conflits puis le CD déploiera une page avec Mkdocs et Material à l'adresse `<utilisateur>.gitlab.io/<projet>`.

A présent il est possible de collaborer sur la documentation en local et de "push" les modifications qui seront intégrées automatiquement à GitLab Pages et visible après un nouveau Job.