

Objectif

L'objectif de ce projet est de mettre en place une solution permettant d'afficher sur une page un *roster* de différents personnages de **Dofus**, avec plusieurs informations les concernant.

Informations générales

Ce projet sera à compléter en **3 versions** de difficultés croissantes.

Voici le contenu de chaque version :

1. 2 services : backend et frontend.
2. 3 services : base de données (Postgres), backend et frontend.
3. 4 services : Redis, base de données (Postgres), backend et frontend.

Pour chaque version, un dépôt GitHub vous sera fourni avec le code de chaque service. Vous devrez fournir les Dockerfiles de chaque service, ainsi que le fichier `docker-compose.yml`.

En complément, vous devrez mettre en place un workflow GitHub permettant de construire et pousser les images sur Docker Hub, ainsi que les manifestes Kubernetes permettant de déployer la stack pour chaque version.

Consignes

Le projet doit être réalisé en **binôme ou trinôme** (maximum).

La durée de réalisation est de **4 heures**.

Vous avez le droit de consulter toutes les ressources à votre disposition pendant le projet.

Pour délivrer vous utiliserez Github Assignment, sur l'adresse suivante :

<https://classroom.github.com/a/-aXGTY2H>

Livrables attendus

- Pour la version 1 : 2 Dockerfile, 1 docker-compose, 1 workflow Github, 2 deployments, 2 services
- Pour la version 2 : 3 Dockerfile, 1 docker-compose, 1 workflow Github, 3 deployments, 3 services
- Pour la version 3 : 4 Dockerfile, 1 docker-compose, 1 workflow Github, 4 deployments, 4 services

Sujet

Dépôt

URL : <https://github.com/sevrius/dofus>

Version 1

Dans cette version, les personnages Dofus sont inscrits en dur dans le service backend.
Le frontend est uniquement responsable d'afficher les données récupérées depuis le backend.

Branche : v1

Version 2

Dans cette version, les personnages Dofus sont stockés dans une base de données Postgres ajoutée à l'architecture. Le backend récupère désormais les données depuis cette base de données, tandis que le frontend reste inchangé.

Branche : v2

Version 3

Dans cette version, nous introduisons une notion de popularité pour chaque personnage. La popularité est stockée dans Redis, et son initialisation est réalisée par un script supplémentaire, [init_popularity.py](#). Le backend regroupe les données provenant des deux sources (Postgres et Redis), et le frontend affiche désormais une colonne supplémentaire pour la popularité.

Branche : v3