

SAÉ 1.03

Installation d'un poste de développement

Phase B - Chaîne de traitement

Introduction

Note : toutes les images (sauf **html2pdf**) contiennent **PHP** en version **8.3 CLI**¹. Si besoin, vous pouvez en faire usage. **Vous ne devez pas utiliser le PHP installé sur vos postes de travail.** L'idée est qu'à l'exception de Docker, votre chaîne de traitement ne doit pas nécessiter la présence d'un outil particulier (tel que PHP par exemple) pour exécuter la chaîne. Tout est packagé dans des conteneurs. Les filtres Unix de base et le Bash de votre machine locale sont acceptés dans votre chaîne.

Etape B-1 - La chaîne

Conseils

Vous devez réaliser la chaîne de traitement automatisé de tous les fichiers que vous recevez. Voici quelques conseils pour vous aiguiller sur la méthode :

- Vos scripts sont à écrire en Bash ou en PHP
- A l'exception du script qui déroule tout l'enchaînement et qui doit être exécuté sur votre machine locale, tous vos autres scripts doivent être exécutés dans des conteneurs dont on vous a fourni les images soit en Phase A soit ci-après dans ce document.
- L'option **--rm** d'un **docker run** vous permettra d'avoir des conteneurs éphémères qui ne laissent pas de traces derrière eux.
- L'option **-d** d'un **docker run** vous permet de laisser tourner un conteneur en arrière plan et de pouvoir continuer l'exécution de votre chaîne de traitement.
- L'option **--name** d'un **docker run** permet de donner un nom à un conteneur, ce qui est plus pratique pour le manipuler que d'utiliser son ID

¹ CLI : Command Line Interface, (Ligne de commande, dans le Terminal)

- Pour envoyer vos fichiers de travail et vos scripts de transformation dans un conteneur ainsi que pour récupérer le résultat produit depuis ce conteneur, vous avez la commande **docker container cp**

Surprise !

Vous allez donc créer la chaîne de traitement automatisées des fichiers, mais vous venez juste de recevoir un nouveau lot de fichiers dans lequel l'un d'entre eux a évolué (sur Moodle). C'est la dure vie de développeur·euse ! Voici les changements :

- Le tableau des sites touristiques comporte désormais un titre mais l'équipe de développement du site Web n'en a pas besoin. Vous devez l'ôter de manière automatique, ainsi que les en-têtes des colonnes.
- Certains départements sont absents, vous devez quand même les faire apparaître dans la liste/table en laissant vide le nom du site ainsi que le nombre annuel de visiteurs.
- Il faut trier les sites par ordre décroissant de numéro de département.

Ce n'est pas tout, un client qui se respecte a toujours des bonnes idées et des besoins qui évoluent !

Il a désormais besoin d'afficher aussi le nom du département qui à placer entre le code du département et le nom du site touristique.

Vous avez à votre disposition un fichier **DEPTS** qui liste tous les départements ordonnés par code de département.

Etape B-2 - Le tableau des sites en PDF

Le client veut mettre en ligne sur son site, 2 PDFs d'une seule page chacun affichant le tableau des sites touristiques avec les noms des départements, et triés l'un (**sites-dept.pdf**) par numéro de département et l'autre (**sites-visites.pdf**) par nombre décroissant de visiteurs annuels.

Pour le **sites-visites.pdf**, il faudra départager les départements ayant le même nombre de visiteurs en vous utilisant sur le code département (ordre croissant) comme second critère de tri.

Il veut aussi mettre en ligne un troisième PDF (**sites-regions.pdf**) synthétisant les départements et les visites par région. Le tableau doit donc présenter les régions (nom) et le total des visiteurs (somme des visiteurs de chaque site des départements de la région), par ordre décroissant du nombre de visiteurs global de chaque région.

Vous avez à votre disposition un fichier **REGIONS** qui associe un nom de région et les codes des départements qui la constituent.

Chacun de ces tableaux doit tenir sur une unique page A4 et, pour rester lisible, doit donc être dessiné sur plusieurs colonnes si nécessaire. Prévoir un titre et l'affichage du logo des Offices de Tourisme de France sur chacun d'eux.

Comme au TP 2 sur Docker, votre script doit générer les fichiers PDF à partir d'un HTML et de fichiers compagnons (les images). Mais avant de pouvoir en arriver là, vous devez générer le HTML et injecter le tout dans un conteneur de conversion HTML ➔ PDF.

Vous avez l'image Docker **sae103-html2pdf** (cf sujet Partie 1) qui doit vous servir à générer ces PDFs. Ce n'est pas le même que celui du TP2 mais son utilisation est assez proche.

La commande principale de cette image est **weasyprint**

-  **Rendu** : une archive nommée **etape_b2-<nom_équipe>.zip** qui doit être la compression d'un dossier **etape_b2/** contenant ce qui est décrit dans les **Attendus**.
-  **Délai** : la date qu'on fixera ensemble pour le rendu final
-  **Dépôt** : Moodle
-  **Attendus** : Le ou les scripts de génération des 3 PDFs et de lancement de leur génération, ainsi qu'un exemple de chacun des 3 PDFs produits par les scripts.

Il vous faudra aussi remplir et déposer un fichier **PARTICIPATION.md**