
SAÉ 1.03

Installation d'un poste de développement

Inscription de l'équipe

Avant tout, veuillez constituer vos équipes en remplissant la Google Sheet :

<https://docs.google.com/spreadsheets/d/1-mBHUXe25AaWpvrhm7a-B2lxbR69WTFwTuDs2g2GY2M/edit?usp=sharing>

Contexte

Vous travaillez dans une ESN¹ qui a été choisie par le Comité Olympique pour réaliser un site Web sur les JO de Paris 2024.

Une équipe de votre société est déjà en cours de réalisation de ce site Web.

Cette équipe reçoit régulièrement des fichiers (textes et images) de différentes sources pour alimenter ce site Web. Les fichiers reçus ne sont pas exploitables directement car les formats sont variés et ils ne sont pas toujours adaptés au domaine du Web

En outre, il arrive fréquemment que des fichiers reçus et traités par l'équipe de projet du site Web soient à nouveau reçus dans une ou plusieurs versions corrigées par leur source et qu'il faille refaire ce traitement d'adaptation et de nettoyage des données. L'équipe a d'abord effectué ce travail manuellement mais devant la récurrence et la fréquence de ces renvois, le chef de projet a décidé d'automatiser autant que possible ces tâches répétitives.

L'équipe du projet Web JO Paris 2024 n'ayant pas les ressources nécessaires pour s'en occuper, votre équipe d'experts est sollicitée pour produire cette chaîne de traitement automatisée.

¹ Entreprise de **S**ervice du **N**umérique

Rendus





Nommage à respecter strictement

Les noms des documents et des fichiers à rendre sont donnés dans le sujet. Vous devez observer strictement le nommage demandé. Ne les placez pas dans des dossiers si ce n'est pas demandé.

Dans ce document, **<nom_équipe>** doit être remplacé par le nom de votre équipe, comme par exemple **a14** pour l'équipe **4** du groupe TD **A1**.

Contraintes

Il est attendu que vous produisiez et déposiez certains documents sur Moodle. Vous devez respecter :

- Le délai imparti : 
- Le format des documents (un document peut aussi être un script) : 
- Le nommage des documents (voir **Nommage** ci-avant) : 
- Les attendus : 
- De faire figurer **dans chaque document ou fichier** : le nom d'équipe, les participants au document ainsi que le pourcentage de participation de chacun des membres à la production du document/fichier/script.
- La responsabilité du dépôt revient au 1^{er} membre dans la feuille Google Sheet que vous avez remplie avec les noms des membres de l'équipe. Vous ne pourrez pas en changer en cours de SAÉ.

Aucun délai supplémentaire ne sera accordé. Vous devez prendre vos dispositions pour faire un dépôt dans les temps. Comme indiqué ci-dessus, un seul membre est responsable de ce dépôt. Nous n'irons pas chercher un éventuel dépôt par un autre membre.

Le non-respect de tout ou partie de ces consignes pourra impacter votre note de SAÉ.

Le sujet NE vous demande PAS de créer des pages Web pour le site. Vous aurez uniquement à produire des fichiers bruts (fichiers texte, images) qui pourraient servir, dans le cadre d'un réel projet d'entreprise, à fournir de la matière à l'équipe de développement du projet Web.

Phase A - Analyse d'un jeu de fichiers

Sur Moodle, vous disposez d'un jeu de fichiers contenant :

- Des images
- Du texte
- Des tableaux au format Excel

Ce jeu de fichiers n'est pas obligatoirement une liste exhaustive de tous les fichiers reçus, mais il couvre tous les cas de figure que vous pouvez rencontrer pour ce site Web.

Nous pourrions tester vos scripts avec un autre jeu de fichiers similaires. Donc, vous ne devez faire aucune supposition basée sur le contenu spécifique des fichiers donnés en exemple. Par exemple, un fichier de médailles peut contenir un nombre de lignes différent mais les champs seront toujours les mêmes et de même nature. Le nombre de paragraphes d'un fichier texte peut différer avec un autre jeu de fichiers, etc.

Etape 1 - Déterminer les attentes (semaine 1)



Rendu : un document PDF nommé **etape1-<nom_équipe>.pdf** sous forme d'un tableau de synthèse.



Délai : minuit la veille de la séance 2 de SAÉ encadrée



Dépôt : Moodle



Attendus :

L'équipe de développement Web a les besoins suivants. Note : les tags **HTML** indiqués ci-dessous ne doivent pas être dans les fichiers, ils sont donnés à titre indicatif pour vous expliquer comment l'équipe de développement Web utilisera le contenu de vos fichiers dans leurs pages Web.

- Des images au format **webp** d'un poids maximal de **150 Ko** et de dimension (sans déformation) minimale **320x200** et maximale de **960x600**. La taille maximale respectant le poids maximal doit être privilégiée.
- Des textes bruts structurés de la façon suivante :
 - la 1^{ère} ligne servira de titre **<h1>**
 - chacune des lignes suivantes servira de paragraphe **<p>**
- Des documents texte au format CSV pour fabriquer des **<table>** dont la 1^{ère} ligne servira d'entêtes **<th>** et les autres lignes de rangées de **<tr>** avec des champs (séparés par des virgules) servant de cellules **<td>**.

Sur la base de ce jeu de fichiers, rédigez un document synthétique destiné à l'équipe de développement du site Web et à votre chef de projet, listant :

- La nature de chaque fichier du jeu de test (type, taille, poids, format, etc.)

- Un commentaire indiquant si le fichier est adapté pour une exploitation telle que celle décrite par l'équipe de développement du site et, le cas échéant, la ou les raisons pour lesquelles il ne le serait pas
- Les éventuelles actions que vous proposez d'opérer sur chacun de ces fichiers pour une utilisation possible sur le site Web.
- Les gains et avantages éventuels obtenus grâce à ces actions

Etape 2 - Panoplie d'outils (semaine 1)

Vous disposez déjà de tous les outils vus en R1.04 (Bash, PHP, les filtres). Ce sont les outils de base et il ne sert à rien d'aller chercher d'autres outils si vous trouvez déjà votre bonheur dans la combinaison d'un certain nombre d'entre eux.

Vous disposez aussi d'outils supplémentaires que nous vous fournissons pour les besoins de cette SAÉ mais qui, dans la vraie vie d'un·e professionnel·le du développement, vous seraient déjà connus grâce à votre expérience ou par vos recherches sur Internet. Pour vous simplifier la tâche, nous avons déblayé le terrain pour vous ! 😊

Contrairement aux filtres Unix, ces outils supplémentaires ne sont pas installés sur les postes de travail de l'IUT mais vous sont livrés sous forme d'images Docker.

Ces images sont disponibles sur le Hub de l'IUT en n'oubliant pas de faire d'abord un :

```
docker image pull <nom_image>:<version>
```

ou sur vos ordinateurs personnels en préfixant chaque nom d'image d'un **bigpapoo/** :

```
docker image pull bigpapoo/<nom_image>:<version>
```

Note sur l'usage des images fournies

Les images que nous vous fournissons et que nous allons décrire un peu plus loin ont toutes le même mode de fonctionnement :

- Vous êtes accueilli par un Bash dans chaque conteneur. De là, si vous êtes en mode interactif, vous pourrez lancer la commande principale de chaque images (cette commande sera donnée dans le descriptif donné pour chaque image plus loin)
- Dossier de travail : **/work**. Si vous créez un conteneur en mode interactif avec l'une de ces images, vous vous retrouverez dans ce dossier.

- Dossier de données : **/data**. Vous n'avez pas l'obligation d'utiliser ce dossier mais il existe dans chaque image et est certainement l'endroit idéal pour y placer vos fichiers et documents de travail (ceux à convertir et ceux que vous produisez).
- Pour envoyer vos données (fichiers à traiter) et récupérer les fichiers produits, vous utiliserez la commande **docker container cp**. que vous avez vue dans **SAÉ 1.03 - Partie 2 - Utilitaires**.

sae103-excel2csv

Sans surprise et comme son nom le laisse entendre, cette image permet de convertir une feuille Excel (**.xlsx**) en un fichier au format CSV (**.csv**), un format que vous devez connaître, mais si ce n'est pas le cas, documentez-vous sur Internet.

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

sae103-imagick

Celle-ci est moins évidente. Elle est construite autour d'un outil Unix très puissant qui est un grand classique de la transformation d'images : **Image Magick**. La commande qui s'exécute dans ce conteneur s'appelle **convert**. Consultez le site Web d'Image Magick (<https://imagemagick.org>) pour découvrir notamment les types de fichiers et d'images que cet outil est capable de convertir et de créer.

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

sae103-html2pdf

Est-il besoin de présenter celle-ci ? Vous l'avez déjà rencontrée dans le sujet **SAÉ 1.03 - Partie 2 - Utilitaires**. Vous devez utiliser cette image-ci avec le nom **sae103-html2pdf** mais son utilisation reste similaire.

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



Rendu : une archive nommée **etape2-<nom_équipe>.zip** contenant un documents PDF et les fichiers associés décrits dans les **Attendus**



Délai : minuit la veille de la séance 2 de SAÉ encadrée



Dépôt : Moodle



Attendus :

- Vous devez convertir tous les fichiers et documents qui le nécessitent au regard de ce que vous avez déclaré dans le 1^{er} PDF.
- Vous devez rédiger un document PDF nommé **etape2-<nom_équipe>.pdf** qui décrit, pour chaque fichier converti, la façon dont vous avez procédé, y compris la façon dont vous avez envoyé et récupéré les fichiers dans le conteneur.

- Vous devez proposer (par écrit dans le PDF) une solution pour pouvoir traiter non plus manuellement des fichiers individuels mais une série de fichiers de même nature (par exemple un lot d'images ou encore un lot de fichiers Excel). Le traitement proposé doit se faire sans intervention humaine. Pour le moment, il n'est pas demandé de réaliser de script ou de coder quoi que ce soit pour réaliser ces actions, vous devez juste décrire le mécanisme que vous proposez de mettre en place.
- Créez l'archive **etape2-<nom_équipe>.zip** contenant les fichiers convertis et le document PDF.

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 3 - 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 : **C**ommand **L**ine **I**nterface, (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 médailles 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.
- Il faut opérer le calcul du total des médailles par pays (Or + Argent + Bronze)
- Il faut trier les pays par ordre décroissant de médailles d'Or, puis d'Argent pour les pays ayant le même nombre de médailles d'Or, puis de Bronze pour les départager en cas d'égalité sur l'Or et l'Argent.
- Les pays ayant le même nombre de médailles d'Or, d'Argent et de Bronze doivent avoir le même classement (dans ce cas, les ex-æquo doivent être triés par ordre alphabétique) et le rang des ex-æquo (à partir du second) doit afficher un tiret (-) au lieu du rang. Exemple :

```
44,Afrique du Sud,1,3,2,6  
-,Jamaïque,1,3,2,6  
-,Thaïlande,1,3,2,6
```

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 des drapeaux miniatures devant chaque pays.

L'équipe de développement du site Web a besoin que vous lui fournissiez, en même temps que le fichier des médailles, un dossier complet avec tous les drapeaux des pays présents dans le tableau des médailles (et uniquement ceux-là), et ce en deux formats : **largeur 20 pixels et hauteur variable** pour tous les drapeaux et en 80x60 pixels en style "flottant". Les noms doivent être le code ISO Alpha 2 du pays

(https://fr.wikipedia.org/wiki/ISO_3166-1)

Pour ce faire, vous devez télécharger dynamiquement chaque drapeau depuis le site <https://flagcdn.com>.

Consultez <https://flagcdn.com> et <https://flagpedia.net/download/api> qui détaillent le format d'URL permettant de télécharger chaque drapeau dans le format souhaité. Attention, il est possible de télécharger des packs de toutes les images. Ce n'est pas ce qui est attendu, vous devez scripter le téléchargement de chaque drapeau individuel.

Inspirez-vous de la section "Embed on your website" à l'URL

<https://flagpedia.net/download/api> pour savoir comment formater une URL permettant de récupérer des images individuelles (note : **wget** qui est évoqué ci-après permet de faire ce qu'un navigateur fait, notamment quand il récupère les images associées aux balises **<img...>**)

Vous disposez aussi d'une nouvelle image Docker pour vous y aider : **sae103-wget**

sae103-wget

La commande **wget** est un outil Unix permettant de récupérer n'importe quelle ressource Web (une page HTML, une image, un fichier CSS, un script JS, etc.) comme le fait un navigateur, mais en ligne de commande. Évidemment ce qui est ainsi récupéré n'est pas affiché (**wget** n'a pas cette capacité) mais simplement stocké sur disque. Faites des essais.

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



Rendu : une archive nommée **etape3-<nom_équipe>.zip** qui doit être la compression d'un dossier **etape3/** contenant tout le nécessaire pour effectuer les traitements de la chaîne.



Délai : la date qu'on fixera ensemble pour le rendu final



Dépôt : Moodle



Attendus :

Les scripts nécessaires pour effectuer les traitements de la chaîne, ainsi que le script mettant en œuvre la chaîne.

Vous devez aussi fournir une documentation PDF sur la manière de lancer la chaîne pour quelqu'un ne travaillant pas dans votre équipe (votre enseignant par exemple 😊)

N'oubliez pas de détailler qui a fait quoi et en quelle proportion.

Etape 4 - Le tableau des médailles en PDF

Le client veut mettre en ligne sur son site, un PDF d'une seule page affichant le tableau des médailles avec les noms des pays, leur classement, le décompte des médailles de chaque métal et un petit drapeau du pays devant son nom.

Il veut aussi ajouter une dernière colonne représentant le pourcentage de médailles que chaque pays a obtenus. C'est un pourcentage global, tous métaux confondus.

Ce tableau doit tenir sur une unique page A4 et peut (doit) donc être dessiné sur plusieurs colonnes. Prévoir un titre et un logo JO Paris 2024.

Comme au TP 2 sur Docker, votre script doit générer un fichier 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 ce PDF. 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 **etape4-<nom_équipe>.zip** qui doit être la compression d'un dossier **etape4/** contenant ce qui est décrit dans les **Attendus**.



Délai : la date qu'on fixera ensemble pour le rendu final



Dépôt : Moodle



Attendus :

Les scripts de génération du PDF et de lancement de cette génération, ainsi qu'un exemple de PDF produit.