

Etape 2  
SAÉ 1.03  
Installation d'un poste de  
développement

Le Magadur Armelle  
Keraudren Johan  
Duvernet–Le Bras Gwendal  
Martin Esmeralda

Groupe E16

# Etape 1 :

## Récupération des images.

On considère que le client ne se situe pas à l'iut donc on va récupérer les images sur l'hub distant.

## Récupération de la première image (sae103-excel2csv).

Commande à exécuter sur un terminal ;

```
keraudrenjohan@jkeraudren ~ % docker image pull bigpapoo/sae103-excel2csv:latest
```

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

(latest = dernière version) => On utilise le hub « bigpapoo/ » car nous ne sommes pas à l'iut.

## Récupération de la deuxième image (sae103-imagick).

Commande à exécuter dans un terminal ;

```
keraudrenjohan@jkeraudren ~ % docker image pull bigpapoo/sae103-imagick:latest
```

Même explication qu'au-dessus .

## Récupération de la troisième image (sae103-html2).

Commande à exécuter dans un terminal ;

```
keraudrenjohan@jkeraudren ~ % docker image pull bigpapoo/sae103-html2pdf
```

Même explication qu'au-dessus .

## Étape 2 :

### Gestion du fichier tableur :

Ouvrir 2 terminaux différents.

#### Sur le 1<sup>er</sup>

-création d'un container :  
Commande

```
keraudrenjohan@jkeraudren ~ % docker run -ti bigpapoo/sae103-excel2csv
```

Se déplacer via la commande `cd /app`.

#### Sur le 2<sup>ième</sup> terminal :

Se placer dans le fichier où se situe le fichier à convertir.

On va ensuite envoyer le fichier dans notre container avec la commande `docker cp`.

```
keraudrenjohan@jkeraudren ~ % docker cp ./<nom_fichier>.xlsx <id_container>/app
```

#### Retour sur le 1<sup>er</sup>.

Normalement votre container est lancé (sinon les autres étapes n'ont pas fonctionné).

Ainsi, dans votre container vous devez réaliser la conversion via la commande `ssconvert`

(Étant sur ordi perso ,je n'arrive pas à lancer de container , ainsi l'image sert juste de démonstration .)

```
keraudrenjohan@jkeraudren ~ % ssconvert 'Tableau des medailles v1'.xlsx resultat.csv
```

#### Sur le 2<sup>ième</sup> terminal :

Récupération du fichier à l'intérieur du container , pour cela on va faire l'inverse que pour l'envoi des fichiers

```
keraudrenjohan@jkeraudren ~ % docker cp <id_container>/app/resultat.csv resultat.csv
```

Vous avez maintenant votre fichier au bon format !

## Gestion des fichiers image:

Même processus que pour les fichiers tableurs.  
Ouvrir 2 terminaux différents.

1<sup>er</sup> terminal :

On ouvre un terminale avec l'image.

```
keraudrenjohan@jkeraudren ~ % docker run -ti bigpapoo/sae103-imagick
```

2<sup>ème</sup> terminale :

Envoie des fichiers dans le container:

```
keraudrenjohan@jkeraudren ~ % docker cp <nom_fichier> <id_container>:/data
```

Retour sur le 1<sup>er</sup>.

Exécution des commandes :

- Pour changer la taille d'une image :

```
keraudrenjohan@jkeraudren ~ % convert <nom_fichier> -resize taille*taille <nom_fichier>.format
```

(On peut aussi changer le format de notre image via cette commande.)

- Pour changer le format :

```
keraudrenjohan@jkeraudren ~ % convert <nom_fichier> -define format <nom_fichier>.format
```

- Pour changer le poids de l'image on peut changer sa qualité :

```
keraudrenjohan@jkeraudren ~ % convert <nom_fichier> -quality 75 <nom_fichier>.format
```

Sur le 2<sup>ème</sup> terminal :

Récupération du fichier à l'intérieur du container , pour cela on va faire l'inverse que pour l'envoi des fichiers

```
keraudrenjohan@jkeraudren ~ % docker cp <id_container>/app/<fichier_resultat>.format <fichier_resultat>.format
```

Vous avez maintenant vos fichiers au bon format et à la bonne taille !

Attention, pour l'image .PDF, aucune solution trouvée.

## Gestion des fichiers texte :

Même processus que pour les fichiers tableurs.  
Ouvrir 2 terminaux différents.

### 1<sup>er</sup> terminal :

On ouvre un terminal avec l'image.

```
keraudrenjohan@jkeraudren ~ % docker run -ti bigpapoo/sae103-html2pdf
```

### 2<sup>ème</sup> terminal :

Envoie des fichiers dans le container :

```
keraudrenjohan@jkeraudren ~ % docker cp <nom_fichier> <id_container>:/data
```

### Retour sur le 1<sup>er</sup>.

Exécution des commandes :

(Possible solution , cela ne fonctionne pas pour nous.)

```
keraudrenjohan@jkeraudren ~ % weasyprint cyclisme1.pdf -m webp cyclisme.webp
```

### Sur le 2<sup>ème</sup> terminal :

Récupération du fichier à l'intérieur du container, pour cela on va faire l'inverse que pour l'envoi des fichiers

```
keraudrenjohan@jkeraudren ~ % docker cp <id_container>/app/<fichier_resultat>.format <fichier_resultat>.format
```

## Idée d'automatisation :

- Utiliser un script permettant d'ouvrir le container , en le renommant pour ne pas avoir besoin de récupérer son ID , ensuite envoyer le fichier avec docker cp et ensuite l'exécuter avec docker exec puis le récupérer comme vu précédemment.
- On peut utiliser du script mais aussi du php pour vérifier si le script à bien fonctionner.

Travail réalisé à 100% par Keraudren Johan.