

■ Actions effectuées sur chaque fichier :

Avant de commencer à modifier les fichiers :

il faut récupérer les images dockers nécessaires avec les commandes :

```
docker image pull bigpapoo/imagick
```

```
docker image pull bigpapoo/sae103-excel2csv
```

puis lancer les conteneurs correspondants avec les commandes :

```
docker container run -ti --platform=Linux/amd64 bigpapoo/imagick
```

```
docker container run -ti --platform=Linux/amd64 bigpapoo/sae103-excel2csv
```

On obtient f1 (l'ID du conteneur imagick) et f2 (celui du conteneur sae103-excel2csv) avec la commande "docker container ls" dans un second terminal qui n'est pas lié à un des conteneurs. On renomme T1 le terminal dans lequel on peut interagir avec le conteneur imagick, T2 celui pour sae103-excel2csv et T3 celui qui n'est pas lié avec un conteneur et qui se situe dans le dossier où se trouve le jeu de fichiers fourni décompressé.

Modification des fichiers :

- 'aviron.jpeg' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp aviron.jpeg f1:/work
```

Dans le conteneur (T1), on exécute la commande :

```
convert aviron.jpeg -resize 960x600! aviron.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée et la convertir au format WEBP.

Enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/aviron.webp .
```

- 'beach-volley.png' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp beach-volley.png f1:/work
```

dans le conteneur donc dans T1, on exécute la commande :

```
convert beach-volley.png -resize 960x600! beach-volley.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée et la convertir au format WEBP.

Enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/beach-volley.webp .
```

- 'départ-triathlon.jpeg' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp départ-triathlon.jpeg f1:/work
```

dans le conteneur donc dans T1, on exécute la commande

```
convert départ-triathlon.jpeg -resize 960x600! départ-triathlon.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée et la convertir au format WEBP.

Enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/départ-triathlon.webp .
```

- 'podium.png' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp podium.png f1:/work
```

dans le conteneur donc dans T1, on exécute la commande :

```
convert podium.png -resize 960x600! podium.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée et la convertir au format WEBP.

Enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/podium.webp .
```

- 'teddy-riner.webp' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp teddy-riner.webp f1:/work
```

dans le conteneur donc dans T1, on exécute la commande :

```
convert teddy-riner.webp -resize 960x600! teddy-riner.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée.
enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/teddy-riner.webp .
```

- 'cyclisme1.pdf' :

ajout de l'image dans le répertoire /work du conteneur f1 avec la commande dans T3 :

```
docker cp cyclisme1.pdf f1:/work
```

dans le conteneur donc dans T1, on exécute la commande :

```
convert cyclisme1.pdf -resize 960x600! cyclisme1.webp
```

ce qui redimensionne l'image en 960x600 pixels stricts, donc l'image est déformée et convertie au format WEBP.

Enfin, on récupère l'image modifiée avec la commande (dans T3) :

```
docker cp f1:/work/cyclisme1.webp .
```

- 'Tableau des médailles.xlsx' :

ajout du fichier dans le répertoire /app du conteneur f2 avec la commande dans T3 :

< CHAUVEL Arthur - LE BOURHIS Maël - LE CHEVERE Yannis - LE SECH Marceau >
< IUT de Lannion - Année 2024/2025 - SAÉ 1.03 - 3/5 >

docker cp 'Tableau des médailles.xlsx' f2:/app

dans le conteneur donc dans T2, on exécute la commande :

ssconvert 'Tableau des médailles.xlsx' 'Tableau des médailles.csv'

ce qui convertit le tableur en fichier CSV, format attendu par l'équipe de développement.
Enfin, on récupère le fichier modifié avec la commande (dans T3) :

docker cp f2:/app/'Tableau des médailles.xlsx' .

- 'Sports emblématiques.data' :

ajout du fichier dans le répertoire /app du conteneur f2 avec la commande dans T3 :

docker cp 'Sports emblématiques.data' f2:/app

dans le conteneur donc dans T2, on exécute la commande

ssconvert 'Sports emblématiques.data' 'Sports emblématiques.txt'

ce qui convertit le fichier en fichier texte brut au format TXT, attendu par l'équipe de développement.
Enfin, on récupère le fichier modifié avec la commande (dans T3) :

docker cp f2:/app/'Sports emblématiques.txt' .

Ensuite, il faut modifier le fichier, il faut créer un dossier 'sportsEmblématiques', puis dans le dossier créer les fichiers 'athlétisme.txt', 'cyclisme.txt', 'sportAquatiques.txt' et 'sportsCollectifs.txt'.

Dans le dossier 'athlétisme.txt', coller la section 1 du fichier 'Sports Emblématique.data'.
Pour le fichier 'cyclisme' la section 2, pour le fichier 'sportsCollectifs' la section 3 et pour le fichier 'sportsAquatiques' la section 4.

Ainsi nous avons un dossier avec 4 fichiers textes contenant les 4 sections du fichier 'Sports emblématiques.data' initial.

■ Description du mécanisme d'automatisation

Mécanisme d'automatisation :

Conformément à la consigne, les fichiers à traiter appartiennent à des catégories similaires (séries de fichiers de même nature) et seront organisés dans des dossiers spécifiques selon leur type.

Automatisation du traitement :

Le programme s'appuiera sur une boucle principale qui parcourra les différents dossiers à traiter. Pour chaque dossier, une seconde boucle analysera les fichiers qu'il contient et appliquera automatiquement les transformations nécessaires en fonction de leur type :

Images : Redimensionnement à des dimensions fixes (960 x 600 pixels) et conversion au format .webp .

Fichiers Excel (.xlsx): Conversion au format CSV (.csv), format attendu par l'équipe de développement.

Fichiers .data : Conversion en fichiers texte. Si nécessaire, le contenu sera découpé en sections distinctes, qui seront ensuite enregistrées dans de nouveaux fichiers texte structurés.

Une fois le traitement terminé, les fichiers modifiés seront placés dans un répertoire de sortie (/work). À la fin de chaque traitement, le programme effectuera une vérification pour garantir que toutes les transformations ont été correctement réalisées.

■ Participation à l'étape

Pourcentage approximatif de la participation à la rédaction du document et la réalisation de l'étape.

CHAUVEL Arthur = 40%

LE BOURHIS Maël = 20%

LE CHEVERE Yannis = 20%

LE SECH Marceau = 20%