

# SAÉ 1.03

## Installation d'un poste de développement

### Contexte

Vous travaillez dans une ESN<sup>1</sup> et un client fait appel à votre société pour réaliser un outil capable de produire une série de PDF à destination de leurs équipes commerciales régionales.

Ce client est présent sur tout le territoire national et il veut présenter, à ses commerciaux, des résultats trimestriels dont il va vous fournir les informations brutes, que vous devrez traiter et formater en quelque chose de propre et d'agréable à lire.

Ce travail sera récurrent et, dans un premier temps, il devra pouvoir être réalisé par n'importe quel membre de l'équipe de développement.

La mise à disposition de cet outil à des utilisateurs finaux n'est pas prévue pour cette SAÉ, mais une documentation détaillée doit être fournie pour guider un technicien dans le traitement et la production des PDF, sans requérir de connaissance particulière du projet.

### Attendus

#### Environnement Dockerisé

Vous devrez monter un environnement de développement qui, à terme, pourrait devenir un environnement de production, même si ce dernier point ne fait pas partie des objectifs de cette SAÉ.

Cet environnement sera Dockerisé, paramétrable et documenté.

Les différents conteneurs devront s'échanger leur données de manière automatique (sans intervention de l'utilisateur) par l'usage de volumes mappés.

---

<sup>1</sup> Entreprise de **S**ervice du **N**umérique

Même si vos postes de travail disposent de certains des outils suivants, vous ne devez pas en faire usage et supposer que cette chaîne de traitement pourra se faire, à l'avenir, sur un ordinateur qui ne disposera que de Docker. Vous ne devez donc utiliser que des conteneurs faisant tourner ces outils. C'est la philosophie Docker et ce sera donc aussi la vôtre pour cette SAÉ.

## Langages et outils

Pour chaque étape, vous devrez respecter les langages et les outils demandés.

Vous aurez à votre disposition :

- PHP
- Bash
- Filtres Unix
- Images Docker : **sae103-php**, **sae103-qr**code, **sae103-imagick**, **sae103-html2pdf**

Attention, l'image **sae103-html2pdf** n'est pas la même que **html2pdf** du TP Docker !

Les images mises à votre disposition sont sur le dépôt Docker de l'IUT ainsi que sur le dépôt public de Docker. Si vous souhaitez travailler depuis l'extérieur, comme vous ferez tacitement appel au dépôt public, vous devez donc préfixer vos noms d'image par **bigpapoo/**, par exemple une image **sae103-php** à l'IUT devient **bigpapoo/sae103-php** chez vous.

Toutes les images disposent de **bash** et des filtres Unix. Pour les autres outils, seules les images dédiées disposent des outils en question.

Pour la SAÉ, utilisez toujours les images avec un tag **latest**. Il est possible que ces images évoluent un tout petit peu durant la période de SAÉ, mais vous serez tenus au courant si ça arrive et en utilisant **latest** vous êtes sûrs d'avoir la dernière version.

Attention, si vous avez déjà fait un **docker image pull** d'une image, même en utilisant le tag **latest**, Docker ne fera pas automatiquement la mise à jour si une nouvelle version **latest** est mise à disposition. C'est à vous de forcer les choses si besoin en ajoutant une option **--no-cache** au lancement d'un **docker image pull**. Encore une fois, vous serez avertis si ça arrive.

Astuce pour la syntaxe d'utilisation des images de cette SAÉ : sauf pour celles dont on vous donne la syntaxe dans le sujet, si vous lancez un **run** avec une image fournie et que vous ne précisez aucun autre argument après le nom de l'image, vous obtiendrez une aide sur la syntaxe à utiliser pour créer un conteneur afin d'exécuter l'action avec cette image. Dans ce cas, le conteneur ne fera rien d'autre après avoir affiché l'aide sur sa syntaxe et s'arrêtera immédiatement.

## Documentation

Rien ne devra être laissé sans explication ni à l'appréciation du technicien qui aura la charge de mettre en œuvre la chaîne d'outils. Vous devrez prévoir le nécessaire (par script et/ou en documentant) pour le guider de A à Z dans le processus.

L'idée est d'avoir un environnement reproductible aisément pour que la chaîne d'opérations puisse être facilement redéployée plusieurs fois par an, sur n'importe quel poste d'un développeur de l'équipe, et sans nécessiter de connaissance des outils ni de formation préalable à ces outils.

## Sources

Sur Moodle, vous trouverez des sources de données sous forme de fichiers bruts que vous allez devoir retravailler au moyen de scripts pour préparer le travail du générateur de PDF.

Ces fichiers contiendront le code de la région concernée. Ces codes sont au format **ISO 3166-2**. Consultez : [https://fr.wikipedia.org/wiki/Région\\_française](https://fr.wikipedia.org/wiki/Région_française).

Vous serez évalués avec le jeu de données fourni sur Moodle ainsi que sur un second jeu, de même nature, mais dont vous n'aurez pas connaissance au préalable. Pour clarifier les choses : vous devez être suffisamment génériques dans le traitement des données et ne pas vous appuyer sur ses caractéristiques figées comme, par exemple, supposer que les données ont toujours le même nombre de lignes. Vous devrez notamment vérifier la cohérence des données attendues et traiter les erreurs en conséquence, ou corriger automatiquement les données si c'est possible et opportun.

Certaines sources proviennent de sites Internet (Wikipédia et autres, pour les logos et les textes). Vous ferez figurer un crédit graphique et texte en dernière page du document.

Certaines informations qu'on vous demande de faire apparaître sur les PDF sont donc issues des fichiers qu'on va vous fournir et d'autres doivent être trouvées par vous-mêmes et placées dans un ou plusieurs fichiers de paramétrage. Ce sera différencié dans le détail qui suit (**config**<sup>2</sup> ou **client**<sup>3</sup>). Enfin, certaines informations seront **calculées**<sup>4</sup>.

## Rendu

Vous devrez produire un certain nombre de PDF depuis un lot de fichiers bruts.

---

<sup>2</sup> Dans un fichier de configuration que vous avez à créer et à remplir vous-mêmes.

<sup>3</sup> Donné par le client dans les fichiers qu'il fournit (source Moodle).

<sup>4</sup> Par votre code.

Vous devrez générer une unique archive compressée en **tar.gz** contenant tous les PDF. Voici la syntaxe pour créer une archive **tar.gz**, qui est un format incontournable dans le monde Unix/Linux<sup>5</sup>, et non pas **ZIP** qui est issu du monde Windows :

```
| tar czvf nom_archive.tar.gz nom_dossier_a_archiver
```

Avec ces options (**c** pour **create**), le fichier d'archive est écrasé s'il existe déjà. Il y a des options pour ajouter à une archive déjà existante (pas nécessaire ici, voir le **man** si besoin).

Voici la syntaxe pour lister le contenu d'une archive **tar.gz** :

```
| tar tzvf nom_archive.tar.gz nom_dossier_a_archiver
```

Le **t** est pour... une signification obscure que seuls les barbus connaissent... list ?

Voici enfin la syntaxe pour extraire le contenu d'une archive **tar.gz** :

```
| tar xzvf nom_archive.tar.gz nom_dossier_a_archiver
```

Le **x** est pour **extract**.

Vous aurez noté que les options ne sont pas préfixées d'un - (tiret), c'est une exception à la plupart des autres commandes Unix. Cependant, il existe des versions de **tar** qui acceptent aussi l'usage du préfixe - (tiret) classique. L'usage est classiquement de ne pas utiliser le tiret.

L'option **v** est pour **verbose**, vous pouvez l'omettre si vous voulez moins de logs durant l'exécution de la commande **tar**.

L'option **z** indique la compression, sans elle votre archive n'est pas compressée. Il s'agit du format **gzip** produit par la commande éponyme.

Enfin, l'option **f** pour **file**, permet d'indiquer qu'il faut créer l'archive sous forme d'un fichier dont le nom est donné dans l'argument suivant ce **f**.

Vous pourrez aussi trouver parfois l'extension **.tar.gz** raccourcie en **.tgz**.

Les PDF devront porter le nom de la région, préfixé de son code **ISO 3166-2**. Vous fournirez un fichier de configuration **regions.conf**, qui devra être dans un format facilement modifiable pour qu'il puisse être changé ou adapté à l'avenir. Ce fichier de configuration est à créer manuellement par vous, et contiendra tout ce qui n'est pas

---

<sup>5</sup> **tar** signifie **tape archive**, une archive sur bande (magnétique), oui ça remonte à très loin !

fourni par le client et dont vous pourrez avoir besoin (par exemple, la relation entre code **ISO 3166-2** et le nom de région). Vous choisirez un format texte+séparateur.

Exemple de nom de PDF : **FR-BRE-Bretagne.pdf**

Voici maintenant le contenu d'un PDF

## Page de couverture

- Nom de la région (**config**)
- Population (**config**)
- Superficie (**config**)
- Nombre de départements (**config**)
- Logo de la région (**client**)

Le logo sera fourni par le client, et les autres données seront à rechercher (une seule fois) par vos soins et à placer dans le fichier de configuration **regions.conf**, dont on a parlé précédemment.

## Page 1

- Un titre "Résultats trimestriels XX-YYYY" où XX est le numéro du trimestre et YYYY l'année en cours. Ces informations sont **calculées** avec la date du moment du traitement.
- Texte d'introduction fourni par le **client**.  
Les URL devront être cliquables en ne conservant que le texte associé, sans afficher l'URL dans le texte.  
Vous limiterez le texte pour que tout tienne dans une seule page<sup>6</sup>, tableau des résultats compris.
- Un tableau des résultats :
  - Colonne 1 : Nom du produit
  - Colonne 2 : Ventes du trimestre
  - Colonne 3 : Chiffre d'affaires du trimestre
  - Colonne 4 : Ventes du même trimestre année précédente
  - Colonne 5 : CA du même trimestre année précédente
  - Colonne 6 : Evolution de CA en %age et en valeur absolue. Vous afficherez la valeur en **gras-rouge** s'il y a une chute du CA et en **gras-vert** dans le cas contraire.

La colonne 6 est **calculée**, les autres sont fournies par le **client**.

---

<sup>6</sup> Explorez peut-être la propriété CSS **overflow** d'un tag HTML.

## Page 2

- Un titre “Nos meilleurs vendeurs du trimestre”.
  - La liste horizontale des photos (avatars) des 3 meilleurs commerciaux du trimestre, avec leur nom et le CA réalisé placés sous chaque photo. La liste doit être triée par CA réalisé décroissant.
- Les photos sont toutes fournies par le **client**.

## Page 3

- Un lien vers une page fictive du site de la société :  
**`https://bigbrain.biz/<code_region>`** où le **`<code_région>`** est au format **ISO 3166-2**.
- Un QRCode codant l'URL ci-dessus

## Pied de page

Vous ferez figurer la date et l'heure de génération du PDF sur chaque pied de page, au format **JJ-MM-AAAA HH:MM**.

# Etapes

La SAÉ se déroule en 3 semaines et 2 périodes de livrables.

Vous déposerez vos livrables sur Moodle.

## Semaines 1 et 2

Travaux, outils et livrables :

- Création du fichier de configuration **regions.conf**.  
Livrable : le fichier. Si besoin, ce fichier pourra évoluer en semaine 3 mais devra être relivré en semaine 3.
- Traitement du fichier de données d'une région pour en extraire 3 fichiers nettoyés et prêts à être fusionnés dans le document HTML final (voir semaine 3) :
  - le texte de la page 1 (**texte.dat**)
  - les données du tableau (**tableau.dat**)
  - les noms des meilleurs commerciaux (**comm.dat**).

Vous avez toute liberté pour le format de ces fichiers. Ils vont vous servir pour fusionner les modèles de pages HTML avec les données issues de ces fichiers.

Soyez donc judicieux dans vos décisions, pour simplifier le travail de fusion (en semaine 3).

Livrables : le script en **PHP** et un exemplaire de 3 fichiers générés (le texte, les données du tableau, les noms des meilleurs) depuis le fichier source fourni par le client. Vous utiliserez l'image **sae103-php** pour faire fonctionner votre script. Elle contient un **PHP 7.4**. Vous fournirez une petite documentation sous forme d'un fichier **LISEZMOI\_1.txt** décrivant comment lancer votre script.

- Dans un unique document HTML, les modèles de la page de couverture et des 3 pages de détail, avec du contenu fictif et statique (du *lorem ipsum* par exemple ou ce que vous voulez d'autre).

Livrables : le document **HTML** et sa feuille de style **CSS**.

Afin de générer des pages au format **A4**, nous vous fournissons sur Moodle<sup>7</sup> un exemple de document **HTML** + une feuille de style **CSS** qui, une fois "moulinés" par le conteneur de la semaine 3 (voir **sae103-htm12pdf** plus loin), produisent un **PDF** de 2 pages **A4**. Analysez et inspirez-vous de ce modèle très sommaire pour créer votre modèle qui servira de base pour les PDF.

- Un fichier **TACHES\_1.txt** détaillant la liste des tâches réalisées en semaines 1-2 et la répartition du travail individuel (en %) dans le groupe

## Semaine 3

Travaux, outils et livrables :

- Conversion en **PNG** en nuances de gris des avatars fournis par le client. Les images finales doivent avoir une taille de **200x200** pixels et le bandeau "Created on Face Co" doit être coupé. Vous écrirez un script Bash pour traiter le lot. Ce script devra fonctionner dans un conteneur à base d'une image **sae103-imagick**.

Livable : le script Bash qui doit tourner dans un conteneur. Vous fournirez une petite documentation sous forme d'un fichier **LISEZMOI\_2.txt** décrivant comment lancer votre script.

- Génération du QRCode de la dernière page, ça ne devrait pas être trop dur pour ça 😎

Livable : un exemple de QRCode. Vous fournirez une petite documentation sous forme d'un fichier **LISEZMOI\_3.txt** décrivant comment lancer votre script de génération.

- Transformation du modèle **HTML** créé en semaines 1-2 en un script **PHP** qui va remplacer (fusionner) les textes et les parties factices par les vraies données issues des 3 fichiers préparés aussi en semaines 1-2. Vous utiliserez un conteneur sur la base d'une image **sae103-php** et, pour chaque région dont vous aurez reçu

---

<sup>7</sup> **sample-html1-to-pdf.html** et **sample-style-to-pdf.css**

un fichier de la part du client, vous produirez un document final en **HTML** qui va servir à la transformation en PDF (l'étape suivante). Vous avez un exemple dans Moodle : **sample-model.html** qui est converti en **sample-model.php** et qui exploite un fichier **sample-model.data**.

Attention, cet embryon de code n'est qu'un exemple de transformation d'un modèle de page statique en une page-script **PHP** qui est encore loin de ce que vous devez produire, mais constitue un premier pas dans la bonne direction ! 😎

Livrable : le script **PHP** permettant la fusion, qui est issu du modèle **HTML**. Vous fournirez une petite documentation sous forme d'un fichier **LISEZMOI\_4.txt** décrivant comment lancer votre script de génération.

- Génération du PDF d'une région dans un conteneur à base d'une image **sae103-htm12pdf**<sup>8</sup> dont voici la syntaxe d'exécution :

```
docker run --rm -ti -v /Docker/<votre_login>:/work →  
sae103-htm12pdf "html2pdf fichier.html fichier.pdf"
```

Attention, c'est tout sur une seule ligne, les → indiquent que ça se poursuit, en dessous, sur une seule et même ligne de commande, sans retour à la ligne.

Livrable : tous les fichiers nécessaires ainsi qu'un script permettant de lancer le processus de génération du PDF, dans un conteneur sur base d'une image **sae103-htm12pdf**.

Vous fournirez une petite documentation sous forme d'un fichier **LISEZMOI\_5.txt** décrivant comment procéder pour générer le PDF final à partir des fichiers nécessaires.

- Le fichier **regions.conf**, qu'il ait évolué ou pas depuis le 1<sup>er</sup> livrable.
- Une documentation sous forme d'un fichier **LISEZMOI\_FINAL.txt** et tout le nécessaire pour traiter tous les PDF et produire une archive contenant tous les PDF.

Cela signifie donc de livrer aussi un script Bash permettant de lancer la chaîne de processus jusqu'à la production finale de l'archive de PDF. Puisqu'il devra enchaîner le lancement de plusieurs conteneurs, ce script sera donc évidemment exécuté sur l'hôte et non pas dans un conteneur. L'action du **tar** sera faite aussi, dans ce script, sur l'hôte.

- Un fichier **TACHES\_2.txt** détaillant la liste des tâches réalisées en semaine 3 et la répartition du travail individuel (en %) dans le groupe.

Bon courage à tous !

---

<sup>8</sup> Ne fonctionne que sur un vrai Linux, pas sur macOS.