

# **Guide d'installation**

# **API de comparaison des risques**

## **version prototype**

## Table des matières

1	Préambule.....	3
2	Prérequis.....	3
3	Premier exemple.....	3
3.1	Contexte.....	3
3.2	Procédure.....	3
3.2.1	Créer le service.....	3
3.2.2	Tester l'API.....	4
3.2.3	Logs.....	4
3.2.4	Base de données.....	4
3.2.5	Dupliquer le service.....	5
4	Second exemple.....	5
4.1	Contexte.....	5
4.2	Procédure.....	5
4.2.1	Créer le service.....	5
4.2.2	Tester l'API.....	6
4.2.3	Dupliquer le service.....	6
4.2.4	Logs et base de données.....	6
4.3	Mock data.....	6
4.4	Commandes indispensables.....	6

# 1 Préambule

Ce document à pour but d'aider au déploiement de l'API de comparaison des risques. L'outil choisi est « docker ». L'API est disponible dans les versions windows et linux. Deux exemples seront présentés, tout deux avec Windows 10, le second se faisant avec le concours d'une machine virtuelle sous linux.

# 2 Prérequis

Docker installé et fonctionnel. L'installation en elle même ne sera pas abordée, vous trouverez plus d'informations ici :

<https://docs.docker.com/engine/installation/#supported-platforms>

# 3 Premier exemple

## 3.1 Contexte

Windows 10

Docker for windows

windows container

## 3.2 Procédure

### 3.2.1 Créer le service

Récupérer l'image du service

```
docker pull sepulwarrior/comparaisonrisques:prototype-windows
```

Créer un container à partir de l'image récupérée

```
docker create --name compa_risques_01 sepulwarrior/comparaisonrisques:prototype-windows
```

Démarrer le container

```
docker start compa_risques_01
```

Récupérer son IP

```
docker exec compa_risques_01 ipconfig
```

### 3.2.2 Tester l'API

{URL\_de\_base}/api/patient?limit=1 ( retourne la première fiche patient )

{URL\_de\_base}/swagger ( testing )

L' {URL\_de\_base} sera cette IP. ( le container écoute sur le port 80 )

### 3.2.3 Logs

Des logs sous forme de fichiers sont disponible ici : c:\app\log (arborescence du container)

Il est possible de lier ce répertoire avec un répertoire de l'hôte. ( ici  
C:\DockContainersSharedFiles\compa\_risques\_01\log pour l'exemple )

```
docker create --name compa_risques_02 -v  
C:\DockContainersSharedFiles\compa_risques_01\log:c:\app\log  
sepulwarrior/comparaisonrisques:prototype-windows
```

De cette manière les logs sous forme de fichiers sont partagés avec l'hôte.

### 3.2.4 Base de données

La base de données utilisée est SQLite, celle-ci est donc un fichier se trouvant ici : c:\app\data (arborescence du container)

En théorie\*, il est possible de lier ce répertoire avec un répertoire de l'hôte. ( ici  
C:\DockContainersSharedFiles\compa\_risques\_01\data pour l'exemple )

```
docker create --name compa_risques_02 -v  
C:\DockContainersSharedFiles\compa_risques_01\data:c:\app\data  
sepulwarrior/comparaisonrisques:prototype-windows
```

Si deux containers sont monté sur le même répertoire, ces deux containers partageront de ce fait une base de données commune.

\*Les tests ont levé des erreurs avec Windows, avec linux ça fonctionne.

### 3.2.5 Dupliquer le service

Répétez l'opération, chaque container aura une IP différente.

## 4 Second exemple

### 4.1 Contexte

Windows 10

Docker toolbox windows

utilisation de virtualbox

linux container

### 4.2 Procédure

#### 4.2.1 Créer le service

Récupérer l'image du service

```
docker pull sepulwarrior/comparaisonrisques:prototype-linux
```

Créer un container à partir de l'image récupérée

```
docker create --name compa_risques_03 -p 81:80  
sepulwarrior/comparaisonrisques:prototype-linux
```

Notez l'option "-p 81:80", celle-ci permet de mapper le port 80 du container au port 81 de la machine virtuelle.

Démarrer le container

```
docker start compa_risques_03
```

Le container tourne dans une machine virtuelle (virtualbox)

Récupérer l'IP de la machine virtuelle

```
docker-machine -ls
```

### 4.2.2 Tester l'API

{URL\_de\_base}/api/patient?limit=1 ( retourne la première fiche patient )

{URL\_de\_base}/swagger ( testing)

L' {URL\_de\_base} sera cette IP. ( sur le port mappé par l'option -p )

### 4.2.3 Dupliquer le service

Dans ce cas-ci dupliquer le service en mappant sur des port différents de la machine virtuelle.

### 4.2.4 Logs et base de données

De la même manière que vu précédemment sous windows, les répertoires /app/log et /app/data peuvent être montés. Il seront montés sur la machine virtuelle.

## 4.3 Mock data

Dans le répertoire data se trouve également le fichier mock\_data.json, ce fichiers contient une base de données test. Si le fichier existe et que la base de données est vide, le service récupérera les données de test au démarrage. Supprimez le fichier mock et la base de données, redémarrez le service et vous commencerez avec une base de données vierge. Dans le cas ou le répertoire app\data est monté, celui-ci ne contiendra pas le fichier mock (système de fichier de l'hôte), une copie du fichier mock est disponible ici : \app\mock.

## 4.4 Commandes indispensables

Se logger ( sur docker hub pour)

```
docker login
```

Lister toutes les images

```
docker images
```

Lister tout les containers

```
docker ps -a
```

Créer et démarrer un container ( create + start )

```
docker run compa_risques_01
```

Attacher un container (utile pour voir les logs)

```
docker attach compa_risques_01
```

Paramètres d'un container ( e.g. Hostname, IP )

```
docker inspect compa_risques_01
```

Exécute une commande à l'intérieur du container (e.g. ipconfig)

```
docker exec compa_risques_01 ipconfig
```

Ouvre une invite de commande dans le container windows

```
docker exec -it compa_risques_01 cmd
```

Ouvre une invite de commande dans le container linux

```
docker exec -it compa_risques_01 sh
```

Pour le reste :

<https://docs.docker.com/engine/reference/commandline/docker/>