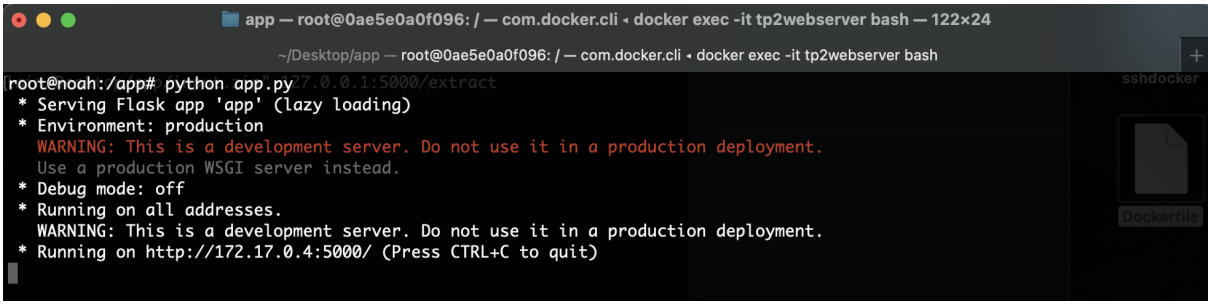
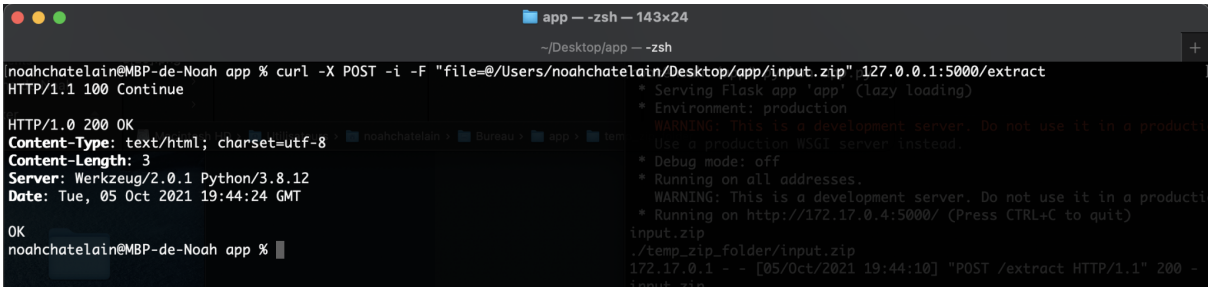
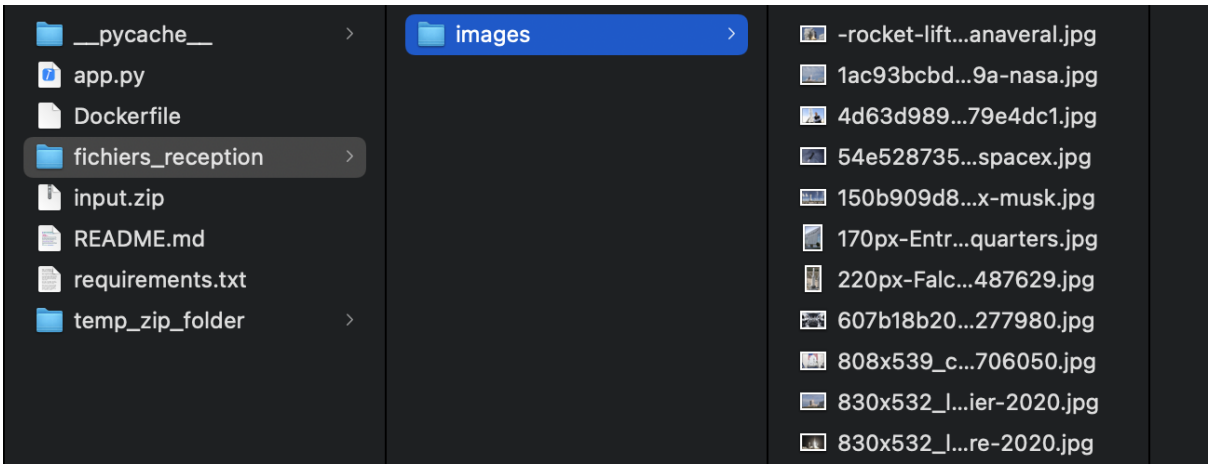
TP2 WebServers - Noah Chatelain & Ambroise Gyre

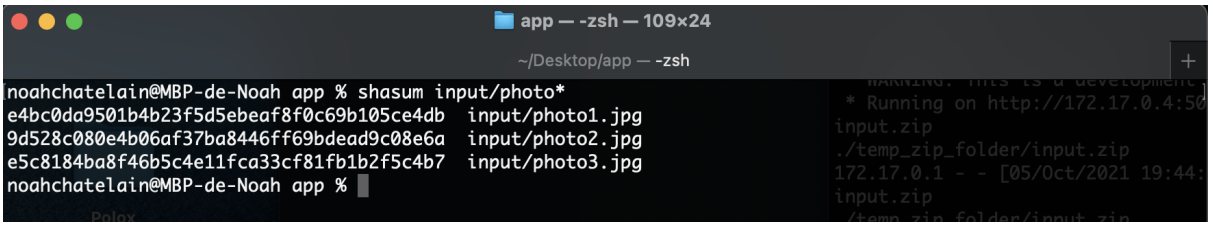
Création d'une route permettant d'envoyer un fichier .zip au serveur, le serveur le décompresse, puis le place dans le dossier /*fichiers_reception*



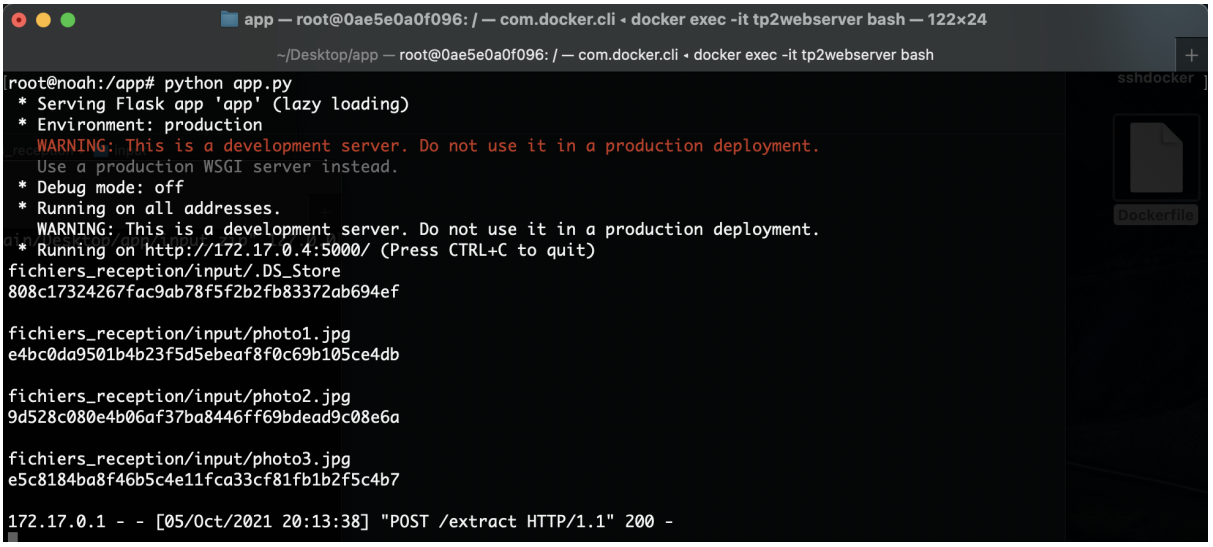




2ème étape, obtenir le code SHA-1 de nos photos qui seront à l'intérieur de l'archive *input.zip* à envoyer à notre serveur

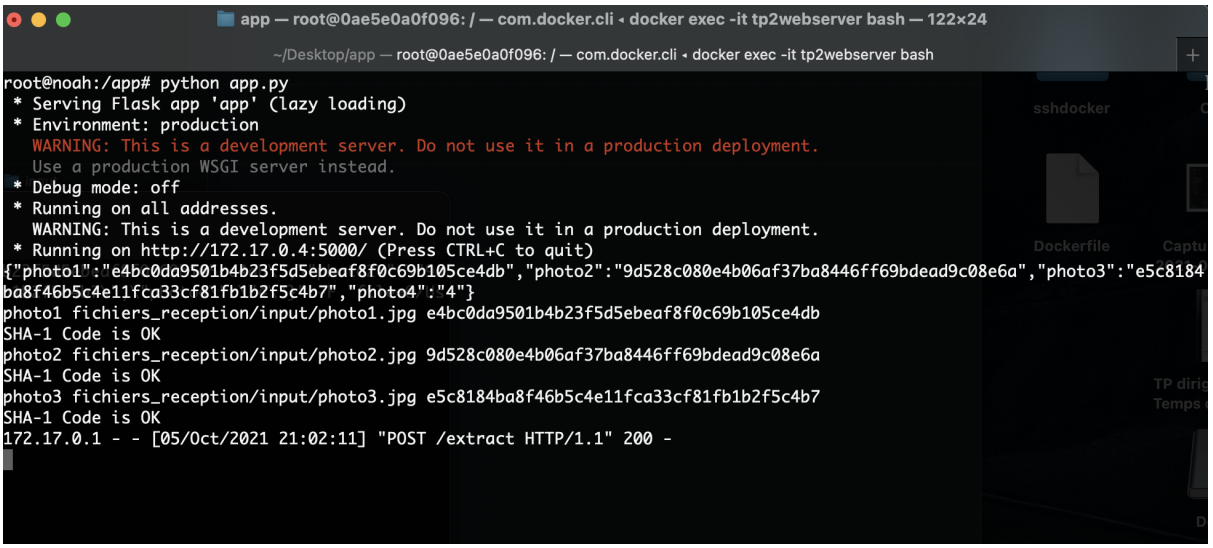


Maintenant que nous connaissons nos code SHA-1, nous pouvons vérifier en python que nous recevons bien les bonnes images

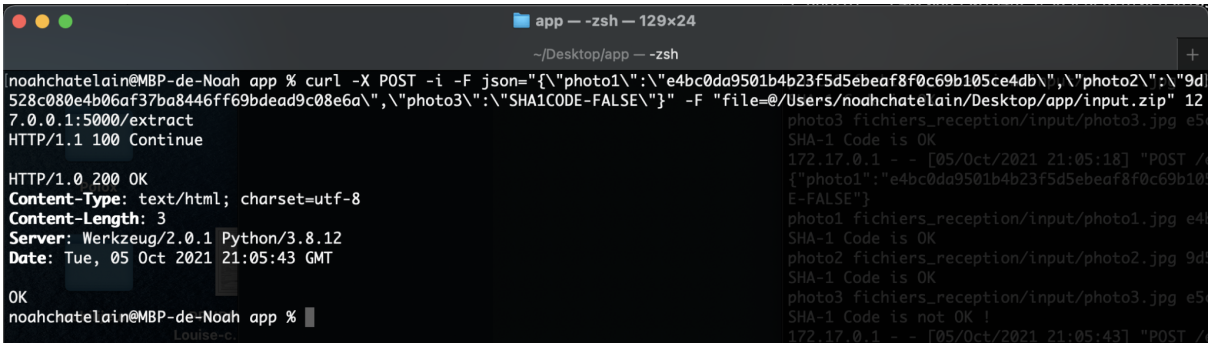


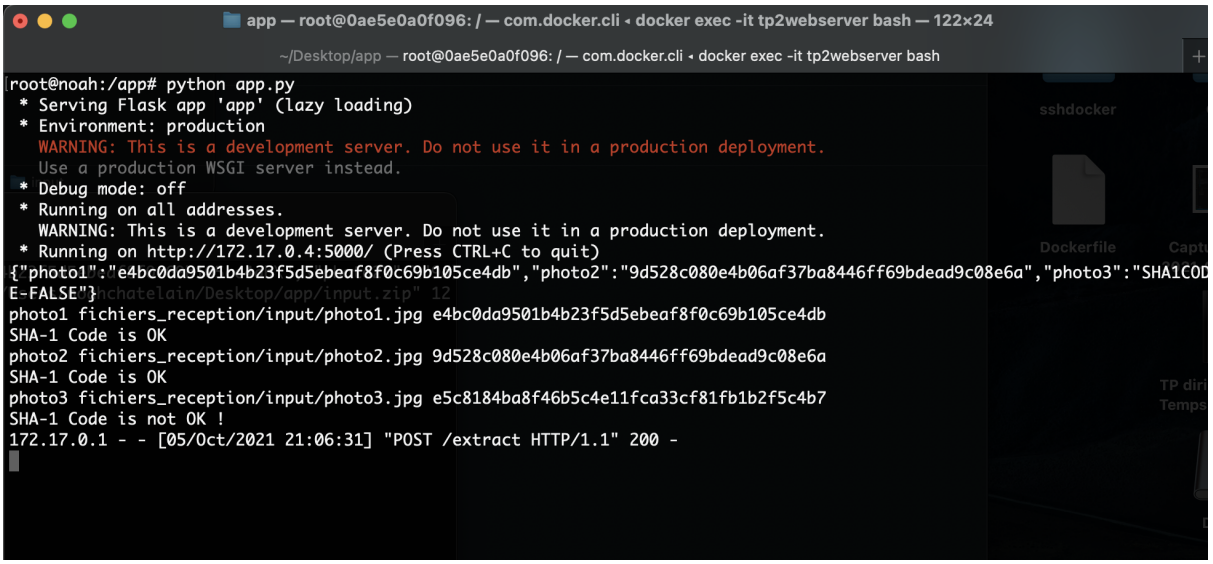
Nous voyons bien ici que nos codes SHA-1 sont identiques à l'étape précédente

curl -X POST -i -F json="{\"photo1\":\"e4bc0da9501b4b23f5d5ebeaf8f0c69b105ce4db\",\"photo2\":\"9d528c080e4b06af37ba8446ff69bdead9c08e6a\",\"photo3\":\"e5c8184ba8f46b5c4e11fca33cf81fb1b2f5c4b7\"}" -F "file=@/Users/noahchatelain/Desktop/app/input.zip" 127.0.0.1:5000/extract



Les vérifications sur les checksums fonctionnent bien. Lorsque l'on teste avec une erreur de code SHA-1 dans le JSON, nous sommes bien avertis :

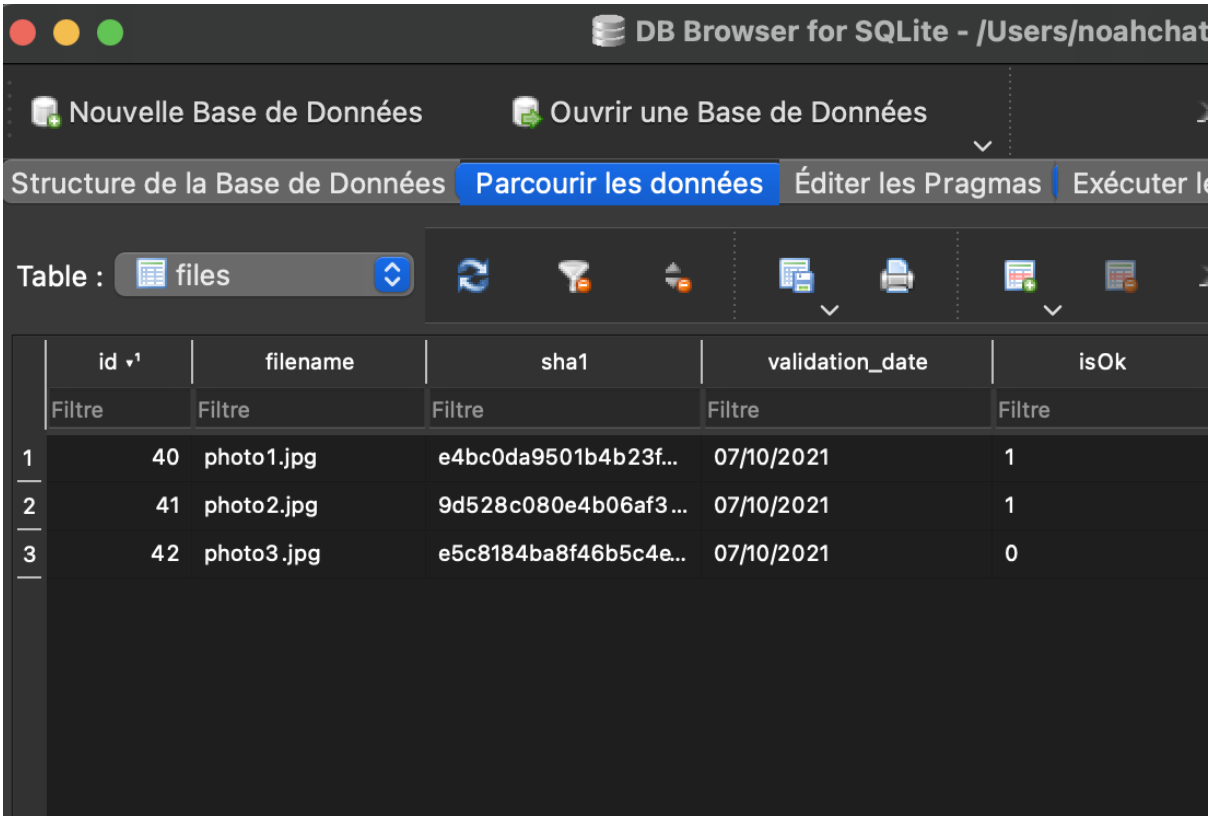




Lorsqu'un fichier possède le bon code SHA-1, alors il est ajouté au dossier */fichiers_reception*, mais lorsqu'il n'est pas bon, il est supprimé du dossier */fichiers_reception* et il est ajouté au fichier zip */log_error_sha1_code_files.zip*

Essayer l'erreur avec cette commande :
curl -X POST -i -F json="{\"photo1\":\"e4bc0da9501b4b23f5d5ebeaf8f0c69b105ce4db\",\"photo2\":\"9d528c080e4b06af37ba8446ff69bdead9c08e6a\",\"photo3\":\"SHA1CODE-FALSE\"}" -F "file=@/Users/noahchatelain/Desktop/app/input.zip" 127.0.0.1:5000/extract

Ajout d'une base de donnée où l'on retrouve toutes les photos, que leur code SHA-1 soit bon ou mauvais. Avec le nom, le code SHA-1, la date de validation, et s'il est valide ou non.



	id	filename	sha1	validation_date	isOk
	Filtre	Filtre	Filtre	Filtre	Filtre
1	40	photo1.jpg	e4bc0da9501b4b23f...	07/10/2021	1
2	41	photo2.jpg	9d528c080e4b06af3...	07/10/2021	1
3	42	photo3.jpg	e5c8184ba8f46b5c4e...	07/10/2021	0

Ajout de la fonctionnalité CSV. Permet d'ajouter nos fichiers à une feuille CSV, exactement comme la base de données.

curl -X POST -i -F json="{\"photo1\":\"e4bc0da9501b4b23f5d5ebeaf8f0c69b105ce4db\",\"photo2\":\"9d528c080e4b06af37ba8446ff69bdead9c08e6a\",\"photo3\":\"SHA1CODE-FALSE\"}" -F "file=@/Users/noahchatelain/Desktop/app/input.zip" 127.0.0.1:5000/extract

files

photo1.jpg	**e4bc0da9501b4b23f5d5ebeaf8f0c69b105ce4db**	**07/10/2021**	**True**
photo2.jpg	9d528c080e4b06af37ba8446ff69bdead9c08e6a	07/10/2021	True
photo3.jpg	e5c8184ba8f46b5c4e11fca33cf81fb1b2f5c4b7	07/10/2021	False

Extrait de la feuille CSV contenant les informations des fichiers

Ajout d'une route permettant de télécharger ce fichier CSV directement pour le client.
http://127.0.0.1:5000/download