

# Captures d'écran

Voici quelques captures d'écran réalisées pendant l'exécution des différentes commandes et requêtes

- Dockerfile

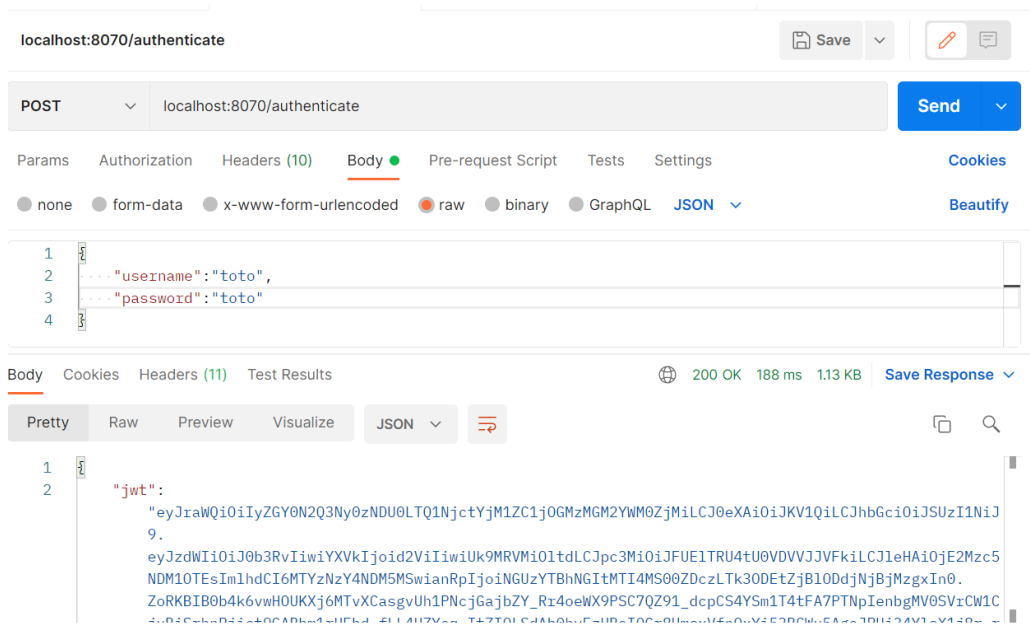
A l'exécution du Dockerfile, la compilation du projet est réalisée. Voici un aperçu de cette compilation et de la fin d'exécution du fichier Dockerfile.

```
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 45.798 s
[INFO] Finished at: 2021-11-23T16:27:09Z
[INFO] -----
Removing intermediate container 6b2414f606ee
---> 9afd54363fe9
Step 6/8 : WORKDIR /episen-ms-security/target/
---> Running in cee8f99f53ad
Removing intermediate container cee8f99f53ad
---> cb9276fa2a45
Step 7/8 : ENTRYPOINT ["java", "-jar", "episen-ms-security-0.0.1-SNAPSHOT.jar"]
---> Running in cb49c1438d24
Removing intermediate container cb49c1438d24
---> d86e878c25a1
Step 8/8 : RUN cd ..
---> Running in 6d21b1814220
Removing intermediate container 6d21b1814220
---> 6323784313fb
Successfully built 6323784313fb
Successfully tagged episen-ms-security:latest
```

- Tests pour l'API.

Le premier test consiste à envoyer les renseignements (nom, mot de passe et rôle si besoin) d'un utilisateur sous une requête POST afin d'obtenir un JWT.

Ci-dessous, le rendu que nous devons obtenir à l'issue de la requête :



Nous voulons ensuite tester si pour un JWT donné, nous avons un utilisateur associé.

Avec le framework Spring Security d'implémenté, toutes les requêtes sont filtrées (mises à part celles pour obtenir un premier JWT cf. ci-dessus avec /authenticate).

Pour cela, nous ajoutons une autorisation avec le JWT obtenu précédemment comme ci-dessous :

✓	Authorization	Bearer eyJraWQiOiIyZGY0N2Q3Ny0zND...
---	---------------	--------------------------------------

Grâce à cela, si le JWT est reconnu et donc autorisé à obtenir les informations souhaitées, alors nous pourrons exécuter correctement notre deuxième requête /hello.

Nous pouvons voir ci-dessous cette exécution et le rendu souhaité « username : toto ».

