Pour instaurer un système d’authentification nous avons ajouté le **Security Bundle**.

Le Security Bundle vous permet de mettre en place différent types d’authentification sur votre application. Il permettra notamment de définir divers espaces. Ils répondent chacun à une logique d’authentification. On nomme ces espaces des firewalls.

Voici différents types d’espace qui sont des firewalls de votre application :

* /Login
* /admin

Avec ces deux exemples ci-dessus qui correspondent à des routes de notre application. On va vérifier si l’utilisateur dispose des droits nécessaires pour accéder à la route. Cela nous permet de sécuriser notre application.

Afin d’installer le **security** bundle nous avons lancer la commande suivante qui est un alias du **security** bundle :

**compose req security**

En fonction de l’utilisation que vous faites du **security** vous allez devoir répondre à quelques questions rapides afin que le bundle soit correctement configuré. Une fois le bundle installé vous allez apercevoir des modifications au niveau du **security.yaml**.

La section encoders : c’est la configuration de l’entité ou des entités pour lesquelles on souhaite encoder le mot de passe.

Les providers : indiquent au composant security où se trouvent les données des utilisateurs.

Les firewalls comme expliqué plus haut ce sont les régions de votre application dont les frontières sont matérialisées par des URLs.

Lorsque vous avez une classe utilisateur il faut y incorporer l’userInterface qui regroupe des méthodes qui nous permettra à l’application de gérer l’encodage et les données d’authentification de l’utilisateur.

Au niveau du controller pour hasher les mots de passe de vos utilisateurs vous pouvez utiliser **l’UserPasswordEncoderInterface.**

Pour gérer l’ensemble du système d’authentification il a fallut créer une classe authenticator.