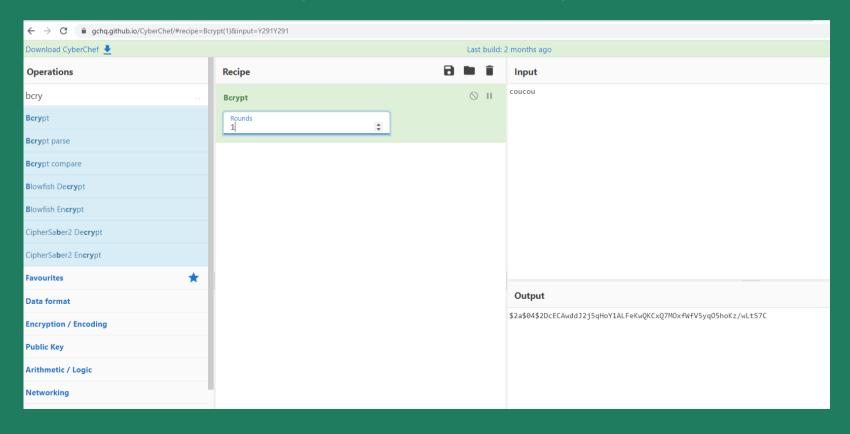
Nous allons reprendre la configuration elytron mais pour maintenant utiliser des mots de passe hashés en BCRYPT à la place du clearText.

Pour cela on va devoir stocker nos utilisateurs avec un mot de passe hashé, on va devoir donc comprendre comment l'algorithme BCRYPT fonctionne.

En plus de la transformation classique d'un algorithme de hashage, bcrypt repète son processus un certains nombre de fois (round ou itération)



De plus l'implémentation d'elytron rajoute un sel (un mot rajouté à chaque mot de passe avant ou après permettant d'éviter les attaques rainbow table).

Nous devons donc avoir 3 informations pour pouvoir générer notre hash:

-Un mot de passe (différent pour chaque utilisateur)

-Un salt (de préférence différent pour chaque utilisateur)

-Un nombre d'itération

Le mot de passe est choisi lors de la création de l'utilisateur Pour le sel nous allons laissé la librairie d'elytron s'en chargé, mais on pourrait prendre le nom de l'utilisateur accolé à un chiffre par exemple.

On choisira 31 itérations (plus le nombre d'itérations est grand, plus le hash est sécurisé, mais plus l'authentification prendra du temps,l'implementation bcrypt de base d'elytron n'autorise de 31 itérations max).

Le hash et le salt seront stocké encodé en base64.

En utilisant la doc officielle, nous allons créer un endpoint pour ajouter un admin à notre base de donnée, avant de mettre en place bcrypt

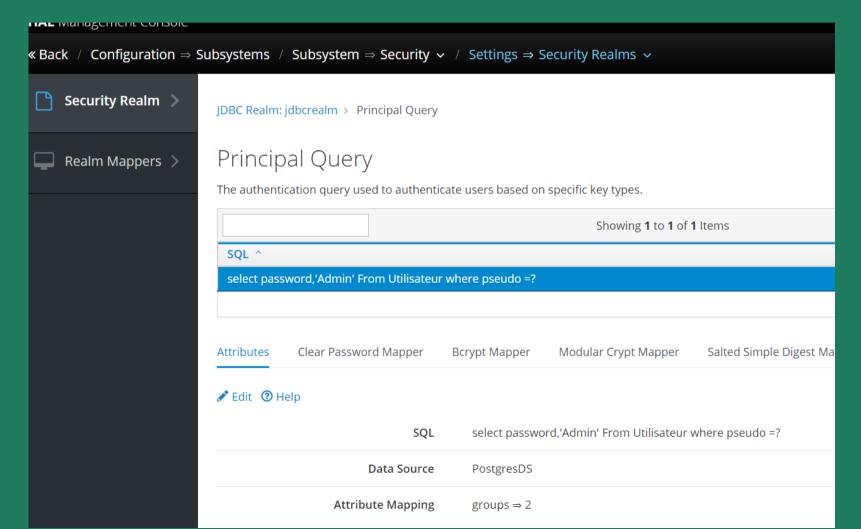
https://github.com/wildfly/wildfly/blob/main/docs/src/main/asciid oc/_elytron/components/JDBC_Security_Realm.adoc#using-ahashed-password-representation

```
1 @GET
@Path("/test")
public void test(){
         PasswordFactory passwordFactory = null;
                passwordFactory = PasswordFactory.getInstance(BCryptPassword.ALGORITHM BCRYPT, ELYTRON PROVIDER);
        } catch (NoSuchAlgorithmException e) {
                e.printStackTrace();
                String mdp="Lepetitchatestmort";
            int iterationCount = 31;
            byte[] salt = new byte[BCryptPassword.BCRYPT SALT SIZE];
            SecureRandom random = new SecureRandom();
            random.nextBytes(salt);
            IteratedSaltedPasswordAlgorithmSpec iteratedAlgorithmSpec = new IteratedSaltedPasswordAlgorithmSpec(iteratio
            EncryptablePasswordSpec encryptableSpec = new EncryptablePasswordSpec(mdp.toCharArray(), iteratedAlgorithmSp
            BCryptPassword original = null;
                        original = (BCryptPassword) passwordFactory.generatePassword(encryptableSpec);
                } catch (InvalidKeySpecException e) {
                        e.printStackTrace();
            byte[] hash = original.getHash();
            Encoder encoder = Base64.getEncoder();
            System.out.println("Encoded Salt = " + encoder.encodeToString(salt));
            System.out.println("Encoded Hash = " + encoder.encodeToString(hash));
            Utilisateur u=new Utilisateur();
            u.setPseudo("seb2");
            u.setSalt(encoder.encodeToString(salt));
            u.setPassword(encoder.encodeToString(hash));
            em.persist(u);
```

Le endpoint /test créé un utilisateur seb2 avec le hash bcrypt et le salt

			-					
		date_naissan timestamp wi		nom character vai	password character varying(255)	prenom character vai	pseudo character vai	salt character varying(255)
1	1				BM4Spphke/urnhp3SdgRZVDx/P3Kpjk=		seb2	J83ucoX407JCDzqSqDrK0Q==

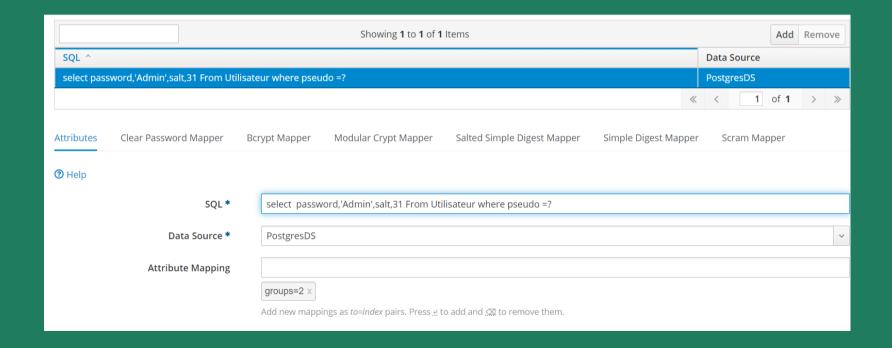
Nous allons maintenant modifier le JDBCRealm pour passer du ClearText au BCRYPT: Security->SecurityRealm->JDBC Realm->Selection de notre realm



Remove sur le clear password mapper

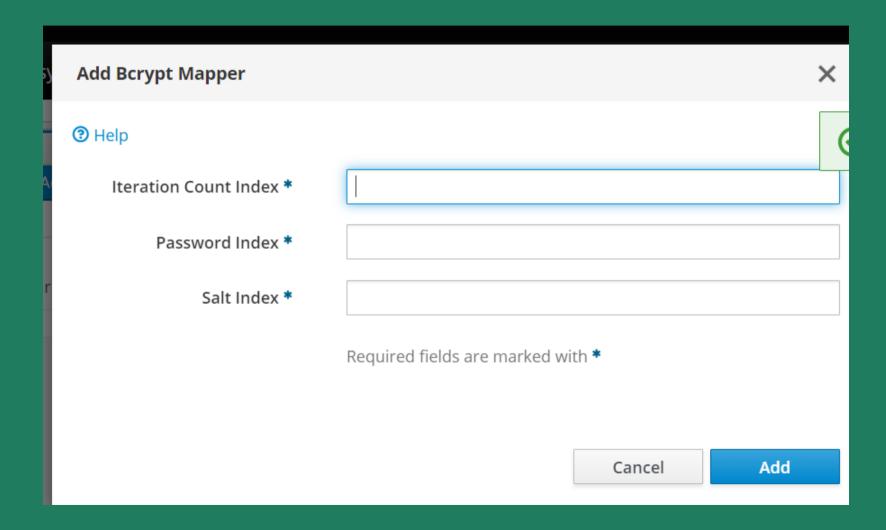
The authenti	ication query used to authen	ticate users based or	n specific						
SQL ^									
select password,'Admin' From Utilisateur where pseudo =?									
Attributes	Clear Password Mapper	Bcrypt Mapper	Modul						
♂ Edit '' Re	eset 👕 Remove 🔞 Help								
	Password Index	1							

Nous modifions la principal query en rajoutant le salt, et le nombre d'itération (ici en constante). Encore une fois ces données pourraient être dans des colonnes ou venant de jointures d'autres tables

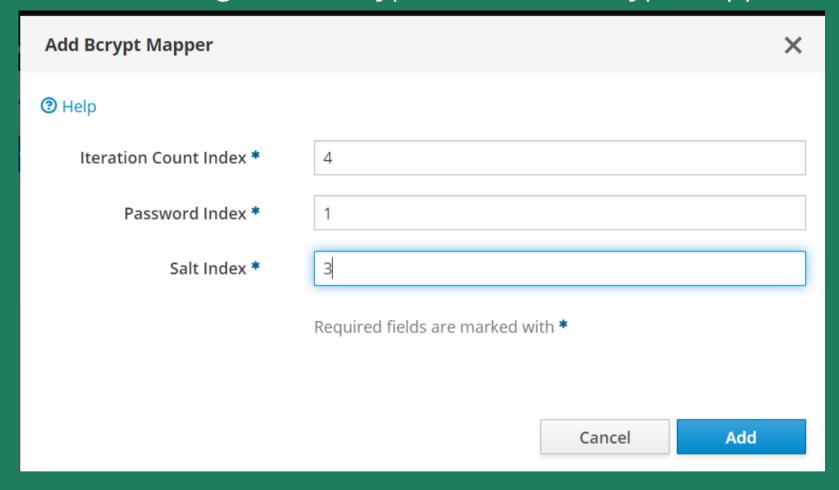


Nous notons que le salt est la 3eme colonne, et le nombre d'itération (31)

On va configurer le bcrypt: Add sur le bcrypt mapper



On va configurer le bcrypt: Add sur le bcrypt mapper



Dans notre principal query: password est colonne 1, salt est colonne 3 et le nombre d'itération colonne 4

Une fois la configuration terminée on redémarre Wildfly. Et Voilà!