

# LU08d - Security

Spring Boot enthält praktischerweise auch einfachere Arten, um mit Passwörtern umzugehen.

## BCryptPasswordEncoder

Beim Aufruf der Methode `encode` wird jedes Mal ein neuer zufälliger Salt generiert und verwendet. Dieser Salt wird zusammen mit dem eigentlichen Hash ausgegeben im Format: `$2a$10$[22-Zeichen-Salt][31-Zeichen-Hash]`

```
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

...
BCryptPasswordEncoder encoder = new BCryptPasswordEncoder();
String passwordHash = encoder.encode(password);
```

Wird zu einem späteren Zeitpunkt der Hash überprüft, wird entsprechend der korrekte Salt gelesen und verwendet.

```
encoder.matches(plainPassword, passwordHash)
```

Dies hat den vorteilhaften Nebeneffekt, dass der Salt nicht mehr separat abgespeichert werden muss.

## Secrets

Damit die Credentials für die Datenbank nicht im Repository ersichtlich sind, können diese in der Datei `application.properties` unter `src/main/resources/application.properties` abgelegt werden. Spring Boot erkennt diese Datei und stellt die Werte zur Verfügung.

```
server.port=8080
db.username=localuser
db.password=
jwt.secret=
```

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu08/d?rev=1761648752>

Last update: **2025/10/28 11:52**

