[**https://stackoverflow.com/questions/71179660/spring-boot-3-jakarta-and-javax**](https://stackoverflow.com/questions/71179660/spring-boot-3-jakarta-and-javax)

**Từ Spring Boot ver 3.x trở đi, sẽ dùng thư viện jakarta của Jakarta EE 9 thay vì javax. Và sẽ dùng Java 17.**

**-> khi import các thư viện vẫn xài thư viện của javax, thì down parent version của Spring Boot xuống ver 2.x.**

**Hoặc đơn giản hơn là cứ vào luôn solution copy file pom.xml cho nhanh.**

1. **Authentication? Authorization**
2. **Encoding? Hashing? Salting?**
   * **Encoding** là mã hoá 1 chuỗi kí tự nhưng có thể dịch ngược lại được.
   * **Hashing** là mã hoá 1 chuỗi kí tự nhưng không thể dịch ngược lại được hoặc rất rất khó có thể dịch ngược lại.
     + Thuật toán hashing ko hoàn hảo, chúng vẫn có tỉ lệ hash ra 2 output giống nhau từ 2 input ban đầu khác nhau. Việc hash 2 input ban đầu khác nhau ra 2 output giống nhau đó gọi là **collision**
     + Điều này sẽ sinh ra vấn đề khi 2 user có password giống nhau sẽ hash ra cùng 1 output giống nhau. Điều này là nên tránh vì khi ta lưu password trên hệ thống, mọi hashing value nên là unique (hashing value rất khó có thể dịch ngược lại nhưng ko có nghĩa là ko thể). **Salt** sinh ra để khắc phục vấn đề này.
   * **Salting** là hành động thêm 1 chuỗi kí tự bất kì vào cuối 1 password cần được mã hoá trước khi hashing. Với việc ta định nghĩa trước mỗi user sẽ có một **salt** (còn gọi là secret key) riêng biệt, ta sẽ đảm bảo rằng password của từng user nếu kết hợp với salt sẽ luôn hash ra chuỗi ko trùng nhau. Việc này sẽ làm tăng độ phức tạp của hashing value nên rất nhiều lần và việc dịch ngược lại gần như là ko thể.