SonarQube && Docker

ติดตั้ง docker และเข้าสู่ระบบ

ลิงค์เข้าสู่การติดตั้งสำหรับ windows 10 home

<https://medium.com/@komsan7410/%E0%B8%95%E0%B8%B4%E0%B8%94%E0%B8%95%E0%B8%B1%E0%B9%89%E0%B8%87-docker-desktop-%E0%B8%9A%E0%B8%99-windows-10-home-x64-3a91cc204783>

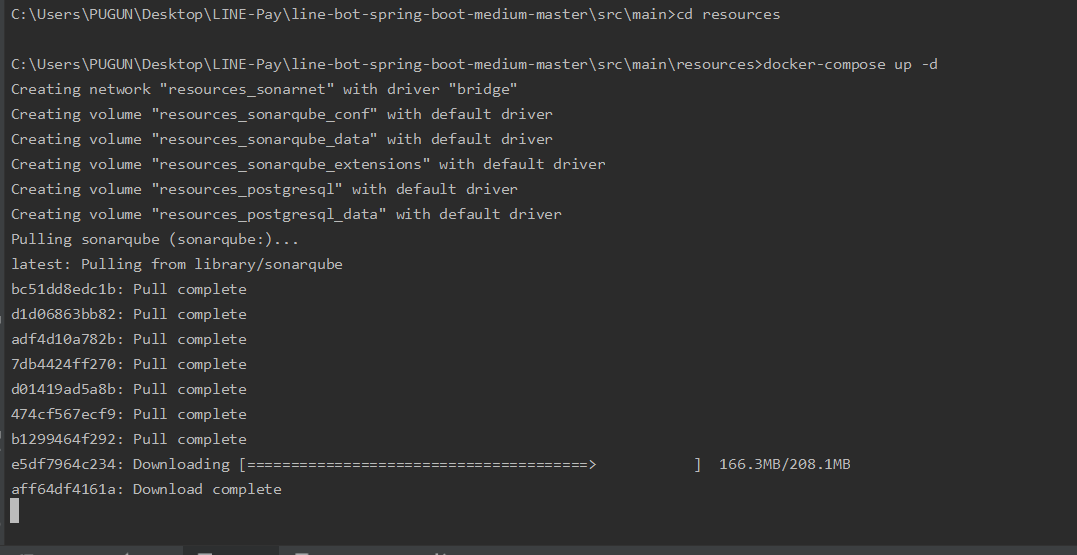
ลิงค์สมัคร docker <https://hub.docker.com/>

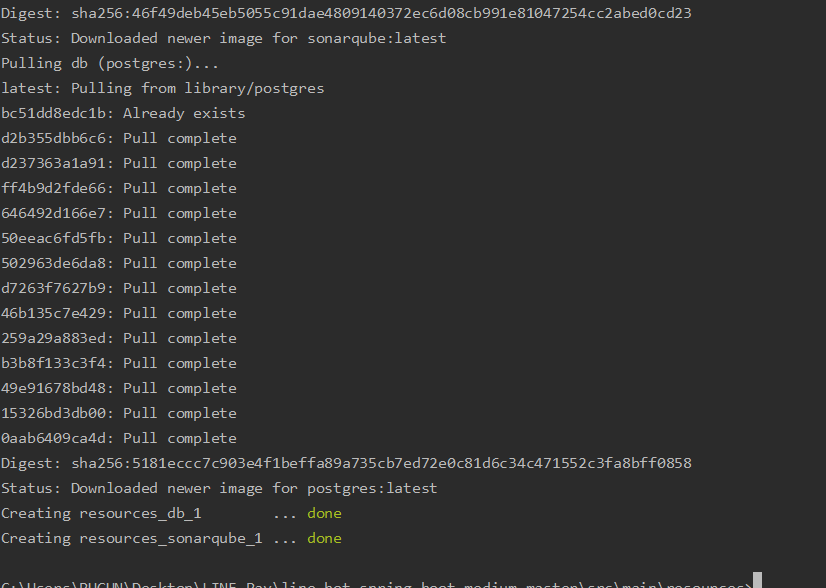
เมื่อทำการติดตั้งและเข้าสู่ระบบ docker ได้แล้ว เปิดโปรเจคที่ตั้งการตรวจสอบขึ้นมา สร้างไฟล์ docker-compose.yml เพื่อ Install SonarQube และใส่โค้ดดังนี้

version: "2"  
  
services:  
 sonarqube:  
 image: sonarqube  
 ports:  
 - "9000:9000"  
 networks:  
 - sonarnet  
 environment:  
 - sonar.jdbc.url=jdbc:postgresql://db:5432/sonar  
 volumes:  
 - sonarqube\_conf:/opt/sonarqube/conf  
 - sonarqube\_data:/opt/sonarqube/data  
 - sonarqube\_extensions:/opt/sonarqube/extensions  
  
 db:  
 image: postgres  
 networks:  
 - sonarnet  
 environment:  
 - POSTGRES\_USER=sonar  
 - POSTGRES\_PASSWORD=sonar  
 volumes:  
 - postgresql:/var/lib/postgresql  
 *# This needs explicit mapping due to https://github.com/docker-library/postgres/blob/4e48e3228a30763913ece952c611e5e9b95c8759/Dockerfile.template#L52* - postgresql\_data:/var/lib/postgresql/data  
  
networks:  
 sonarnet:  
 driver: bridge  
  
volumes:  
 sonarqube\_conf:  
 sonarqube\_data:  
 sonarqube\_extensions:  
 postgresql:  
 postgresql\_data:

จากนั้นทำการเข้าถึงที่อยู่ของไฟล์ docker-compose.yml แล้วใช้คำสั่งรัน docker-compose up -d

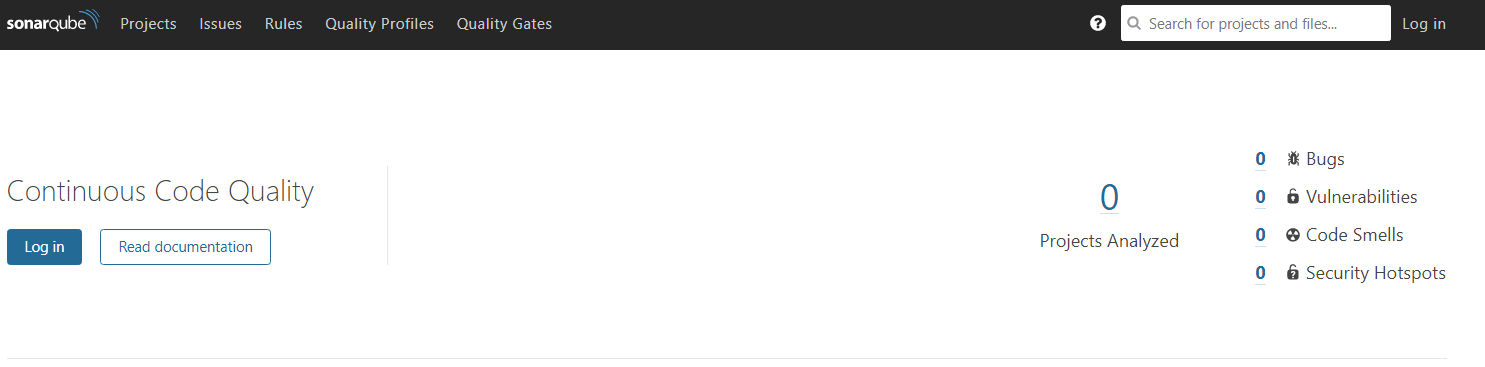
โปรแกรมจะทำการติดตั้ง



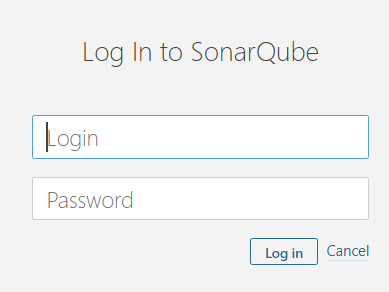


จะเห็นได้ว่ามีการสร้าง sonarqube\_db\_1 ที่เป็น DB และ sonarqube\_sonarqube\_1 ที่เป็น SonarQube สามารถลองใช้คำสั่ง docker ps เพื่อดูว่า docker container ของ 2 service นี่ทำงานอยู่หรือไม่

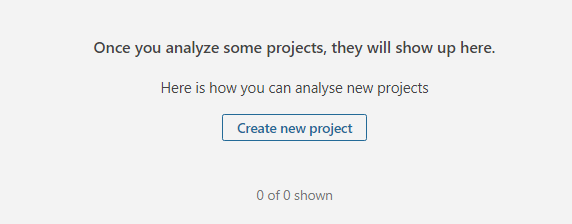
สามารถเข้าถึง SonarQube ด้วย URL <http://localhost:9000/about> ก็จะเห็นว่ายังไม่มี project อะไรที่สร้างขึ้น

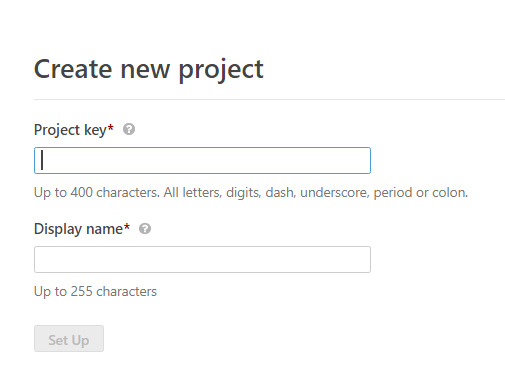


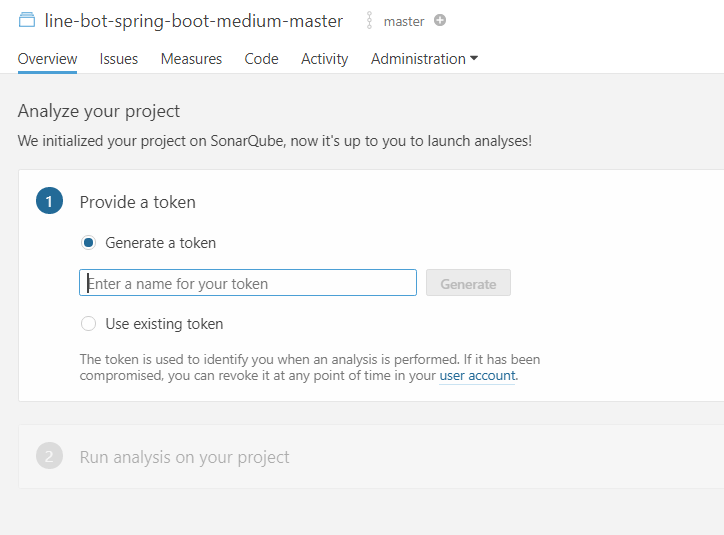
Login เข้าสู่ระบบ SonarQube ด้วย username และ password ตั้งต้นเป็น admin

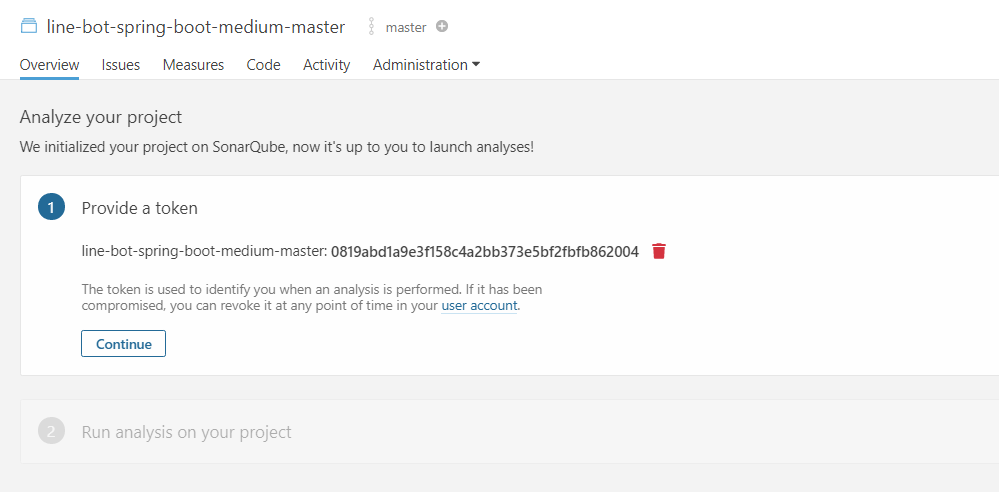


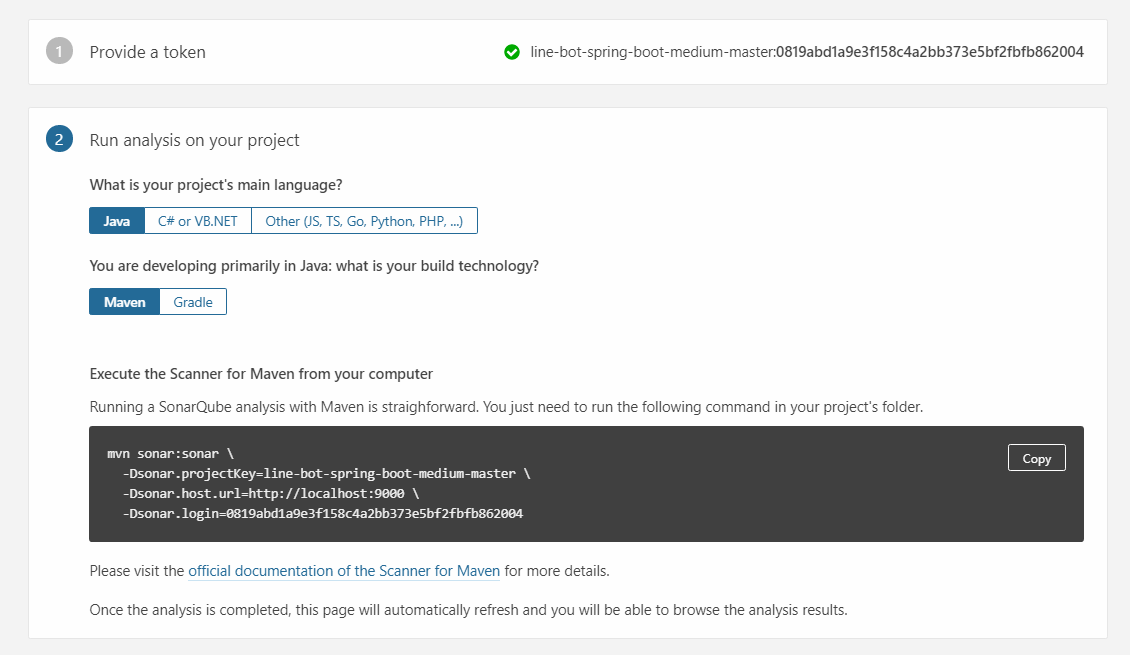
และสร้าง project ขึ้นมาใหม่ โดยทั้งชื่อให้กับ project











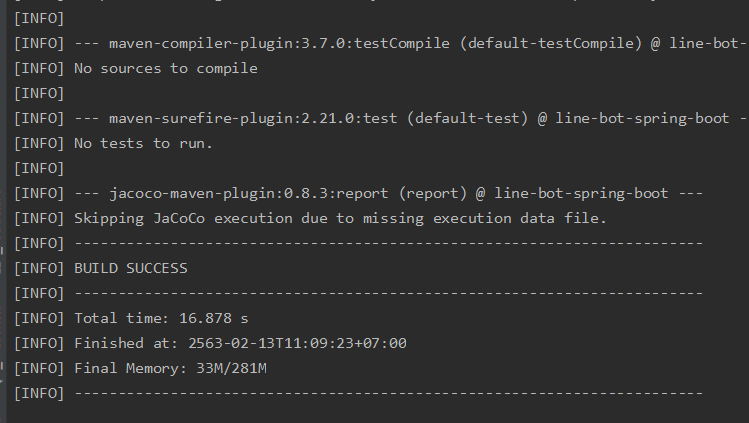
Install Jacoco

เนื่องจากว่า test coverage ของ JUnit นั้นทำงานได้ไม่ดีนักกับ SonarQube ดังนั้นจะใช้ plugin ที่ชื่อว่า Jacoco มาช่วยสร้าง test coverage ให้กับ Spring Boot project ด้วยการเพิ่ม plugin ของ Jacoco ในpom.xml ดังนี้

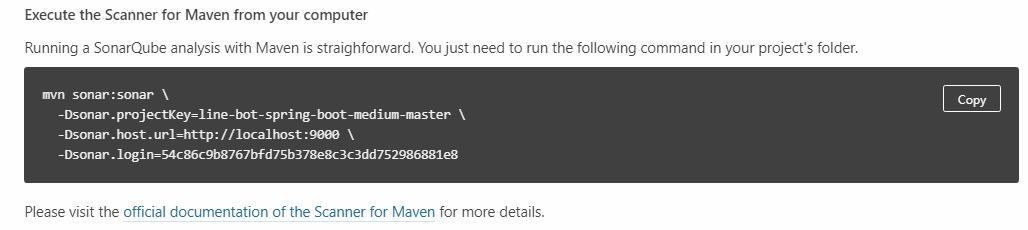
<plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
</plugin>  
<plugin>  
 <groupId>org.jacoco</groupId>  
 <artifactId>jacoco-maven-plugin</artifactId>  
 <version>0.8.3</version>  
 <executions>  
 <execution>  
 <goals>  
 <goal>prepare-agent</goal>  
 </goals>  
 </execution>  
 <!-- attached to Maven test phase -->  
 <execution>  
 <id>report</id>  
 <phase>test</phase>  
 <goals>  
 <goal>report</goal>  
 </goals>  
 </execution>  
 </executions>  
</plugin>

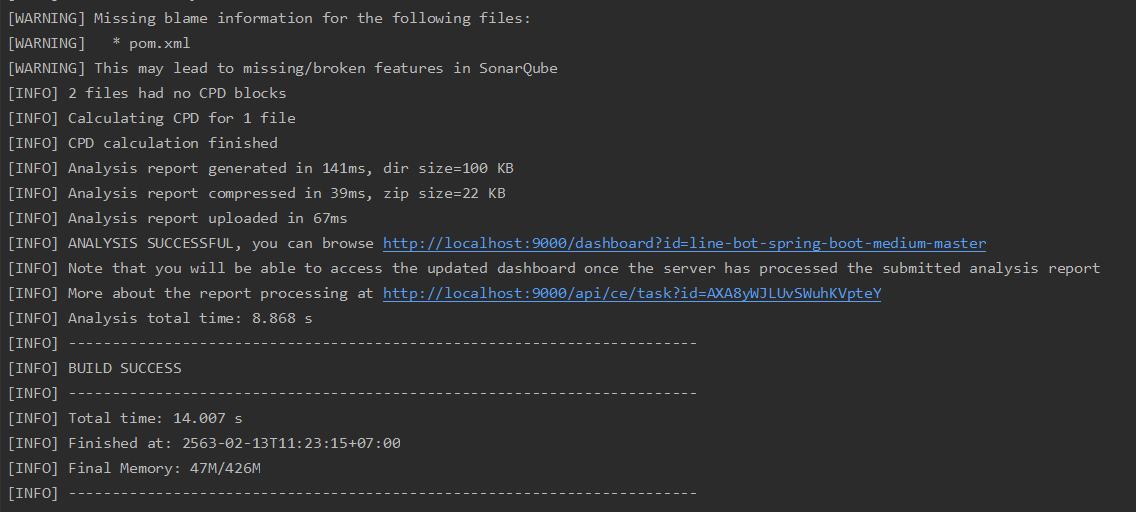
Run SonarQube

ใช้คำสั่ง mvn test เพื่อรัน test ใน project ก็จะได้ test result และ test coverage

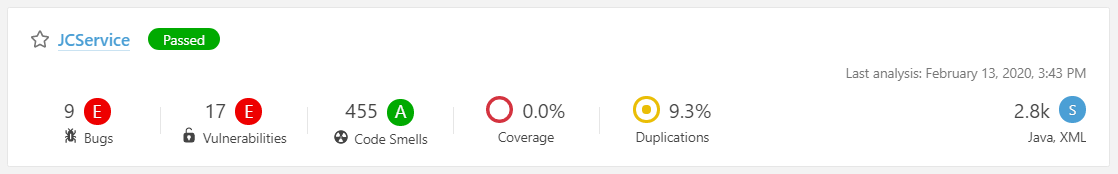


\*\*\*\*\*\*จากนั้นรัน SonarQube ด้วยคำสั่งที่ได้จากการสร้างโปรเจค เมื่อรันเรียบร้อยจะได้ BUILD SUCCESS

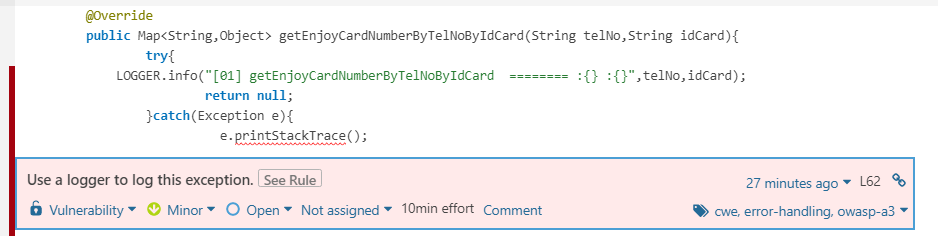




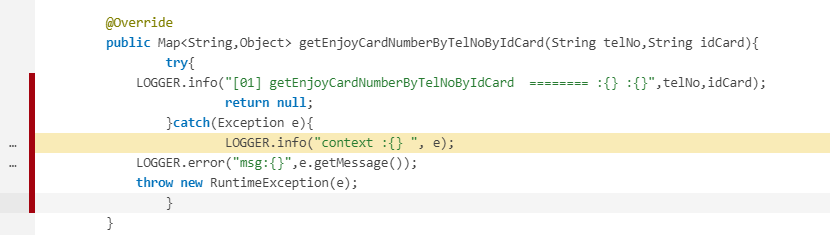
เมื่อรันแล้ว กลับเข้าหน้า SonarQube อีกครั้ง



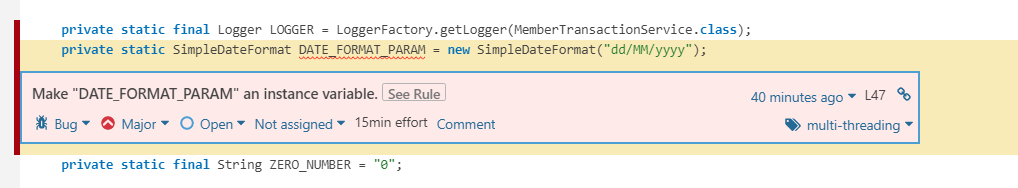
เข้าไปที่โปรเจค เพื่อดูโค้ดที่ต้องแก้



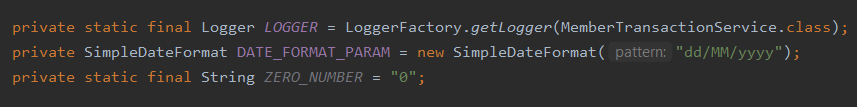
///แก้



private static || private static final



//แก้ ตัด static || static final ออก



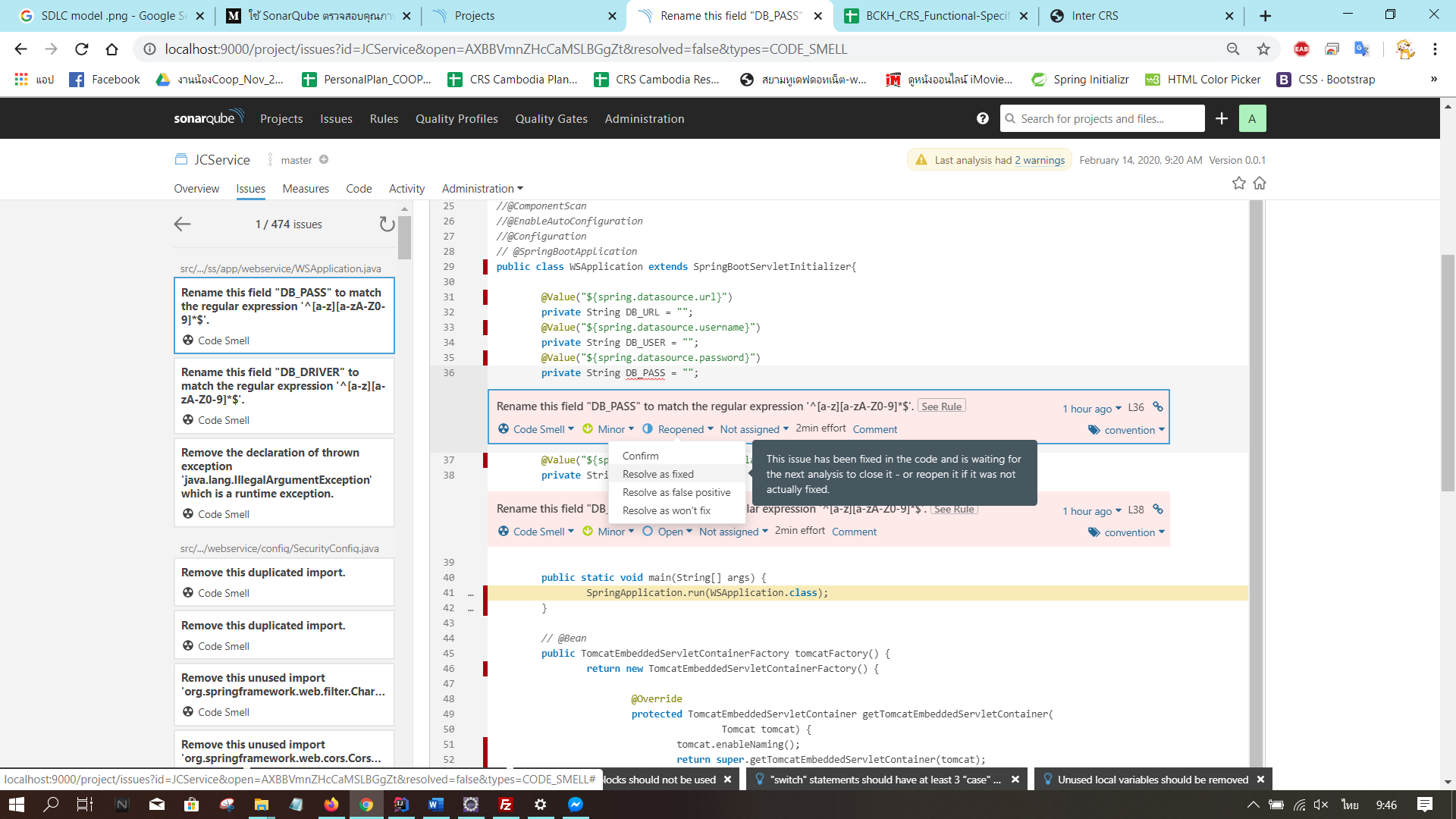


//แก้ ตัด args ออก

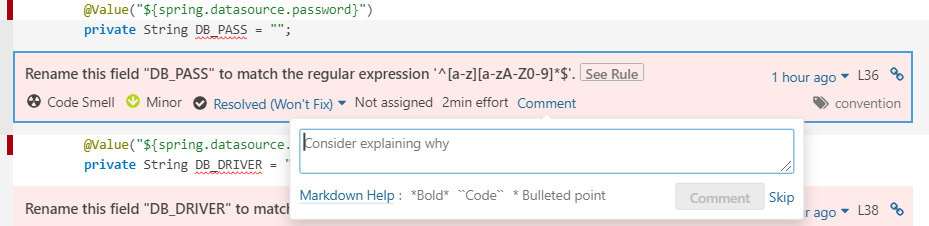


กรณีไม่ต้องการแก้ไขโค้ด

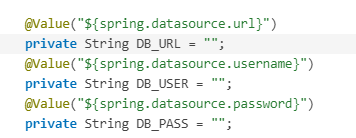
เลือกโค้ด open 🡪 Resolve as won’t fix



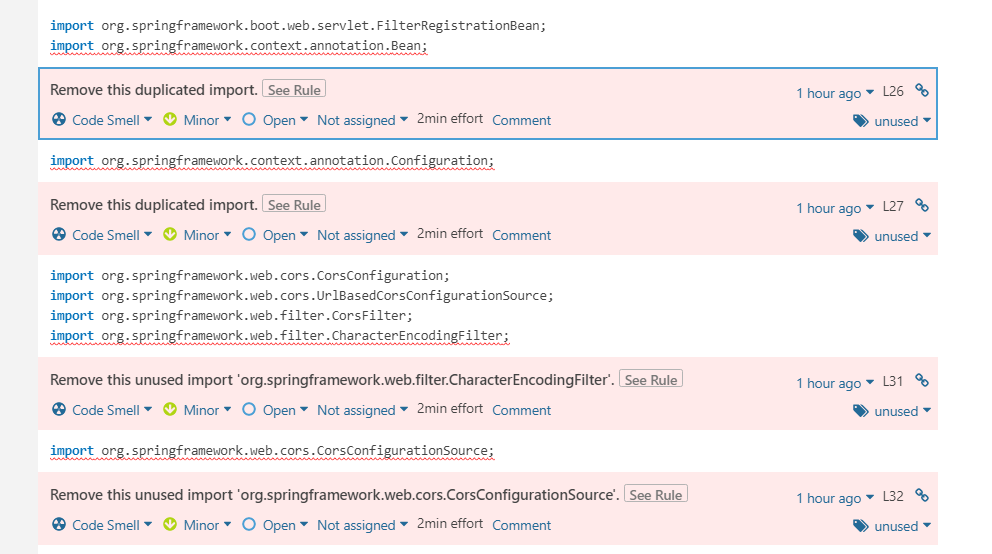
กด skip

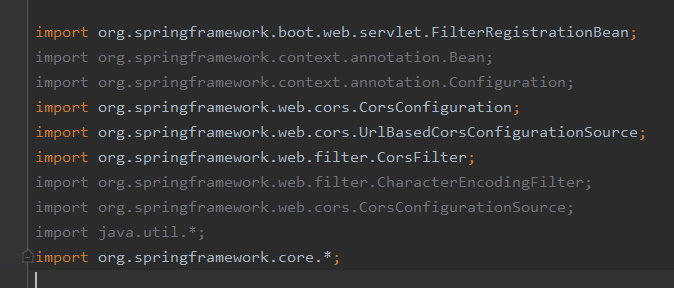


ทำการรีเฟรชหน้าจออีกครั้ง

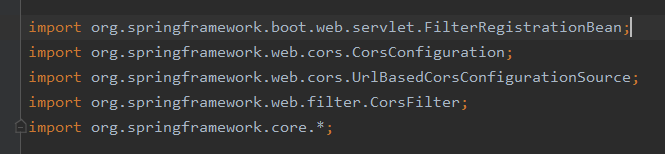


// ลบส่วนที่ไม่ใช้ออก

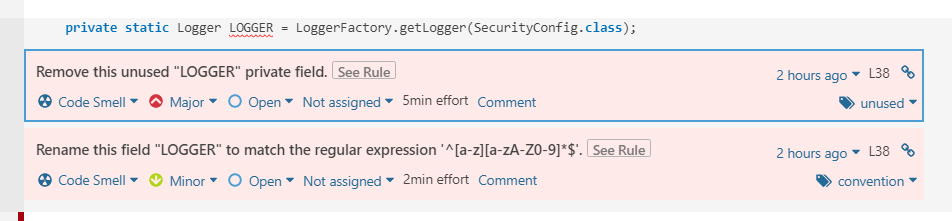




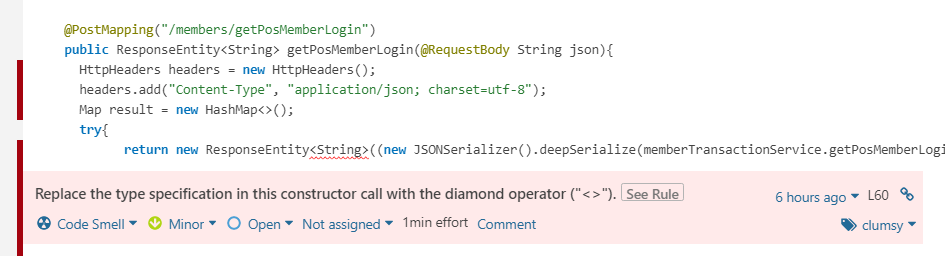
/// แก้



ให้ลบออก || การตั้งชื่อตัวแปรผิดรูปแบบ



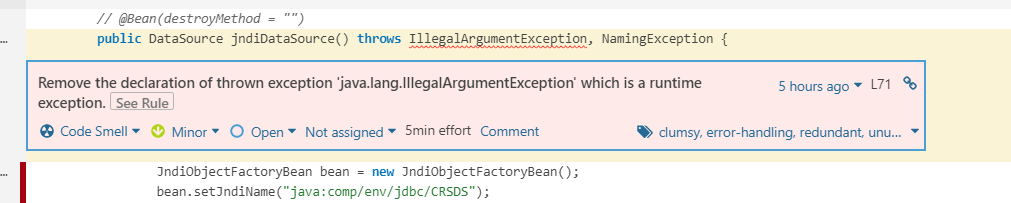
ให้นำออบเจกต์ใน <> ออก



//แก้



เปลี่ยนรูปแบบการเขียน

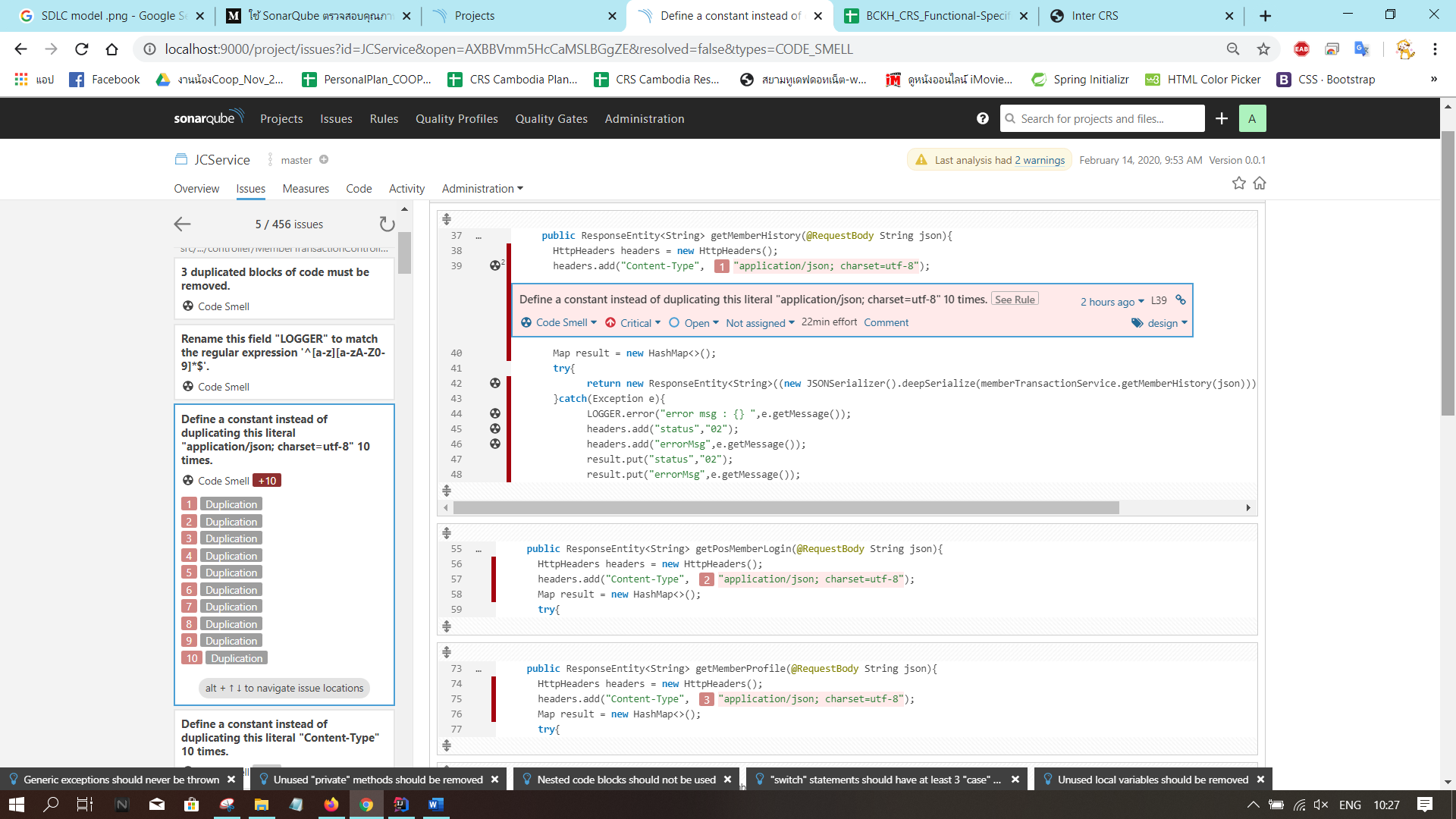




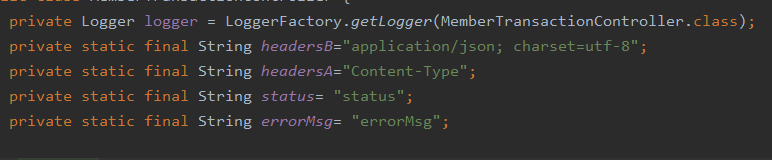
//แก้



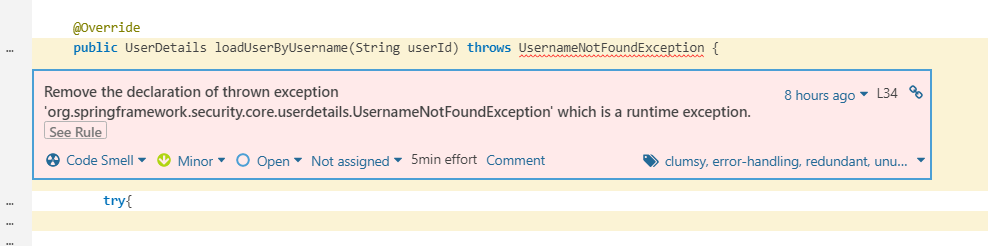
ให้ตั้งตัวแปรขึ้นมาเพื่อเก็บค่า

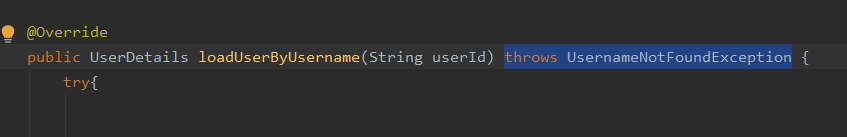


// แก้



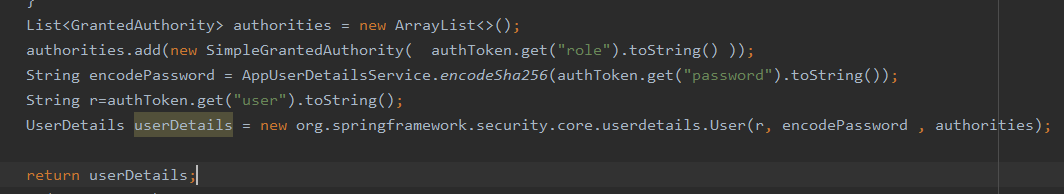
ลบ throws UsernameNotFoundException



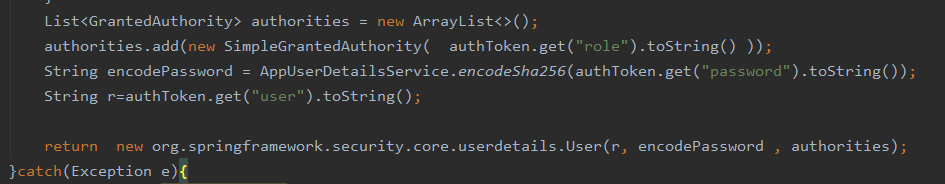


ให้เปลี่ยนการรีเทรินค่า

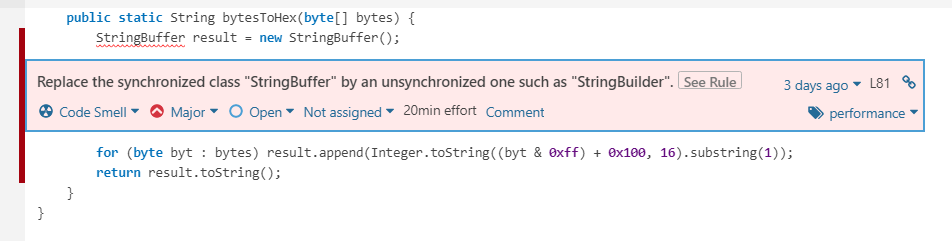


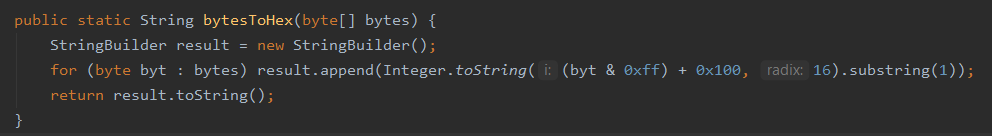


///แก้

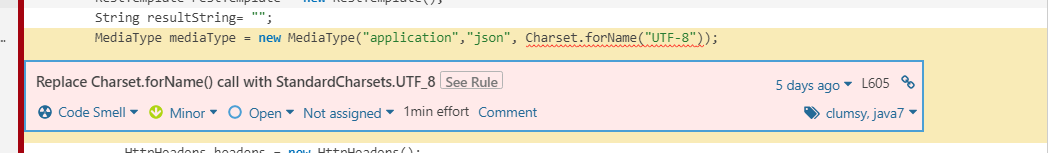


ให้เปลี่ยนไปใช้ StringBuilder แทน





ให้เปลี่ยนรูปแบบการใช้คำสั่ง



///แก้





//แก้

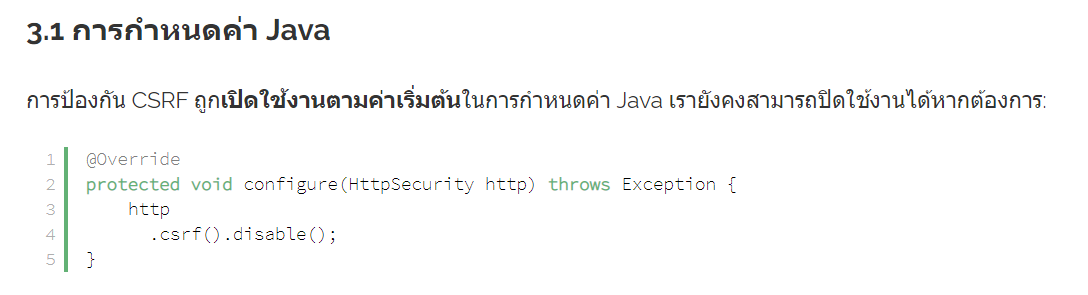
<https://rules.sonarsource.com/java/tag/owasp/RSPEC-5122>



ความหมายของ http.csrf().disable();

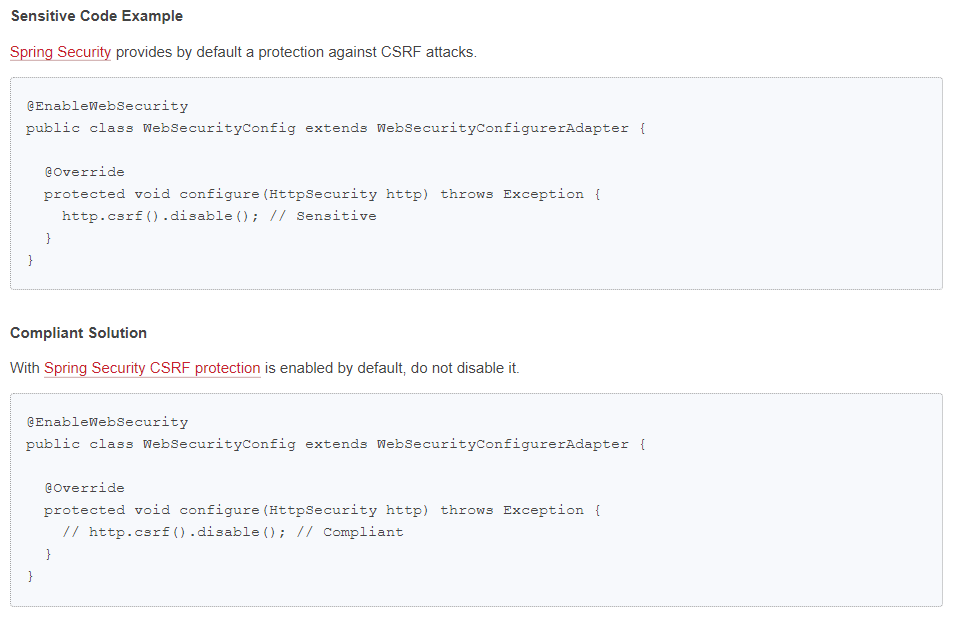
<https://www.baeldung.com/spring-security-csrf>

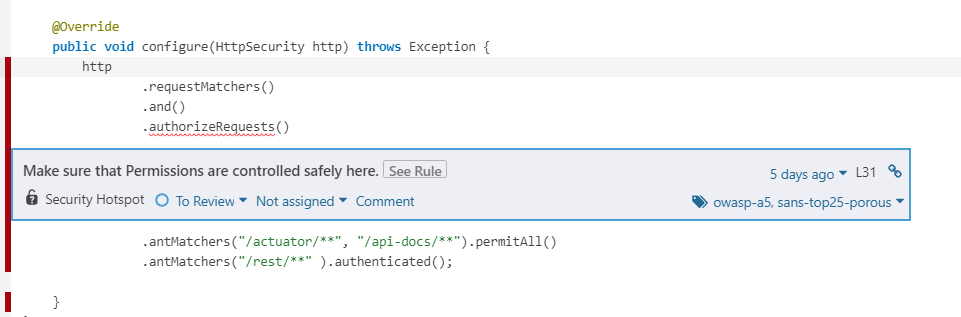




///การแก้ไขจาก sonarsource

<https://rules.sonarsource.com/java/tag/spring/RSPEC-4502>





////แนะนำการแก้ไข

<https://stackoverflow.com/questions/58529328/getting-sonar-critical-defect-on-http-security-configuration-authorizerequests>



