**Užitečné linky**

Registration with Spring Security – Password Encoding: <https://www.baeldung.com/spring-security-registration-password-encoding-bcrypt>

Spring Security guidelines - <https://www.marcobehler.com/guides/spring-security>

REST Exceptions handling - <https://www.toptal.com/java/spring-boot-rest-api-error-handling>

Video: Spring Security: REST security with JWT - <https://www.youtube.com/watch?v=B03UruvOl30>

využiju spring security framework - Security - Spring security dependency...

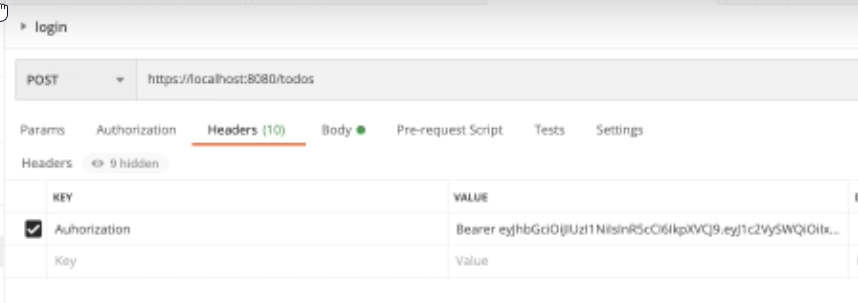
Authentication:

1.registration endpoint - /register - vytvořím novou entitu s registrací

psw zde můžu jako plain text ale ideálně to zakódovat

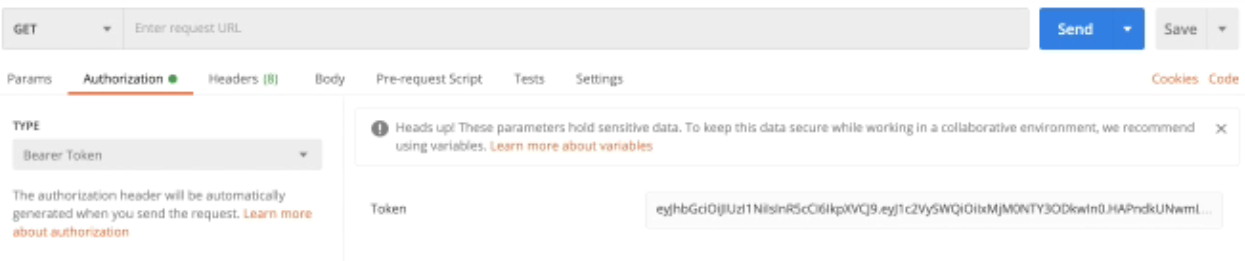
2.login endpoint: username+ psw, jako response bude jwt token (jwt.io) - ten má platnost třeba hodinu.

todle musí uživatel vždy zaslat s get dotazem (token je v Headers):



my to musíme validovat a pak až vrátíme todos

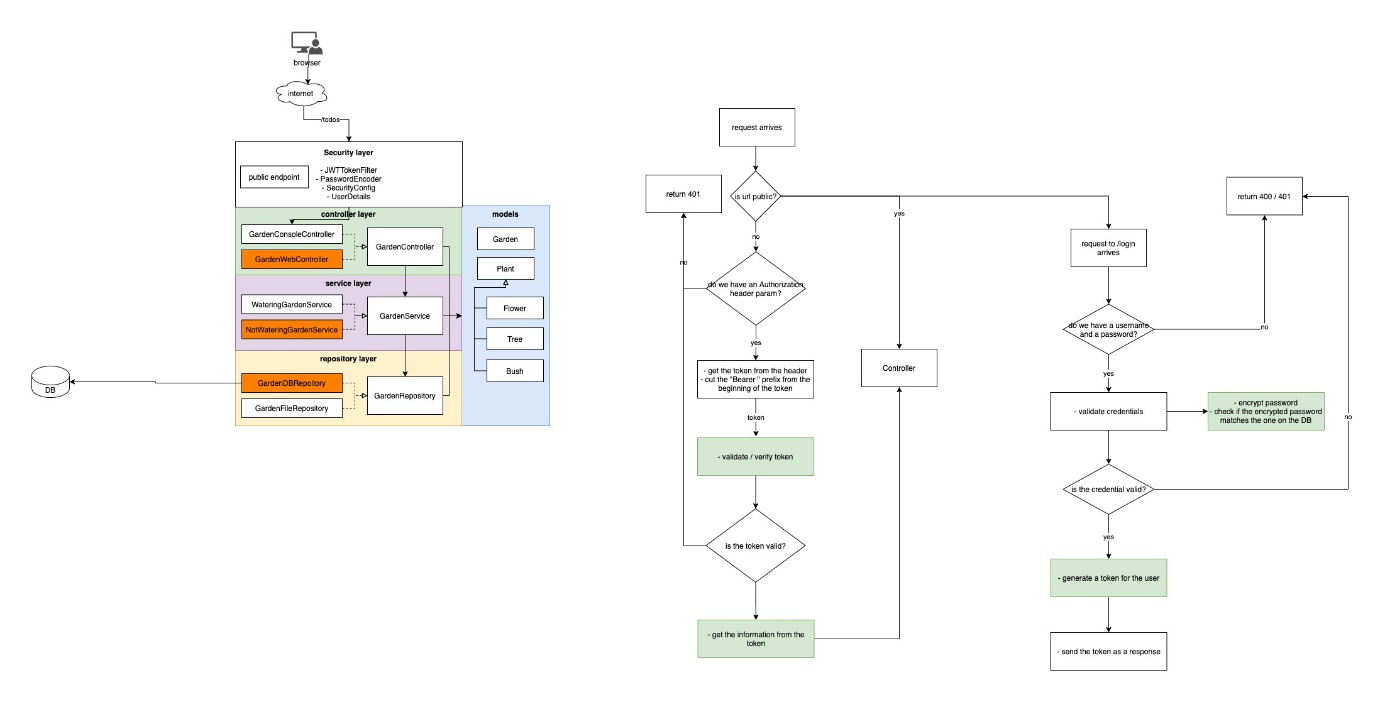
V postmanovi mohu využít část Authorization:

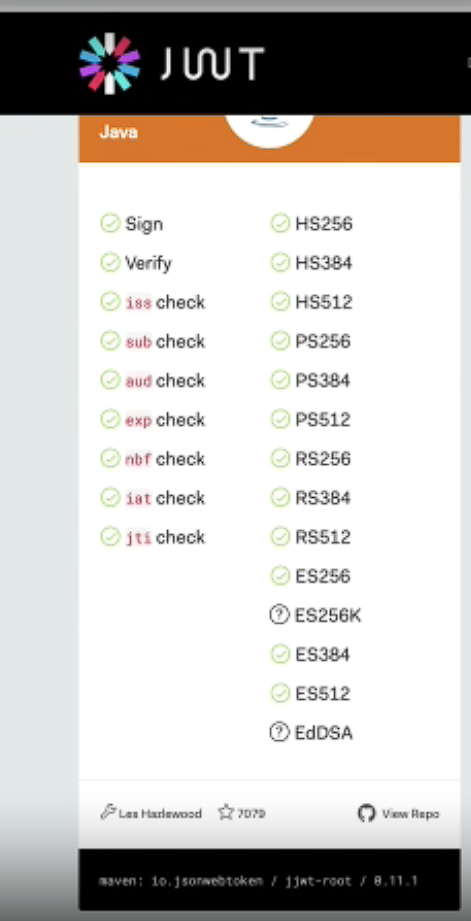


Takže všechny ostatní endpointy kromě login a register musí být dostupné pouze s validním tokenem.

Test Environment:

-inmemory database - použiju H2 database



 Bcrypt pro šifrování hesel

Deprecate JwtParserBuilder#setSigningKey(String)

<https://github.com/jwtk/jjwt/issues/617>

Porovnávání jsonů pomocí jsonassert

<https://www.baeldung.com/jsonassert>

dependency: compile 'org.skyscreamer:jsonassert:1.5.0'