



















For oAuth2 configuration details

<http://www.tinmegali.com/en/2017/06/25/oauth2-using-spring/>

https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html

Springboot Security

1. Add these dependencies

compile "org.springframework.boot:spring-boot-starter-security"

compile "org.springframework.boot:spring-boot-starter-web"

testCompile "org.springframework.boot:spring-boot-starter-test"

1. Simply put, by default, the application will get Basic Authentication enabled.

security.basic.enabled=true

security.user.name=user

security.user.password=password

1. We currently have an application that is secured with one user named user, but what if we wanted more users.

@EnableWebSecurity

@Configuration

public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired

    protected void configureUser(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .inMemoryAuthentication()

                .withUser("sunil").password("pass123").roles("USER")

                .and()

                .withUser("admin").password("admin123").roles("ADMIN");

    }

    // We do not want the default behavior of form authentication

    // before HTTP Basic authentication we get

    // from WebSecurityConfigurerAdapter.

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http.authorizeRequests()

                .anyRequest().fullyAuthenticated()

                .and()

                .httpBasic();

    }

}

**Splunk or Paper Trail  log drain**

1) Open Paper Trail

<https://papertrailapp.com/>

2) Enter an email and password

   Click Subscribe

   Click Setup a new system

   Capture the host and port at the top of the page.

For example: logs5.papertrailapp.com:13071

3) Create a  CUPS (create user provided service)

cf create-user-provided-service paper-trail -l syslog-tls://logs5.papertrailapp.com:13071

cf bind-service spring-music paper-trail

cf restage spring-music

Pcf autoscaler

1) cf marketplace

2) Create a autoscaler service instance.

cf create-service app-autoscaler standard autoscaler

3) Bind the service to articulate.

cf bind-service spring-music autoscaler

4) Restart the application.

cf restart spring-music

Since you have limited resources of 2GB it might fail

Default policy is 2 minimum instance and 5 max instance

You can change the minimum and maximum instance

<https://akrabat.com/custom-urls-for-a-cloud-foundry-application/>

cf --help

cf push

cf domains

cf routes

cf stop spring-music

cf create-domain tanushreeghosh tanushree.io

cf orgs

cf restage spring-music

cf delete-orphaned-routes

cf target

cf unmap-route sony-kasala.cfapps.io

**No Gradle or maven installed in the system**

Tried to run using gradlew within the project folder

D:\S2-Course\spring-music>gradlew clean build

Downloading <https://services.gradle.org/distributions/gradle-4.4-bin.zip>

Exception in thread "main" java.net.ConnectException: Connection refused: connect

        at java.net.DualStackPlainSocketImpl.connect0(Native Method)

        at java.net.DualStackPlainSocketImpl.socketConnect(Unknown Source)

**Add the below 2 properties in gradle.properties file within the project folder**

**systemProp.http.proxyHost**=**proxy usl which in ie browser**

**systemProp.http.proxyPort**= **port number**

**systemProp.https.proxyHost**= **proxy usl which in ie browser**

**systemProp.https.proxyPort**=**port number**

D:\S2-Course\spring-music>gradlew clean build

Sample url for uploading a file <http://www.technicalkeeda.com/spring-tutorials/spring-mvc-file-upload-restful-web-service-example>

<https://akrabat.com/custom-urls-for-a-cloud-foundry-application/>

1) Spring boot sample github url <https://github.com/scottfrederick/spring-music>

2) Slueth + Zipkin <https://dzone.com/articles/tracing-in-microservices-with-spring-cloud-sleuth>

For spring boot here is the url you can refer <https://www.baeldung.com/spring-boot>

 