Create a Spring Boot Application to authenticate users with LDAP

1. Copy the previous project **spring-boot-saml-demo** and RENAME it as

spring-boot-ldap-demo

1.1 Change **groupId** & **artifactId** of the project in **pom.xml**

```
<modelVersion>4.0.0</modelVersion>
    <groupId>com.springboot.ldap</groupId>
    <artifactId>spring-boot-ldap-demo</
artifactId>
    <version>0.0.1-SNAPSHOT</version>
```

Integrating LDAP for Authentication

Close all OPEN Files

20. Collapse the earlier project **spring-boot-saml-demo**

Start Making changes to **spring-boot-ldap-demo**

21. Add LDAP dependencies in **pom.xml**

```
<!-- LDAP DEPENDENCIES -->
<dependency>
     <groupId>org.springframework</groupId>
     <artifactId>spring-tx</artifactId>
</dependency>
<dependency>
     <groupId>org.springframework.ldap</groupId>
     <artifactId>spring-ldap-core</artifactId>
</dependency>
<dependency>
     <groupId>org.springframework.security</groupId>
     <artifactId>spring-security-ldap</artifactId>
</dependency>
<dependency>
     <groupId>com.unboundid</groupId>
     <artifactId>unboundid-ldapsdk</artifactId>
</dependency>
```

23. Navigate to your project (in the `src/main/resources` folder).

Create a folder named `ldap`

24. Create a file **`ldapschema.ldif`** in **`ldap`** folder.

Right Click -> ldap folder
-> New
-> File
-> File name as **“ldapschema.ldif”**

LDAP Server Setup

We are going to use the **LDAP Data Interchange Format** (**LDIF**) to set up our users on our LDAP server.

The LDIF is a standard text-based representation for LDAP data, and changes to that data

`Add the following LDAP data in` **`ldapschema.ldif`**

```
dn: dc=packtpub,dc=com
objectclass: top
objectclass: domain
objectclass: extensibleObject
dc: packtpub

dn: ou=groups,dc=packtpub,dc=com
objectclass: top
objectclass: organizationalUnit
ou: groups

dn: ou=people,dc=packtpub,dc=com
objectclass: top
objectclass: organizationalUnit
ou: people

dn: uid=john,ou=people,dc=packtpub,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Tomcy John
uid: tjohn
userPassword: tjohn@password

dn: cn=admins,ou=groups,dc=packtpub,dc=com
objectclass: top
objectclass: groupOfUniqueNames
cn: admins
ou: admin
uniqueMember: uid=tjohn,ou=people,dc=packtpub,dc=com

dn: cn=users,ou=groups,dc=packtpub,dc=com
objectclass: top
objectclass: groupOfUniqueNames
cn: users
ou: user
uniqueMember: uid=tjohn,ou=people,dc=packtpub,dc=com
```

Change the **application.yml** for LDAP configuration

```
spring:
  ldap:
   # Embedded Spring LDAP
   embedded:
     base-dn: dc=packtpub,dc=com
     credential:
       username: uid=admin
       password: secret
     ldif: classpath:ldap/ldapschema.ldif
     port: 8389
     validation:
       enabled: false
  mvc:
    view:
      prefix: /WEB-INF/views/
      suffix: .jsp
    static-path-pattern: /resources/**
```

The `ldap` section is self-explanatory—we are setting up the embedded LDAP server with various parameters.

Change SpringSecurityConfig class for the LDAP configuration

```
package com.demo.springboot.config;

import java.util.Arrays;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authe
nticationManagerBuilder;
import
org.springframework.security.config.annotation.method.configuration.EnableGl
obalMethodSecurity;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSe
curity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurity
ConfigurerAdapter;
import org.springframework.security.ldap.DefaultSpringSecurityContextSource;

@EnableWebSecurity
@Configuration
@EnableGlobalMethodSecurity(securedEnabled = true)
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {
   private static final Logger LOG =
                    LoggerFactory.getLogger(SpringSecurityConfig.class);
   @Override
   protected void configure(HttpSecurity http) throws Exception {
       http.authorizeRequests()    .antMatchers("/admins").hasRole("ADMINS")
               .antMatchers("/users").hasRole("USERS")
               .anyRequest().fullyAuthenticated()
               .and()
               .httpBasic(); // Use Basic authentication
   }

   @Override
   public void configure(AuthenticationManagerBuilder auth) throws Exception
{
       auth
               .ldapAuthentication()
               .userDnPatterns("uid={0},ou=people")
               .userSearchBase("ou=people")
               .userSearchFilter("uid={0}")
               .groupSearchBase("ou=groups")
               .groupSearchFilter("uniqueMember={0}")
               .contextSource(contextSource())
               .passwordCompare()
               .passwordAttribute("userPassword");
   }
```

```
    @Bean
    public DefaultSpringSecurityContextSource contextSource() {
        LOG.info("Inside configuring embedded LDAP server");
        DefaultSpringSecurityContextSource contextSource = new
                DefaultSpringSecurityContextSource(
                Arrays.asList("ldap://localhost:8389/"),
"dc=packtpub,dc=com");
        contextSource.afterPropertiesSet();
        return contextSource;
    }
}
```

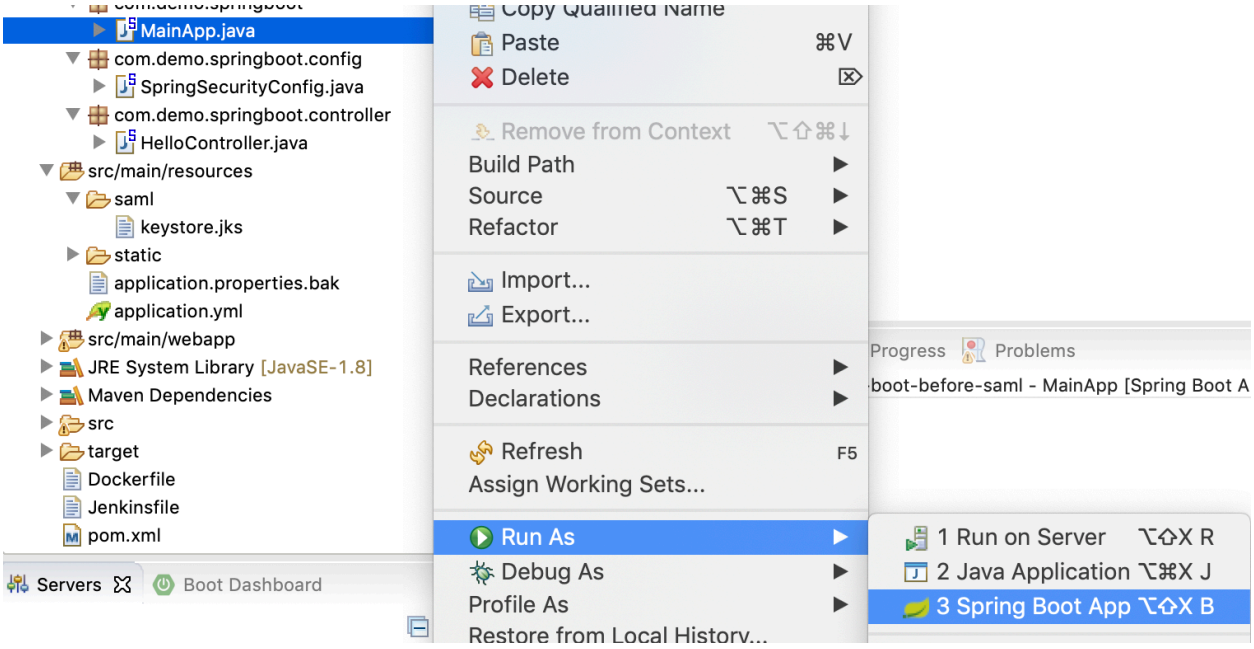
The first `configure` method is very similar to what we saw in the previous SAML example.

We have just added certain matches and separated the roles. With these changes, it will still perform basic authentication.

The second `configure` method is where we have set up authentication using the LDAP server.

The LDAP server stores user information in a directory-like format. This method details how to find the user by navigating through the directory structure.

33. RUN the APP



34. OUTPUT

Visit ***http://localhost:8080***

```
2019-10-29 18:07:39.667  INFO 22460 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2019-10-29 18:07:39.668  INFO 22460 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/8.5.29
2019-10-29 18:07:39.681  INFO 22460 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found
2019-10-29 18:07:39.803  WARN 22460 --- [ost-startStop-1] o.a.tomcat.util.scan.StandardJarScanner  : Failed to scan [file:/Users/nanda/.m2/repository/xalan/serializer/2.7.1/xml-apis.jar] from classloader hierarchy

java.io.FileNotFoundException: /Users/nanda/.m2/repository/xalan/serializer/2.7.1/xml-apis.jar (No such file or directory)
        at java.util.zip.ZipFile.open(Native Method) ~[na:1.8.0_201]
        at java.util.zip.ZipFile.<init>(ZipFile.java:225) ~[na:1.8.0_201]
        at java.util.zip.ZipFile.<init>(ZipFile.java:155) ~[na:1.8.0_201]
        at java.util.jar.JarFile.<init>(JarFile.java:166) ~[na:1.8.0_201]
        at java.util.jar.JarFile.<init>(JarFile.java:130) ~[na:1.8.0_201]
        at org.apache.tomcat.util.compat.JreCompat.jarFileNewInstance(JreCompat.java:188) ~[tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.JarFileUrlJar.<init>(JarFileUrlJar.java:65) ~[tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.JarFactory.newInstance(JarFactory.java:49) ~[tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.StandardJarScanner.process(StandardJarScanner.java:374) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.StandardJarScanner.processURLs(StandardJarScanner.java:309) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.StandardJarScanner.doScanClassPath(StandardJarScanner.java:266) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.StandardJarScanner.scan(StandardJarScanner.java:229) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.jasper.servlet.TldScanner.scanJars(TldScanner.java:262) [tomcat-embed-jasper-8.5.29.jar:8.5.29]
        at org.apache.jasper.servlet.TldScanner.scan(TldScanner.java:104) [tomcat-embed-jasper-8.5.29.jar:8.5.29]
        at org.apache.jasper.servlet.JasperInitializer.onStartup(JasperInitializer.java:101) [tomcat-embed-jasper-8.5.29.jar:8.5.29]
        at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5204) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1421) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1411) [tomcat-embed-core-8.5.29.jar:8.5.29]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) [na:1.8.0_201]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_201]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_201]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_201]

2019-10-29 18:07:39.803  WARN 22460 --- [ost-startStop-1] o.a.tomcat.util.scan.StandardJarScanner  : Failed to scan [file:/Users/nanda/.m2/repository/xalan/xalan/2.7.1/xercesImpl.jar] from classloader hierarchy

java.io.FileNotFoundException: /Users/nanda/.m2/repository/xalan/xalan/2.7.1/xercesImpl.jar (No such file or directory)
        at java.util.zip.ZipFile.open(Native Method) ~[na:1.8.0_201]
        at java.util.zip.ZipFile.<init>(ZipFile.java:225) ~[na:1.8.0_201]
        at java.util.zip.ZipFile.<init>(ZipFile.java:155) ~[na:1.8.0_201]
        at java.util.jar.JarFile.<init>(JarFile.java:166) ~[na:1.8.0_201]
        at java.util.jar.JarFile.<init>(JarFile.java:130) ~[na:1.8.0_201]
        at org.apache.tomcat.util.compat.JreCompat.jarFileNewInstance(JreCompat.java:188) ~[tomcat-embed-core-8.5.29.jar:8.5.29]
        at org.apache.tomcat.util.scan.JarFileUrlJar.<init>(JarFileUrlJar.java:65) ~[tomcat-embed-core-8.5.29.jar:8.5.29]
```

35. Add the following entry in application.yml

```yaml
server:
  tomcat:
    additional-tld-skip-patterns: '*.jar'
```

```
RUN the APP

Visit http://localhost:8080
```

Open a browser and enter `http://localhost:8080`. Enter the username/password as `tjohn/tjohn@password` (look for user setup in the LDIF file).

You will be taken to `home.jsp`, where you will see a friendly welcome message, as shown in the following screenshot: