

LDAP Authentication

Tip submitted by @mleneveut (<https://github.com/mleneveut>) updated by @iliasnaamane (<https://github.com/iliasnaamane>)__

To add an LDAP authentication to your JHipster application, follow these steps :

- Add the dependencies spring-ldap-core and spring-security-ldap. Example for gradle in build.gradle :

```
compile group: 'org.springframework.security', name: 'spring-security-ldap', version:
spring_security_version
```

- Modify the SecurityConfiguration.java, add method configureGlobal(AuthenticationManagerBuilder auth) and getContextSource()

```
@Inject
public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
    auth.ldapAuthentication()
        .userSearchBase("o=myO,ou=myOu") //don't add the base
        .userSearchFilter("(uid={0})")
        .groupSearchBase("ou=Groups") //don't add the base
        .groupSearchFilter("member={0}")
        .contextSource(getContextSource());
}

@Bean
public LdapContextSource getContextSource() {
    LdapContextSource contextSource = new LdapContextSource();
    contextSource.setUrl("ldap://[IP goes here]:[port goes here]");
    contextSource.setBase("dc=mycompany,dc=com");
    contextSource.setUserDn("cn=aUserId,dc=mycompany,dc=com");
    contextSource.setPassword("hisPassword");
    contextSource.afterPropertiesSet(); //needed otherwise you will have a NullPointerException in spring

    return
}
```

- Modify the SecurityUtils.java, method getCurrentUserLogin()

```
} else if (authentication.getPrincipal() instanceof LdapUserDetails) {
    LdapUserDetails ldapUser = (LdapUserDetails) authentication.getPrincipal();
    return ldapUser.getUsername();
}
```

- Add a new CustomAuthenticationManager class which implements the AuthenticationManager interface and override the authentication method in order to force the authentication process to authenticate the user through LDAP.



```

@Component
public class CustomAuthenticationManager implements AuthenticationManager {

    LdapAuthenticationProvider provider = null;

    private static final Logger log = LoggerFactory.getLogger(CustomAuthenticationManager.class);

    private final UserRepository userRepository;

    @Autowired
    private final LdapContextSource ldapContextSource;

    public CustomAuthenticationManager(UserRepository userRepository, LdapContextSource ldapContextSource) {
        this.userRepository = userRepository;
        this.ldapContextSource = ldapContextSource;
    }

    @Override
    public Authentication authenticate(Authentication authentication) {
        log.debug("AUTHENTICATION Login" + authentication.getName());
        log.debug("AUTHENTICATION Password" + authentication.getCredentials().toString());

        BindAuthenticator bindAuth = new BindAuthenticator(ldapContextSource);
        FilterBasedLdapUserSearch userSearch = new FilterBasedLdapUserSearch(
            "", "(uid={})",
            ldapContextSource);
        try{
            bindAuth.setUserSearch(userSearch);
            bindAuth.afterPropertiesSet();
        } catch (Exception ex) {
            java.util.logging.Logger.getLogger(CustomAuthenticationManager.class.getName()).log(Level.SEVERE, null, ex);
        }
        provider = new LdapAuthenticationProvider(bindAuth);
        provider.setUserDetailsContextMapper(new UserDetailsContextMapper() {
            @Override
            public UserDetails mapUserFromContext(DirContextOperations ctx, String username, Collection<? extends GrantedAuthority> clctn) {
                Optional<User> isUser = userRepository.findOneWithAuthoritiesByLogin(username);

                final User user = isUser.get();
                Set<Authority> userAuthorities = user.getAuthorities();
                Collection<GrantedAuthority> grantedAuthorities = new ArrayList<>();
                for(Authority a: userAuthorities){
                    GrantedAuthority grantedAuthority = new SimpleGrantedAuthority(a.getName());
                    grantedAuthorities.add(grantedAuthority);
                }
                return new org.springframework.security.core.userdetails.User(
                    username, "1" , grantedAuthorities);
            }
        });

        @Override

```



```
public void mapUserToContext(UserDetails ud, DirContextAdapter dca) {  
  
    }  
});  
return provider.authenticate(authentication);  
}
```

JHipster is sponsored by:



([https://developer.okta.com/signup?
utm_source=JHipster&utm_medium=logo&utm_campaign=Gold-
Sponsor](https://developer.okta.com/signup?utm_source=JHipster&utm_medium=logo&utm_campaign=Gold-Sponsor))

-  Team (<https://www.jhipster.tech/team/>)
-  Artwork (<https://www.jhipster.tech/artwork/>)
-  Policies (<https://www.jhipster.tech/policies/>)
-  Meetups (<https://www.jhipster.tech/meetups/>)
-  Archives (<https://www.jhipster.tech/documentation-archive/>)



(<https://github.com/jhipster/jhipster-generator>)
(<https://twitter.com/jhipster>)

Copyright © JHipster 2013-2019 | Theme based on Freelancer
(<https://github.com/IronSummitMedia/startbootstrap-freelancer>) and Flat admin
(<https://github.com/tui2tone/flat-admin-bootstrap-templates>)

