LDAP ile Kimliklendirme

LDAP ve Kimliklendirme

- LDAP, pek çok kurumda **merkezi bir kullanıcı deposu** (user realm) ve **kimliklendirme servisi** olarak kullanılır
- Kullanıcılara ait **yetkilerde saklanabilir**
- LDAP ile kimliklendirmenin temel adımları
  - Kullanıcı adı ile **DN** (distinguished name) elde edilmesi
  - Kullanıcının kimliklendirilmesi
  - Kullanıcıya ait yetkilerin tespit edilmesi

LDAP ile Kimliklendirme Yöntemleri

- LDAP ile kimliklendirme iki şekilde gerçekleştirilebilir
  - **LdapAuthenticationProvider** ile
    - Kullanıcının girdiği username ve password'ü kullanılarak ldap server'a bind edilmeye çalışılır
    - Bind authentication adı verilir
  - **LdapUserDetailsService** ile
    - Ldap server'da bir ldap sorgusu çalıştırılır
    - Bu sorgu ile kullanıcının LDAP'taki şifresi elde edilir ve kullanıcının girdiği şifre ile karşılaştırılır

LdapAuthenticationProvider ile Kimliklendirme

- Bind authentication yöntemi en yaygın LDAP ile kimliklendirme senaryosudur
- Kullanıcının girdiği **username** ve **password** ile LDAP dizinine bind edilmeye çalışılır

LdapAuthenticationProvider ile Kimliklendirme

```
<security:ldap-authentication-provider
user-dn-pattern="uid={0},ou=people"
group-search-base="ou=groups"
group-search-filter="member={0}">
</security:ldap-authentication-provider>
```

LDAP grup tanımları ile kullanıcıya ait rollerin de LDAP'dan yüklenmesi mümkündür.

Default olarak LDAP grup isimlerine **ROLE_** ön eki eklenir. İstenirse bu davranış değiştirilebilir

LdapUserDetailsService ile Kimliklendirme

- Bu senaryoda ise **DaoAuthenticationProvider** UserDetails nesnesine erişimek için LdapUserDetailsService'i kullanır
- **LdapUserDetailsService**'e username parametresini vererek LDAP'dan **UserDetails** nesnesine erişilir

LdapUserDetailsService ile Kimliklendirme

- LDAP'a erişebilmek için **ldap-server**'ın **manager-dn** ve **manager-password** attribute'larını set etmek gerekebilir
- Eğer set edilmezlerse LDAP sunucuya default olarak **anonim** yetkilerle erişilmeye çalışılır
- Bazı kurumlarda **manager-dn** ve **manager-password** uygulama içerisinde kullanılmak istenmeyebileceği için bu yöntem buralarda sorun yaratabilir

LdapUserDetailsService ile Kimliklendirme

```
<security:ldap-user-service
user-search-base="ou=people"
user-search-filter="uid={0}"
group-search-base="ou=groups"
group-search-filter="member={0}"
group-role-attribute="cn" id="ldapUserService"/>
<security:authentication-manager>
     <security:authentication-provider
               user-service-ref="ldapUserService" />
</security:authentication-manager>
```

LDAP Server'a Erişim

```
<security:ldap-server ldif="classpath:users.ldif"
root="dc=javaegitimleri,dc=com"/>
```

- **ldif** attribute tanımlanırsa test amaçlı **gömülü bir LDAP sunucu** çalıştırılır

```
<security:ldap-server
url="ldap://localhost:389/dc=javaegitimleri,dc=com" />
```

- **url** attribute ile uzaktaki **ldap sunucu**'ya erişim sağlanır

Active Directory ile Kimliklendirme

- MS Active Directory'nin kendine özgü **standart olmayan** bir süreci vardır
- Dolayısı ile **LdapAuthenticationProvider** kullanılarak kimliklendirme sorun yaratabilmektedir
- Spring Security bunun için **ActiveDirectoryLdapAuthenticationProvider** sınıfı sunmaktadır

Active Directory ile Kimliklendirme

```
<bean id="adAuthenticationProvider"

class="org.springframework.security.ldap.authentication.ad.ActiveDirectoryLdapAuthenticationProvider">
    <constructor-arg value="mydomain.com" />
    <constructor-arg value="ldap://adserver.mydomain.com/" />
</bean>
```

Active Directory ile Kimliklendirme

- Kullanıcılar username veya username@domain şeklinde giriş yaptıklarında kimliklendirmeye tabi tutulurlar
- **Manager user** bilgilerine **gerek yok**tur
- AD'den dönen bilgilerle **UserDetails** populate edilir
- İstenirse bu işlem **UserDetailsContextMapper** ile özelleştirilebilir

Active Directory ile Kimliklendirme

- Kullanıcı rol bilgileri ise default olarak **memberOf** attribute değerlerinden elde edilmektedir
- İstenirse rol bilgileri de **GrantedAuthoritiesMapper** ile özelleştirilebilir

Active Directory ile Kimliklendirme

- AD hataları **BadCredentialsException** ile sonuçlanacaktır
- İstenirse **convertSubErrorCodesToExceptions** **true** yapılarak AD'nin hata kodlarına bakılarak exception mesajı da detaylandırılabilir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com