



<http://www.lycato.com>



Spring Boot

# Spring Security DB

*Nguyễn Nghiệm*





## ☐ AUTHENTICATIONMANAGERBUILDER

- ❖ ACCOUNT

- ❖ ROLE

- ❖ AUTHORITY

## ☐ USERDETAILS

## ☐ USERDETAILSSERVICE





```
protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
    auth.inMemoryAuthentication()  
        .withUser("user1").password(pe.encode("*")).roles("ADMIN", "SUPER")  
        .and().withUser("user2").password(pe.encode("*")).roles("SUPER", "STAFF")  
        .and().withUser("user3").password(pe.encode("*")).roles("STAFF")  
        .and().withUser("user4").password(pe.encode("*")).roles();  
}
```

## Accounts

Username	...
user1	
user2	
user3	
user4	

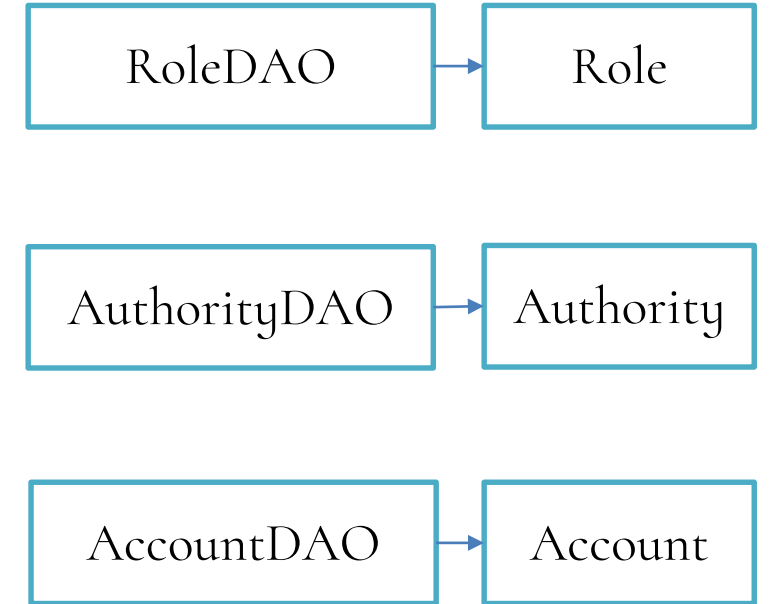
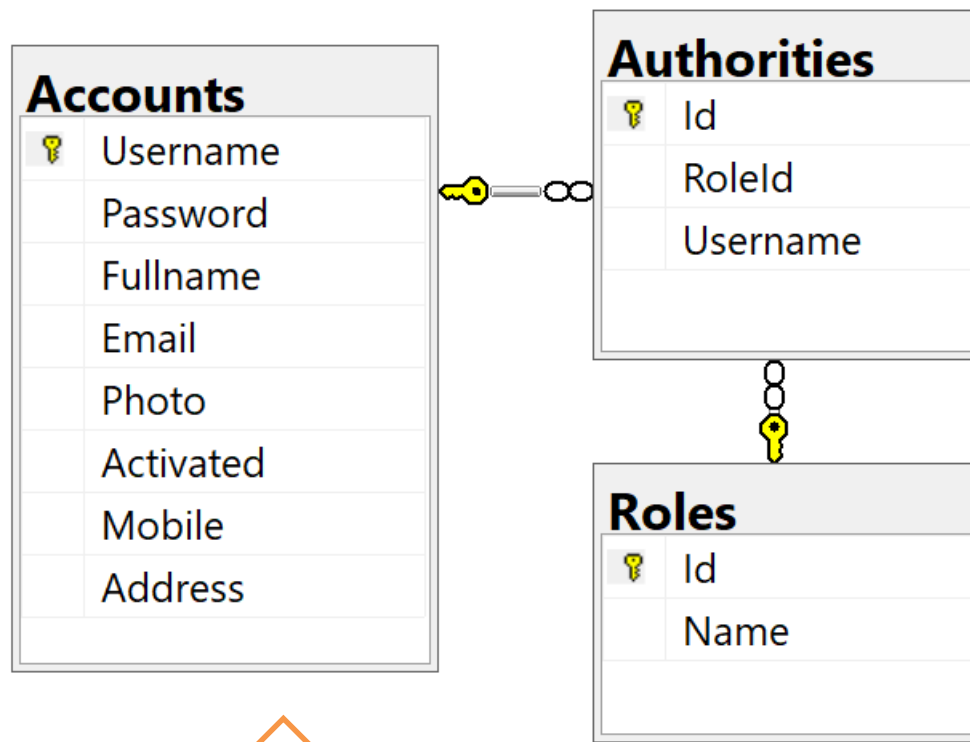
## Authorities

Id	Username	RoleId
1	user1	ADMIN
2	user1	SUPER
3	user2	SUPER
4	user2	STAFF
5	user3	STAFF

## Roles

Id	Name
ADMIN	Quản trị
SUPER	Giám sát
STAFF	Nhân viên

User Source Database



Accounts

Username	...
<i>user1</i>	
<i>user2</i>	
<i>user3</i>	
<i>user4</i>	

Authorities

Id	Username	RoleId
1	<i>user1</i>	<i>ADMIN</i>
2	<i>user1</i>	<i>SUPER</i>
3	<i>user2</i>	<i>SUPER</i>
4	<i>user2</i>	<i>STAFF</i>
5	<i>user3</i>	<i>STAFF</i>

Roles

Id	Name
<i>ADMIN</i>	Quản trị
<i>SUPER</i>	Giám sát
<i>STAFF</i>	Nhân viên



## USER SOURCE

```
protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
    auth.inMemoryAuthentication() // quản lý nguồn users  
        .withUser("user1").password(pe.encode("123")).roles("SUPER").and()  
        ...  
}
```

@Configuration



```
@Autowired  
UserDetailsService userDetailsService; // => đối tượng xác thực user
```

```
@Override  
protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
    auth.userDetailsService(userDetailsService);  
}
```

@Configuration



## USERDETAILSSERVICE & USERDETAILS

```
@Service
public class UserDetailsServiceImpl implements UserDetailsService{
    @Override
    public UserDetails loadUserByUsername(String username)
        throws UsernameNotFoundException { /*Mã xác thực*/ }
}
```

```
public class UserDetailsImpl implements UserDetails {
    @Override String getUsername() {...} // tên đăng nhập
    @Override Collection<? extends GrantedAuthority> getAuthorities() {...} // vai trò
    @Override String getPassword() {...} // lấy mật khẩu
    @Override boolean isAccountNonExpired() {...} // hết hạn ?
    @Override boolean isAccountNonLocked() {...} // bị khóa ?
    @Override boolean isCredentialsNonExpired() {...} // hết hạn mật khẩu ?
    @Override boolean isEnabled() {...} // còn hiệu lực ?
}
```

User Data Model



## VIẾT MÃ XÁC THỰC USER

```
@Service
public class UserDetailsServiceImpl implements UserDetailsService {
    @Autowired
    AccountDAO dao;
    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        try {
            Account account = dao.getById(username);
            return new UserDetailsImpl(account); // Đối tượng UserDetails
        } catch (Exception e) {
            throw new UsernameNotFoundException(username + " not found!");
        }
    }
}
```



## CÀI ĐẶT MÃ CHO USERDETAILS

```
@Data @NoArgsConstructor @AllArgsConstructor
public class UserDetailsImpl implements UserDetails{
    Account account;
    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return account.getAuthorities().stream()
            .map(a -> new SimpleGrantedAuthority("ROLE_" + a.getRole().getId())).toList();
    }
    @Override public String getPassword() {return account.getPassword();}
    @Override public String getUsername() {return account.getUsername();}
    @Override public boolean isAccountNonExpired() {return true;}
    @Override public boolean isAccountNonLocked() {return true;}
    @Override public boolean isCredentialsNonExpired() {return true;}
    @Override public boolean isEnabled() {return account.isActivated();}
}
```





## ✓ AuthenticationManagerBuilder

- ✓ Accounts
- ✓ Roles
- ✓ Authorities

## ✓ UserDetails

## ✓ UserDetailsService





Thank  
you