MỤC TIÊU

- Phân quyền truy cập với vai trò
- Sử dụng người dữ liệu người dùng từ CSDL
- Sử dụng nguồn dữ liệu tùy biến

BÀI 1: AUTHORIZATION (2 ĐIỂM)

Sao chép lại dự án của Lab1.2 và bổ sung, hiệu chỉnh để hoàn thành bài tập này theo yêu cầu sau:

- UserDetailsService: Sử dụng InMemoryUserDetailsManager với các User được phân vai trò như sau:

USERNAME	PASSWORD	ROLES
user@gmail.com	123	USER
admin@gmail.com	123	ADMIN
both@gmail.com	123	USER, ADMIN

- SecurityFilterChain: Phân quyền truy xuất cho các vai trò theo yêu cầu su:

PRIVATE URL	PERMISSION
/poly/url1	Authenticated
/poly/url2	USER
/poly/url3	ADMIN
/poly/url4	USER, ADMIN
Các URL còn lại	PermitAll

- Cấu hình xử lý từ chối truy xuất: truy xuất không đúng vai trò được chuyển về access-denied.html với nội dung trang tùy ý bạn.

BÀI 2: SỬ DỤNG JDBCUSERDETAILSMANAGER (3 ĐIỂM)

Sao chép lại dự án của Lab2.1 và bổ sung, hiệu chỉnh để hoàn thành bài tập này theo yêu cầu sau:

- Tạo CSDL J6Security
  - Tạo CSDL với lược đồ (schema) mặc định theo quy định của Spring Boot Security như sau:

```
CREATE TABLE Users
(
   Username VARCHAR(50) NOT NULL,
   Password VARCHAR(500) NOT NULL,
   Enabled BIT NOT NULL,
   PRIMARY KEY(Username)
)

CREATE TABLE Authorities
(
   Id BIGINT NOT NULL identity(1, 1),
   Username VARCHAR(50) NOT NULL,
   Authority VARCHAR(50) NOT NULL,
   PRIMARY KEY(Id),
   UNIQUE(Username, Authority),
   FOREIGN KEY(Username) REFERENCES Users(Username)
      ON DELETE CASCADE ON UPDATE CASCADE
)
```

- Nhập dữ liệu mẫu cho CSDL J6Security như sau:

```
INSERT INTO Users(Username, Password, Enabled) VALUES('user@gmail.com', '{noop}123', 1)
INSERT INTO Users(Username, Password, Enabled) VALUES('admin@gmail.com', '{noop}123', 1)
INSERT INTO Users(Username, Password, Enabled) VALUES('both@gmail.com', '{noop}123', 1)

INSERT INTO Authorities(Username, Authority) VALUES('user@gmail.com', 'ROLE_USER')
INSERT INTO Authorities(Username, Authority) VALUES('admin@gmail.com', 'ROLE_ADMIN')
INSERT INTO Authorities(Username, Authority) VALUES('both@gmail.com', 'ROLE_USER')
INSERT INTO Authorities(Username, Authority) VALUES('both@gmail.com', 'ROLE_ADMIN')
```

- Khai báo Dependency (pom.xml)

```
<dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
      <groupId>com.microsoft.sqlserver</groupId>
      <artifactId>mssql-jdbc</artifactId>
      <scope>runtime</scope>
</dependency>
```

- Khai báo cấu hình DataSource (application.properties) để kết nối đến CSDL J6Security

```
spring.datasource.url=jdbc:sqlserver://localhost;database=<db>;encrypt=true;trustServerCertificate=true;
spring.datasource.username=sa
spring.datasource.password=123
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
spring.jpa.hibernate.dialect=org.hibernate.dialect.SQLServer2012Dialect
spring.jpa.show-sql=false
spring.jpa.hibernate.ddl-auto = none
```

- UserDetailsService: Sử dụng JdbcUserDetailsManager thay vì InMemoryUserDetailsManager

```
@Bean // CSDL đúng lược đồ mặc định
public UserDetailsService userDetailsService(DataSource dataSource) {
    return new JdbcUserDetailsManager(dataSource);
}
```

BÀI 3: XÂY DỰNG DAOUSERDETAILSMANAGER (3 ĐIỂM)

Sao chép lại dự án của Lab2.1 và bổ sung, hiệu chỉnh để hoàn thành bài tập này theo yêu cầu sau:

- Tạo CSDL J6Security2 gồm 3 bảng Users, Roles và UserRoles và nhập dữ liệu mẫu như lược đồ sau

Users

Username	Password	Enabled
user@gmail.com	{noop}123	1
admin@gmail.com	{noop}123	1
both@gmail.com	{noop}123	1

Roles

Id	Name
ROLE_USER	Nhân viên
ROLE_ADMIN	Quản lý

UserRoles

Id	Username	RoleId
1	user@gmail.com	ROLE_USER
2	admin@gmail.com	ROLE_ADMIN
3	both@gmail.com	ROLE_USER
4	both@gmail.com	ROLE_ADMIN

- Xây dựng các lớp thực thể User, Role và UserRole

```java
@Data
@Entity
@Table(name = "J6users")
public class User {
    @Id
    String username;
    String password;
    boolean enabled;
    @OneToMany(mappedBy = "user", fetch = FetchType.EAGER)
    List<UserRole> userRoles;
}
```

```java
@Data
@Entity
@Table(name = "J6roles")
public class Role {
    @Id
    String id;
    String name;
    @OneToMany(mappedBy = "role")
    List<UserRole> userRoles;
}
```

```java
@Data
@Entity
@Table(name = "J6userroles")
public class UserRole {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id;
    @ManyToOne
    @JoinColumn(name = "username")
    User user;
    @ManyToOne
    @JoinColumn(name = "roleid")
    Role role;
}
```

- Xây dựng các lớp truy xuất dữ liệu UserDAO, RoleDAO và UserRoleDAO

```java
public interface UserDAO extends JpaRepository<User, String> {}
public interface RoleDAO extends JpaRepository<Role, String> {}
```

```
public interface UserRoleDAO extends JpaRepository<UserRole, Long> {}
```

- Xây dựng DaoUserDetailsManager

```
public class DaoUserDetailsManager implements UserDetailsService {
    @Autowired
    UserDAO dao;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        poly.demo.entity.User user = dao.findById(username).get();
        String password = user.getPassword();
        String[] roles = user.getUserRoles().stream()
                .map(ur -> ur.getRole().getId().substring(5))
                .toList().toArray(String[]::new);
        return User.withUsername(username).password(password).roles(roles).build();
    }
}
```

- UserDetailsService: Cấu hình thay thế InMemoryUserDetailsManager bằng DaoUserDetailsManager để quản lý nguồn dữ liệu người sử dụng.

```
@Bean
public UserDetailsService userDetailsService() {
    return new DaoUserDetailsManager();
}
```

- Chạy và kiểm thử các liên kết

BÀI 4: GIẢNG VIÊN CHO THÊM (2 ĐIỂM)