

User Storage

Bryan Hansen
twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



pluralsight hardcore dev and IT training

Storage



in-memory

```
<authentication-manager>
    <authentication-provider>
        <user-service>
            <user name="bryan" password="secret" authorities="ROLE_USER"/>
        </user-service>
    </authentication-provider>
</authentication-manager>

<authentication-manager>
    <authentication-provider>
        <user-service>
            <user name="bryan" password="secret" authorities="ROLE_USER"/>
            <user name="chris" password="secrettoo" authorities="ROLE_USER"/>
        </user-service>
    </authentication-provider>
</authentication-manager>
```

Database

- You can use any database
 - MySQL 5.X
 - <http://dev.mysql.com/downloads/installer/5.6.html>
- Spring Security contains a JdbcDaoImpl that can be configured to any database
- Certain features are only available once connected to a database
- For Development:
 - Username = root
 - Password = password

User Tables

- We need to create two tables
 - Users Table
 - Authorities Table

```
create table users(
    username varchar(50) not null primary key,
    password varchar(50) not null,
    enabled boolean not null);
```

```
create table authorities (
    username varchar(50) not null,
    authority varchar(50) not null,
    constraint fkAuthorities_users
    foreign key(username) references users(username));
    create unique index ix_auth_username on authorities (username,authority);
```

Create User/Authority

- User SQL:

```
insert into users (username, password, enabled)
values ("bryan", "secret", true);
```

- Authorities SQL:

```
insert into authorities (username, authority)
value ("bryan", "ROLE_USER");
```

Maven

- There is a Maven dependency for the MySQL JDBC Driver:
 - groupId = mysql
 - artifactId = mysql-connector-java
 - version = 5.1.6
- Also need Spring JDBC added:
 - groupId = org.springframework
 - artifactId = spring-jdbc
 - version = 3.2.6.RELEASE

security-config.xml

- Create a datasource for the database connection
 - driverClassName = com.mysql.jdbc.Driver
 - url = jdbc:mysql://localhost:3306/fitnessTracker
 - username = root
 - password = password
- Create instance of JdbcDaoImpl
 - class = org.springframework.security.core.userdetails.jdbc.JdbcDaoImpl

Summary

- Covered in-memory configuration
- Installed MySql
 - Created a database schema
 - Created tables
 - Create a user and it authorities
- Added Jars using Maven
- Configured the app to use the database

