Step-by-step guide to Access Control

- Install JDBC driver for Tomcat application server
- Configure JDBC driver + DataSourceRealm
- Modify database schema + data
- Implement authentication (login)
- Implement authorization (protected pages)
Install JDBC driver for Tomcat Application Server

- Configure Tomcat to access Postgresql
  - Place postgresql-8.4-701.jdbc3.jar in $TOMCAT_HOME/lib

Step-by-step Guide to Access Control

- Install JDBC driver for Tomcat application server
- **Configure JDBC driver + DataSourceRealm**
- Modify database schema + data
- Implement authentication (login)
- Implement authorization (protected pages)
Configure JDBC driver

META-INF/context.xml

```xml
<Context path="" debug="5" override="true" reloadable="true">
   <Resource name="jdbc/ClassesDbPool" description="Classes DB Pool"
      driverClassName="org.postgresql.Driver"
      type="javax.sql.DataSource"
      auth="Container"
      url="jdbc:postgresql://localhost/access_control"
      username="postgres"
      password="postgres"
      defaultAutoCommit="false"
      maxActive="10"
      minIdle="0"
      maxIdle="5"
      maxWait="3000"
      removeAbandoned="true"
      removeAbandonedTimeout="60"
      logAbandoned="true"
      validationQuery="SELECT 1" />
   ...
</Context>
```

Configure DataSourceRealm

META-INF/context.xml

```xml
<Context path="" debug="5" override="true" reloadable="true">
   ...
   <Realm className="org.apache.catalina.realm.DataSourceRealm" debug="99"
      dataSourceName="jdbc/ClassesDbPool"
      localDataSource="true"
      users="users"
      userTable="users"
      userNameCol="user_name"
      userCredCol="password"
      roleNameCol="role"
      digest="MD5"/>
</Context>
```
Step-by-step Guide to Access Control

- Install JDBC driver for Tomcat application server
- Configure JDBC driver + DataSourceRealm
- Modify database schema + data
- Implement authentication (login)
- Implement authorization (protected pages)

Modify database schema

- Database schema may be different from what is required by DataSourceRealm
  - Table names and column names can be configured
  - Additional columns are harmless

<table>
<thead>
<tr>
<th>users</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
</tr>
<tr>
<td>john</td>
</tr>
<tr>
<td>ted</td>
</tr>
<tr>
<td>mary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>user_roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
</tr>
<tr>
<td>john</td>
</tr>
<tr>
<td>ted</td>
</tr>
<tr>
<td>mary</td>
</tr>
</tbody>
</table>
Modify database schema

- Database schema may be different from what is required by DataSourceRealm
  - Create additional tables that store data redundantly

```
<table>
<thead>
<tr>
<th>user_name</th>
<th>password</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>john</td>
<td>john</td>
<td>student</td>
</tr>
<tr>
<td>ted</td>
<td>ted</td>
<td>student</td>
</tr>
<tr>
<td>mary</td>
<td>mary</td>
<td>admin</td>
</tr>
</tbody>
</table>
```

CREATE VIEW users AS
SELECT user_name, password
FROM registered_users

CREATE VIEW roles AS
SELECT user_name, role
FROM registered_users

Modify database schema

- Database schema may be different from what is required by DataSourceRealm
  - Create a SQL view that converts the data into the appropriate format (recommended)

```
<table>
<thead>
<tr>
<th>user_name</th>
<th>password</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>john</td>
<td>john</td>
<td>student</td>
</tr>
<tr>
<td>ted</td>
<td>ted</td>
<td>student</td>
</tr>
<tr>
<td>mary</td>
<td>mary</td>
<td>admin</td>
</tr>
</tbody>
</table>
```

CREATE VIEW users AS
SELECT user_name, password
FROM registered_users

CREATE VIEW roles AS
SELECT user_name, role
FROM registered_users
Modify database schema

- Database schema may be different from what is required by DataSourceRealm
  - Create a SQL view that converts the data into the appropriate format (recommended)

```sql
CREATE VIEW users AS
SELECT user_name, password
FROM students
UNION
SELECT user_name, password
FROM admin

CREATE VIEW roles AS
SELECT user_name, 'students'
FROM students
UNION
SELECT user_name, 'admin'
FROM admin
```

<table>
<thead>
<tr>
<th>students</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>john</td>
<td>john</td>
<td></td>
</tr>
<tr>
<td>ted</td>
<td>ted</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>admin</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mary</td>
<td>mary</td>
<td></td>
</tr>
</tbody>
</table>

Insert sample data

- Insert sample users and roles
  - Remember to use MD5 function to hash passwords

```sql
INSERT INTO users (user_name, password)
VALUES ('mary', MD5('mary'));

INSERT INTO user_roles (user_name, role)
VALUES ('mary', 'admin');
```
Step-by-step Guide to Access Control

- Install JDBC driver for Tomcat application server
- Configure JDBC driver + DataSourceRealm
- Modify database schema + data
- Implement authentication (login)
- Implement authorization (protected pages)

Configure login page

WEB-INF/web.xml

<!-- Login configuration uses form-based authentication -->
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>
    Form-Based Authentication Area
  </realm-name>
  <form-login-config>
    <form-login-page>/login.jsp</form-login-page>
    <form-error-page>/login-error.jsp</form-error-page>
  </form-login-config>
</login-config>
Implement login page

**login.jsp**

```html
<html>...
<form method="POST" action="j_security_check">
  Username:
  <input size="12" name="j_username" type="text"/><br/>
  Password:
  <input size="12" name="j_password" type="password"/><br/>
  <input type="submit" value="Login"/>
</form>
...</html>
```

Step-by-step Guide to Access Control

- Install JDBC driver for Tomcat application server
- Configure JDBC driver + DataSourceRealm
- Modify database schema + data
- Implement authentication (login)
- **Implement authorization (protected pages)**
Configure roles

WEB-INF/web.xml

```xml
<web-app>
    ...
    <security-role>
        <role-name>student</role-name>
    </security-role>
    <security-role>
        <role-name>admin</role-name>
    </security-role>
    ...
</web-app>
```

Configure security constraints

WEB-INF/web.xml

```xml
<web-app>
    ...
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>Admin Area</web-resource-name>
            <!-- Define the context-relative URL(s) to protect -->
            <url-pattern>/admin/**</url-pattern>
        </web-resource-collection>
        <auth-constraint>
            <role-name>admin</role-name>
        </auth-constraint>
    </security-constraint>
    ...
</web-app>
```
Test accessing currently logged in user

page.jsp

Username: <%= request.getRemoteUser() %>