« Part 1 – Creating Entity Classes With POJO + Annotation
Part 3 – Basic Implementation of CRUD »

Part 2 – Securing Web Application GlassFish V3 – JAAS (Authentication and Authorization) 24

21 Sep 2012 | Java EE

In this article, I will demonstrate how to connect our web application to our database and set up GlassFish JDBC realm authentication.

If you don't want to use the provided source code for this tutorial you need to create a Maven Web Application. In NetBeans simply go to File -> New Project -> Maven -> Web Application and the rest will be explained in the tutorial.

To be able to secure access to our web application we need to provide
JAAS Context: Identifier for the login module when configuring our web application
JNDI: JDBC resource that we will create for our database
User Table name that contains the list of authorized users
User Name Column
Password Column
Group Table name that contains the list of role groups
Group Table **User Name** Column
Group Name Column

Before we start configuring, we will create our tables first or just let them automatically generated by using eclipselink.ddl-generation. Don't forget to create your database schema, in this project I called my database SimpleCRUD_db.

There will be 4 tables/entities:
Address
User
User_Roles
Role

We will create three roles (you can have as many as you can) User Roles:
- Administrators
- Managers
- Users

EER Model is looking like this at the moment:

Blogroll

- Arquillian testing tool for Java EE
- PrimeFaces JSF Component Suite

Recent Posts

- Glassfish4 and JSF 2.2 ViewScope support (JSF 2.2, CDI, JPA 2 (ORM) / EclipseLink, JAAS , MySQL, example CRUD web application)
- My Book Review – Instant PrimeFaces Starter- Packt Publishing
- Reviewing Instant PrimeFaces Starter- Packt Publishing
- My Book Review – PrimeFaces Cookbook by Oleg Varaksin and Mert Çalışkan
- Reviewing PrimeFaces Cookbook – Packt Publishing – Oleg Varaksin, Mert Çalışkan

Recent Comments

- simtay on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation
- simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA
- Mohamed on Part 4 – Lazy Loading and Pagination with JSF 2.1 / Primefaces 3.4.2 DataTable and Form validation
- dukgo on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA
- simtay on Simple CRUD Web Application with JSF 2.1, PrimeFaces 3.5, Maven and JPA

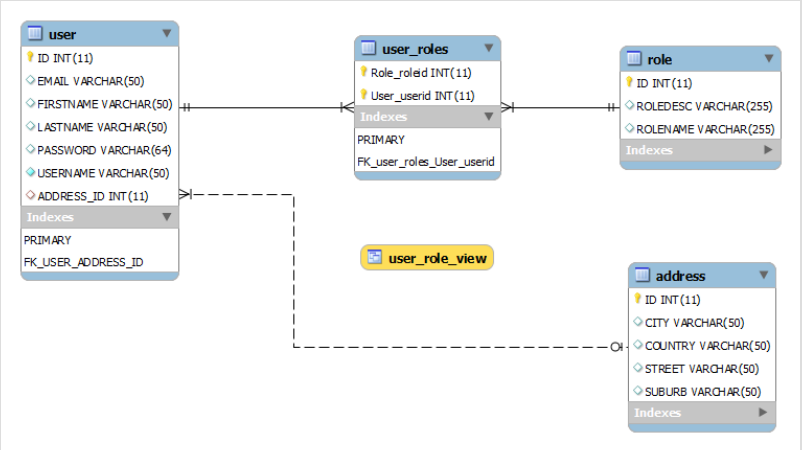
Archives

- November 2013
- August 2013
- April 2013
- March 2013
- January 2013
- September 2012

Categories

- Java EE

Meta



In your persistence.xml file if Table Generation Strategy is set to Generate than you don't have create the tables manually by hand. If you set it to none then you will have to use this SQL script (you can just copy and execute it in command line [I prefer MySQL workbench]) to create your tables :

```
CREATE TABLE `address` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `CITY` varchar(50) DEFAULT NULL,
 `COUNTRY` varchar(50) DEFAULT NULL,
 `STREET` varchar(50) DEFAULT NULL,
 `SUBURB` varchar(50) DEFAULT NULL,
 PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$
delimiter $$

CREATE TABLE `role` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `ROLEDESC` varchar(255) DEFAULT NULL,
 `ROLENAME` varchar(255) DEFAULT NULL,
 PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$
delimiter $$

CREATE TABLE `user` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `EMAIL` varchar(50) DEFAULT NULL,
 `FIRSTNAME` varchar(50) DEFAULT NULL,
 `LASTNAME` varchar(50) DEFAULT NULL,
 `PASSWORD` varchar(64) DEFAULT NULL,
 `USERNAME` varchar(50) NOT NULL,
 `ADDRESS_ID` int(11) DEFAULT NULL,
 PRIMARY KEY (`ID`),
 KEY `FK_USER_ADDRESS_ID` (`ADDRESS_ID`),
 CONSTRAINT `FK_USER_ADDRESS_ID` FOREIGN KEY (`ADDRESS_ID`) REFERENC
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$
delimiter $$

CREATE TABLE `user_roles` (
 `Role_roleid` int(11) NOT NULL,
 `User_userid` int(11) NOT NULL,
 PRIMARY KEY (`Role_roleid`,`User_userid`),
 KEY `FK_user_roles_User_userid` (`User_userid`),
 CONSTRAINT `FK_user_roles_User_userid` FOREIGN KEY (`User_userid`)
 CONSTRAINT `FK_user_roles_Role_roleid` FOREIGN KEY (`Role_roleid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$
```

Also I need to mention that I created a view for JAAS authentication/authorization and used it for JDBC realm configuration.

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`
SQL SECURITY DEFINER VIEW `simplecrud_db`.`user_role_view` AS select
`simplecrud_db`.`user`.`USERNAME` AS `username`,
`simplecrud_db`.`user`.`PASSWORD` AS `password`,
`simplecrud_db`.`role`.`ROLENAME` AS `rolename`
from ((`simplecrud_db`.`user_roles` join `simplecrud_db`.`user`
on((`simplecrud_db`.`user_roles`.`User_userid` = `simplecrud_db`.`use
join `simplecrud_db`.`role`
on((`simplecrud_db`.`user_roles`.`Role_roleid` = `simplecrud_db`.`rol
```

For initial population

```
INSERT INTO `simplecrud_db`.`address`
(`ID`, `CITY`, `COUNTRY`, `STREET`,`SUBURB`)
VALUES
('1', 'city', 'country', 'street', 'suburb'),
('2', 'city', 'country', 'street', 'suburb'),
('3', 'city', 'country', 'street', 'suburb'),
('4', 'city', 'country', 'street', 'suburb'),
('5', 'city', 'country', 'street', 'suburb'),
('6', 'city', 'country', 'street', 'suburb'),
('7', 'city', 'country', 'street', 'suburb'),
('8', 'city', 'country', 'street', 'suburb'),
```

```
('9', 'city', 'country', 'street', 'suburb'),
('10', 'city', 'country', 'street', 'suburb'),
('11', 'city', 'country', 'street', 'suburb'),
('12', 'city', 'country', 'street', 'suburb'),
('13', 'city', 'country', 'street', 'suburb');

INSERT INTO `simplecrud_db`.`role`
(`ID`, `ROLEDESC`, `ROLENAME`)
VALUES
('1', 'Administrator', 'Administrators'),
('2', 'Manager', 'Managers'),
('3', 'User', 'Users');

INSERT INTO `simplecrud_db`.`user`
(`ID`, `EMAIL`, `FIRSTNAME`, `LASTNAME`, `PASSWORD`, `USERNAME`, `AD
VALUES
('1', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('2', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('3', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('4', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('5', 'e-mail', 'name', 'surnamer', '03ac674216f3e15c761ee1a5e255f06
('6', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('7', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('8', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('9', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f067
('10', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f06
('11', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f06
('12', 'e-mail', 'name', 'surname', '03ac674216f3e15c761ee1a5e255f06
('13', 'e-mail', 'name', 'surname', '8cc7a4de25920211fb328c8b3e3f7cc

INSERT INTO `simplecrud_db`.`user_roles`
(`Role_roleid`, `User_userid`)
VALUES
('1', '1'),
('3', '2'),
('3', '3'),
('3', '4'),
('3', '5'),
('3', '6'),
('3', '7'),
('3', '8'),
('3', '9'),
('3', '10'),
('3', '11'),
('3', '12'),
('3', '13');
```

Mapping entities for our tables.

User Entity after adding roles:

```
@Entity
public class User extends BaseEntity implements Serializable {

 @Column(nullable = false, length = 50)
 private String username;

 @Column(length = 50)
 private String firstname;

 @Column(length = 50)
 private String lastname;

 @Column(length = 50)
 private String email;

 @Column(length = 64)
 private String password;

 @OneToOne(cascade = {CascadeType.ALL})
 private Address address;

 @ManyToMany
 @JoinTable(name = "user_roles", joinColumns = {
 @JoinColumn(name = "User_userid")}, inverseJoinColumns = {
 @JoinColumn(name = "Role_roleid")})
 private List<Role> roles;

public User() {
 roles = new ArrayList<Role>();
 address = new Address();
 }

public String getUsername() {
 return this.username;
 }

public void setUsername(String username) {
 this.username = username;
 }

public String getFirstname() {
 return this.firstname;
 }

public void setFirstname(String firstname) {
```

```
 this.firstname = firstname;
 }

public String getLastname() {
 return this.lastname;
 }

public void setLastname(String lastname) {
 this.lastname = lastname;
 }

public String getEmail() {
 return this.email;
 }

public void setEmail(String email) {
 this.email = email;
 }

public String getPassword() {
 return this.password;
 }

public void setPassword(String password) {
 this.password = password;
 }

public Address getAddress() {
 return this.address;
 }

public void setAddress(Address address) {
 this.address = address;
 }

public List<Role> getRoles() {
 return roles;
 }

public void setRoles(List<Role> roles) {
 this.roles = roles;
 }
}
```

Role Entitiy:

```
@Entity
public class Role extends BaseEntity implements Serializable {

 private String roledesc;
 private String rolename;

public Role() {
 }

public Role(Integer roleid, String rolename) {
 this.rolename = rolename;
 }

@ManyToMany(mappedBy = "roles")
 private List<User> users;

public String getRoledesc() {
 return this.roledesc;
 }

public void setRoledesc(String roledesc) {
 this.roledesc = roledesc;
 }

public String getRolename() {
 return this.rolename;
 }

public void setRolename(String rolename) {
 this.rolename = rolename;
 }

public List<User> getUsers() {
 return users;
 }

public void setUsers(List<User> users) {
 this.users = users;
 }
}
```

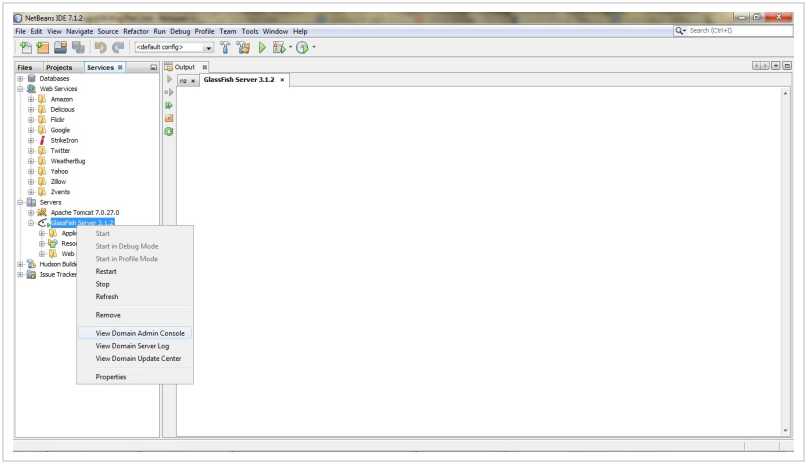
Address Entity will look the same:

```
@Entity
public class Address extends BaseEntity implements Serializable {

 @Column(length = 50)
 private String street;

 @Column(length = 50)
 private String suburb;
```

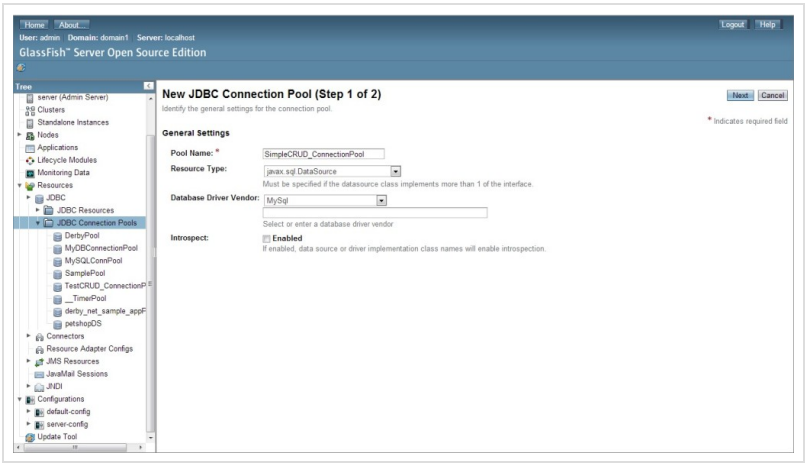
```
9
10    @Column(length = 50)
11    private String city;
12
13    @Column(length = 50)
14    private String country;
15
16   public Address() {
17    }
18
19   public String getStreet() {
20    return this.street;
21    }
22
23   public void setStreet(String street) {
24    this.street = street;
25    }
26
27   public String getSuburb() {
28    return this.suburb;
29    }
30
31   public void setSuburb(String suburb) {
32    this.suburb = suburb;
33    }
34
35   public String getCity() {
36    return this.city;
37    }
38
39   public void setCity(String city) {
40    this.city = city;
41    }
42
43   public String getCountry() {
44    return this.country;
45    }
46
47   public void setCountry(String country) {
48    this.country = country;
49    }
50  }
```

Let's create a connection pool for our database.

1-) Open your GlassFish Domain Admin Console by right clicking on GlassFish Server in NetBeans IDE or simply go to http://localhost:4848/common/index.jsf (GlassFish Server must be running)

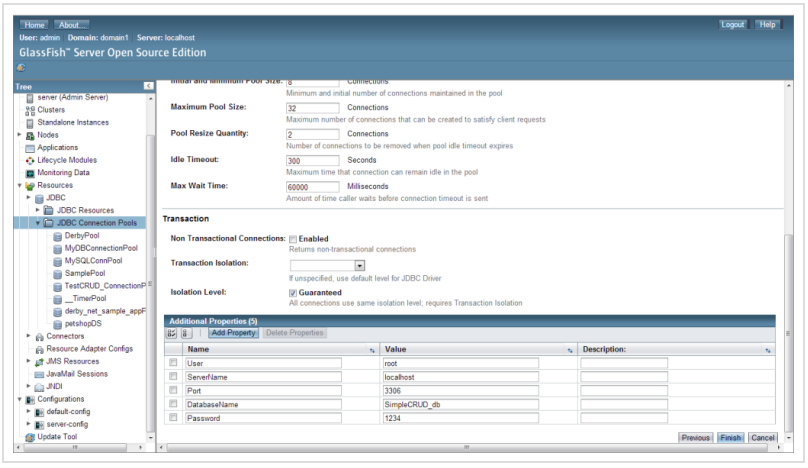


2-) Go to Common Tasks > Resources > JDBC > JDBC Connection Pools > New Connection Pool
3-) Name the connection pool (SimpleCRUD_ConnectionPool), Resource Type: javax.sql.DataSource, DataBase Driver Ventor: MySQL click next



Select all additional properties and delete except:

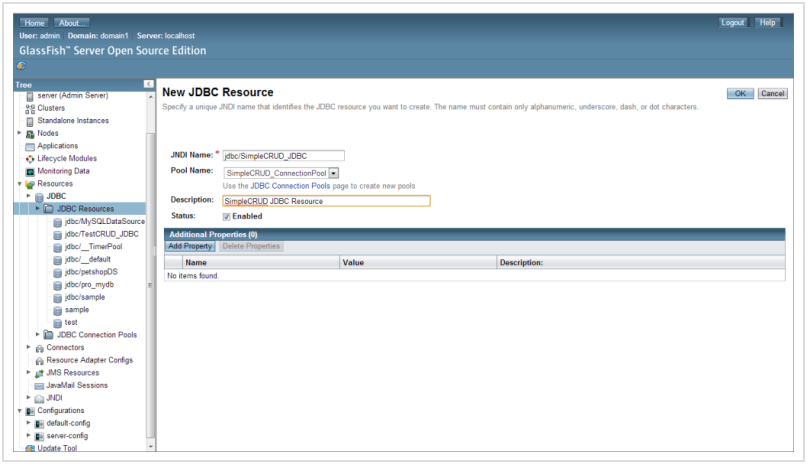
User you can put ***root*** here
Password root's password
ServerName localhost if your web application is not running on your local (127.0.0.1 is the default on standard Ubuntu server) in this case you will have to provide full url of your database server
DatabaseName SimpleCRUD_db
Port usually 3306 it is the default port.



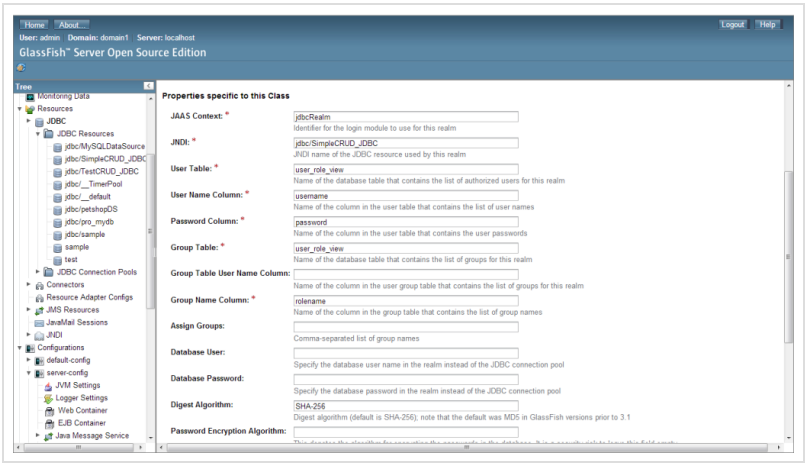
You need to save this configuration and ping to the database, if it succeeds there is no problem.

Next thing we need to do is to create a JDBC resource for our web application.

1-) Go to Common Tasks -> Resources -> JDBC -> JDBC Resources -> New
2-) Name your JNDI starting with 'jdbc/' ie (jdbc/SimpleCRUD_JDBC) and choose the connection pool you have just created click save (in this case it is SimpleCRUD_ConnectionPool).



Now we will create our security realm. Go to Common Task-> Configuration-> Server-Config-> Security-> Realms -> New.
- Name your realm SimpleCRUD_Realm
- Select com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm



JAAS Context: jdbcRealm
JNDI: jdbc/SimpleCRUD_JDBC -> whatever you named it
User Table: user_role_view
User Name Column: username
Password Column: password
Group Table: user_role_view
Group Table User Name Column:
Group Name Column: rolename
Assign Groups:
Database User:
Database Password:
Digest Algorithm: SHA-256
Password Encryption Algorithm:
Encoding:
Charset:UTF-8

Once you are done here, save it and we are ready to define our persistent unit in a persistence.xml file.

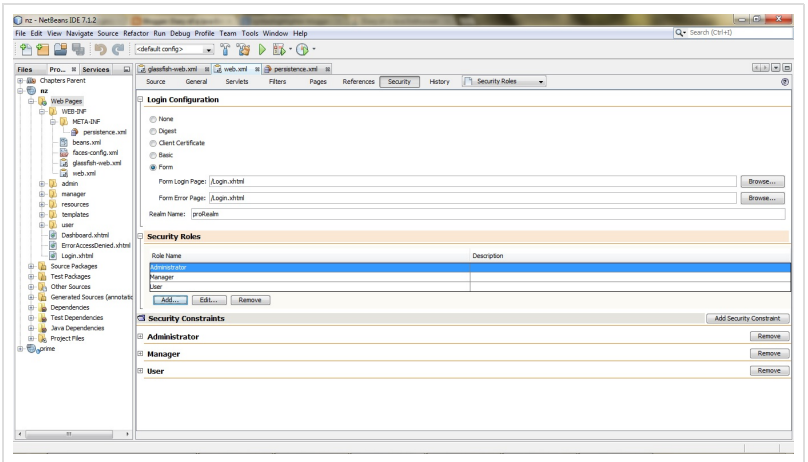
Let's go back to NetBeans IDE and check if there is no presistence.xml file in your web project you will need to add it by right clicking on project -> new -> persistence-unit.

persistence.xml file is going to look like:

```
+ expand source
```

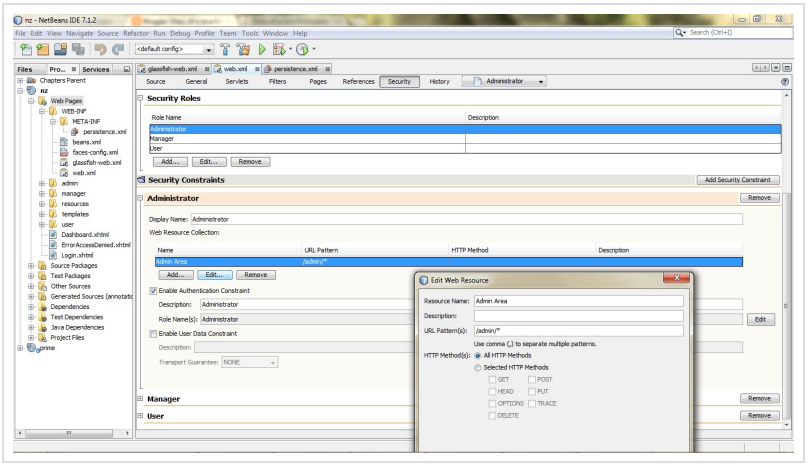
All our Entity Classes must be declared between class tags if you want to add them all by default then exclude-unlisted-classes = false. Eclipselink.ddl-generation option is to create tables if they don't exist in the database.

The next thing is to start configuring our security properties in web.xml file. We will go to Security / Login configuration section, here select "form" and type the file name. i.e. /Login.xhtml. Most importantly we will need to provide our previously created security realm name.



Now configure roles: Administrator, Manager, and User roles wil be declared in **Security Roles** section. (You can have different roles that is up to you, this is just for an example)

The next thing is to bind these roles to Security Constraints. As it is shown with the image, a security role and a corresponding url pattern need to be added along with other roles.



Our login form (Login.xhtml file).

```
+ expand source
```

Last thing we need to do is to define the security role mappings in glassfish-web.xml file.

It is a simple process:

Add Security Role Mapping

Type "*Administrator*" for security Role Name

and "*Administrators*" for Group Name.

Download the source code here
Please try live demo (Username: Admin, Password:1234)
Cheers Emre

---

24 thoughts on "Part 2 – Securing Web Application GlassFish V3 – JAAS (Authentication and Authorization)"

Reply

Dec 4, 2012 4:01 am

Kay Wrobel

Thanks for the great article. There are some inconsistencies on this page I'd like to point out: The ID columns in the ER-Diagram do not match the SQL script. In the script, each table's ID field is `ID` which matches the BaseEntity; the diagram shows columnnameid, which may have been a setting in your MySQL workbench. Also, in the script for the User table, the actual ID column `ID` and and PRIMARY KEY (`userid`) do not match (neither do the other tables). That would create an error when running the script, I'm sure. I am creating a diagram alongside and so am not copying and pasting the script you have. Just pointing those out.

Haven't read through the entire article just yet, but I love how the screen shots look on your introductory page. This is exactly what I have been looking for. I wish that NetBeans would incorporate PrimeFaces more into their CRUD generation. Obviously they'd have to rewrite this to be more like your pages

Reply

Dec 13, 2012 10:11 am

simtay

Hi Kay,
Sorry, I forgot to update the EER diagram image and SQL scripts, thanks for pointing it out. I have replaced the image with the up to date one. I will also deploy this application on a server so you can see it in a production environment.

Live demo (Username: Admin, Password:1234)

Cheers
Emre

Jan 3, 2013 8:56 am

techMagnificent

Reply

INSERT INTO `user` VALUES ('1', 'Admin',
'03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4',
'name', 'surname', 'name@domain.com', '1');
INSERT INTO `user` VALUES ('2', 'Manager',
'03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4',
'name', 'surname', 'name@domain.com', '2');
INSERT INTO `user` VALUES ('3', 'User',
'03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4',
'name', 'surname', 'name@domain.com', '3');

> Thanks for the great tutorial, This is exactly what I am looking for
> Can you please tell the values of the password entered into the database.
>
> Can you also post or extend the current tutorial to support Sorting...
> Thanks

Reply

Jan 4, 2013 9:02 am

simtay

The password is 1234 in SHA256-HEX format.
Thank you

Reply

Jan 8, 2013 8:29 pm

techMagnificent

Thanks Simtay, for the great tutorial, can you share pointers if we need to extend the current use-case with sorting and yes a little search kinda auto-complete

Thanks and Regards
techMagnificent

Reply

Jan 10, 2013 3:48 pm

simtay

Thank you too! I have added Part-6 that covers sorting. You can always look at Primefaces showcase if you need more examples (I am sure you already know that).

Cheers Emre

Reply

Jan 21, 2013 4:59 am

Constantin Popa

Hi,
I just can't make JAAS to work with your demo project.
Can't pass the authentication phase.
Could you please provide the exact versions of the software you're using: Glassfish, Manven, Netbeans, Mysql.
Thanks in advance.

Reply

Jan 21, 2013 2:05 pm

simtay

Please see http://www.simtay.com/simple-crud-web-application-with-jsf-2-1-primefaces-3-5-maven-and-jpa/ JAAS problems are related to Glassfish server. Please try to use Version 3.

Reply

Feb 24, 2013 9:15 pm

Chandan

Thanks for such a great article!!

While creating the MySQL connection-pool using GlassFish admin console, it reported two errors:
1. In MySQL server, 'root' user didn't have password so it raised error. User should ensure that 'password' field is not empty. That is, credential should have non-empty password.
2. It requires MySQL/J connector to be installed in GlashFish at glassfishdomainsdomain1libext if it is not already installed to avoid errors like "Class name is wrong or classpath is not set for :

com.mysql.jdbc.jdbc2.optional.MysqlDataSource Please check the server.log for more details.”

Reply

Feb 26, 2013 7:44 am

simtay

Hi Chandan,
1- Your root user has to have a password.
2- Your Glassfish is not configured properly. Make sure you have mysql-connector-java-5.1.13-bin.jar in your library folder. (i.e. C:\\Program Files\\glassfish-3.1.2.2\\glassfish\\domains\\domain1\\lib) If you need more help please e-mail me via emre.simtay@gmail.com.
Cheers

Reply

May 21, 2013 8:17 am

nerd

Does anybody make that code work with Postgresql ?
I have no idea how to pass the security/ authentication step. It still ends up with login problem on first screen.

Reply

May 21, 2013 11:48 am

simtay

Please download PostgreSQL java connector driver and place it in domain1/lib folder and make sure that you can ping your SQL server in GlassFish admin UI where u create a connection pool see Part 2 I have explained it how. If you have anymore questions please email me.

Reply

May 31, 2013 3:46 am

DPinup

Hi Emre,
And thanks for this wonderful article I found wandering around some security issues.
I downloaded your source code and Mabenly configured. Solved the project problems (however I configured to JDK_1.7 and Glassfish 3+) and database defined as proposed.

After that, Connection pool established and ping correctly checked. Finally Security Realm configured and ...

Application correctly starts but when trying to identify the User/pass, I got the message:

```
Error!
The username or password you provided does not match
our records.
```

The Glassfish Server 3+ log report shows:

```
INFO:
file:/C:/Users/XXXX/Documents/NetBeansProjects/Simtay-
SimpleCRUD-354999e/target/SimpleCRUD-1.0-
SNAPSHOT/WEB-INF/classes/_com.nz_war_1.0-SNAPSHOTPU
logout successful
INFO: EclipseLink, version: Eclipse Persistence
Services - 2.3.2.v20111125-r10461
INFO:
file:/C:/Users/XXXX/Documents/NetBeansProjects/Simtay-
SimpleCRUD-354999e/target/SimpleCRUD-1.0-
SNAPSHOT/WEB-INF/classes/_com.nz_war_1.0-SNAPSHOTPU
login successful
INFO: EJB5181:Portable JNDI names for EJB
UserService:
INFO: Inicializando Mojarra 2.1.6 (SNAPSHOT
20111206) para el contexto '/Simtay-SimpleCRUD-
354999e'
INFO: Running on PrimeFaces 3.4.2
INFO: Using OmniFaces version 1.4.1
INFO: WEB0671: Loading application [SimpleCRUD] at
[/Simtay-SimpleCRUD-354999e]
WARNING: WEB9102: Web Login Failed:
com.sun.enterprise.security.auth.login.common.LoginException:
```

```
Login failed: Fallo de conexión de archivo para
admin.
SEVERE: javax.servlet.ServletException: Exception
thrown while attempting to authenticate for user:
admin
```

I look for some solutions and I didn't find any, as It seems all problems were related to bad configurations of the realms and you did it correctly.
Any idea?
I also found that 1234 –> SHA-256 (not Hex)
"a883dafc480d466ee04e0d6da986bd78eb1fdd2178d04693723da3a8f95d42f4"
Didn't solve the problem neither

Reply

May 31, 2013 8:50 am

simtay

Someone else had this weird problem and I am assuming that you configured everything perfectly. I had a look at his project and everything was configured correctly but he was using different version of glassfish, I guess it was the problem!

What he did was to add last two properties to the config file, domain.xml (C:\Users******\.netbeans\7.1.2\config\GF3\domain1\config) or using Glassfish Administrator panel. But mine is working without these properties.

Reply

Aug 23, 2013 6:20 am

Marlon Freire

Hi,

I'm having this problem when I try to login .. I've looked all Realm configuration and everything is ok .. someone could help me solve this problem?

—

> WARNING: WEB9102: Web Login Failed: com.sun.enterprise.security.auth.login.common.LoginException: Login failed: Failed file login for admin.
> SEVERE: javax.servlet.ServletException: Exception thrown while attempting to authenticate for user: admin

—

best regards,

Marlon Freire

Reply

Aug 26, 2013 7:57 am

simtay

You must be missing something! You should read Part-2 over again! If you can't get it working email me the full error. Cheers

Reply

Sep 16, 2013 9:15 pm

Halil Karakose

Hi Simtay
During realm definition, I leave "Password Encryption Algorithm" section empty and glassfish gives me a warning:

"It is a security risk for not specifying the Password Encrypition Algorith" Continue?

What actually happens if I leave this field empty?

Reply

Sep 17, 2013 11:49 am

simtay

No sure, I have researched it before couldn't come up with anything. Please see this link...

Reply

hjk

Sep 18, 2013 1:32 am

Hi Simtay,
your demo is really great.
I followed all instructions, but I also don't manage to login with 'admin'/'1234' (ignore quotes) on GF 3.1.2.2 using MySQL 5.6
also with the above cited error message:

Error!
The username or password you provided does not match our records.

My console log says:

[#|2013-09-17T15:30:26.110+0200|WARNING|glassfish3.1.2|javax.enterprise.system.container.web.com.sun.web.security|_ThreadID=99;_ThreadName=Thread-2;|WEB9102: Web Login Failed: com.sun.enterprise.security.auth.login.common.LoginException: Login failed: Failed file login for admin.|#]

[#|2013-09-17T15:30:26.111+0200|SEVERE|glassfish3.1.2|com.nz.simplecrud.controller.LoginController|_ThreadID=99;_ThreadName=Thread-2;|javax.servlet.ServletException: Exception thrown while attempting to authenticate for user: admin|#]

Any help appreciated!!

best regards
Juergen

Reply

simtay

Sep 27, 2013 7:58 am

Can you please read part two and check that everything you have done correctly! Cheers

Reply

Can ATAC

Oct 2, 2013 11:04 pm

Hi,
Great tutorial simtay,

Just a little mistake in the web.xml realm-name : should be 'SimpleCRUD_Realm' instead of 'proRealm' to stay coherent with the GF realm configuration screenshot.

Cheers !

Reply

simtay

Oct 31, 2013 8:22 pm

Thanks mate

Reply

Caterpillar

Nov 20, 2013 5:54 am

Hi, are you aware of any nice tutorial that explains the same article but on Tomcat/TomEE platform?

Reply

simtay

Nov 20, 2013 6:57 am

No, sorry mate. I really don't know. But I will be making a tutorial on how to create exact on TomEE using Apache shiro maybe. Not sure.

Leave a Reply

Author (required)

Email (will not be published)(required)

Website

Just to prove you are a human, please answer the following math challenge either out loud

or typing into the text box
one × four =

b i link b-quote code close tags

Post Comment

« Part 1 – Creating Entity Classes With POJO + Annotation
Part 3 – Basic Implementation of CRUD »