

Package User

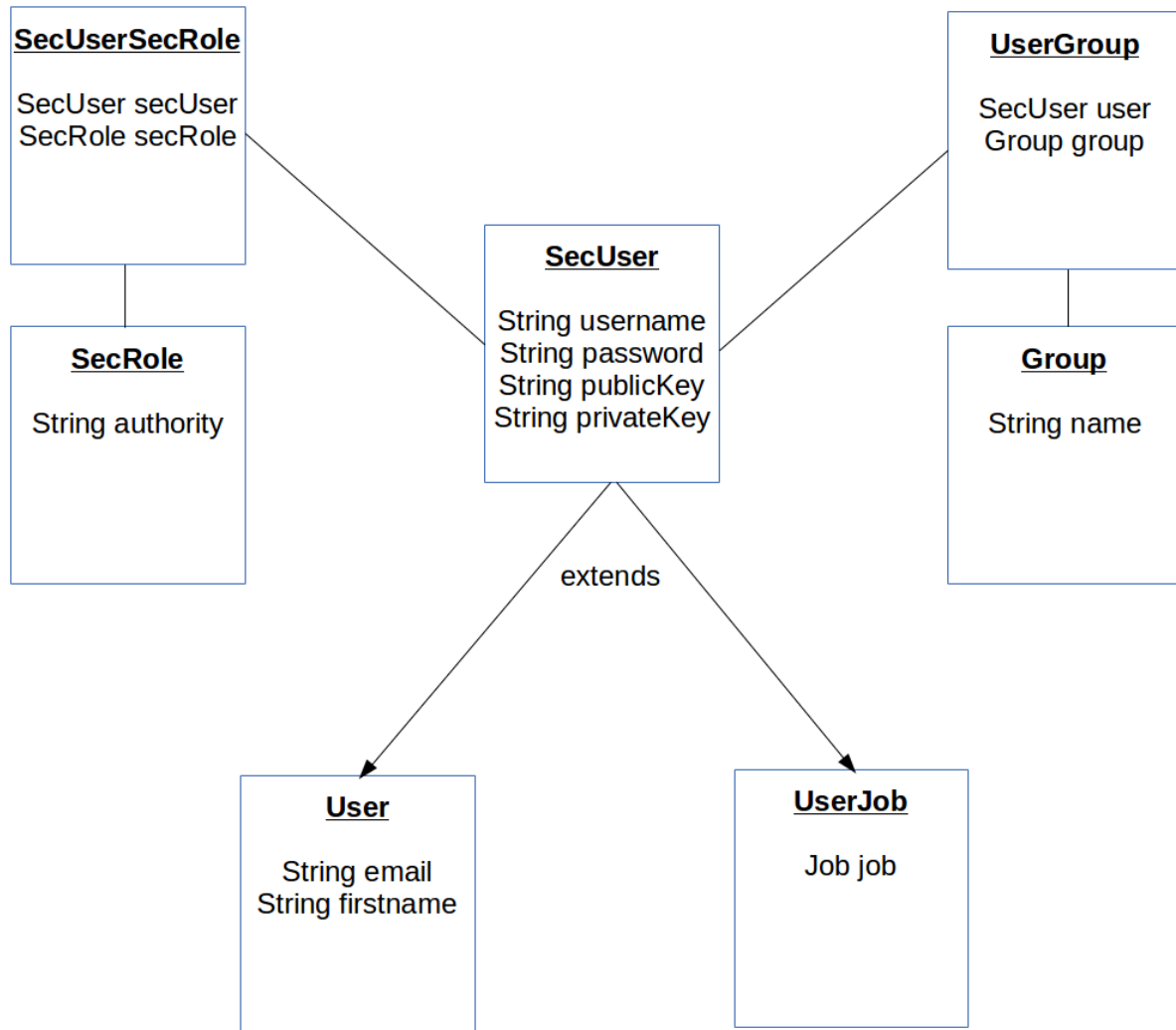
Package summary

There are two kind of User in Cytomine: human user (User) and software user (UserJob). Each of them extends the SecUser class. Each user has some roles (SecRole) and some permissions on domain instances.

To have more details about permissions, [check the permission guide](#).

Package main classes

Domain	Description	Main properties
SecUser	Generic user	String username String password String public/privateKey
User	Human user. Extend SecUser class	String firstname String lastname String email
UserJob	Software user. Extend SecUser class	Job job User user (human user that launch the job)
SecRole	A user role (admin, user,...)	String authority
SecUserSecRole	A link between a user and a role	SecUser secUser SecRole secRole
Group	A set of user	String name
UserGroup	A user in a group	SecUser user Group group



Package description

User

User and UserJob extends SecUser. User and UserJob are able to login to Cytomine. When a user run a job, this will create a new UserJob with the same user role.

A human user connects to the server with a login/password form.
 The form will POST data to the `/j_spring_security_check` service (from Spring Security plugin).

Sign in to Cytomine

Username

Password

Remember me

Sign in

Forgot your [username](#) or your [password](#) ?

A software user (*UserJob*) will use a client and will log thanks to public and private keys by generating a signature. The main code is located in `APIAuthenticationFilters.tryAPIAuthentication()`. This method will check if the signature is valid.

Role

There are 4 kinds of user:

- Guest (`ROLE_GUEST`): This is the lowest role. A guest cannot create project, ontologies, ... and has some specific restriction inside a project. Useful for students (context of education) or demo.
- User (`ROLE_USER`): A simple user allowing to create project and ontologies.
- Admin (`ROLE_ADMIN`): An admin has access to all data and may modify all data. This is a role to manage the platform. By default, an admin is connected as a user. The user must open an "admin session" to be a real admin.
- SuperAdmin (`ROLE_SUPER_ADMIN`): Same as admin, but everytime connected as an admin (not need to open "admin session"). Useful for external software user that needs the admin role.

The service `CurrentRoleService` is session scope service containing some interesting methods:

```
//Active admin session for user
def activeAdminSession(SecUser user)

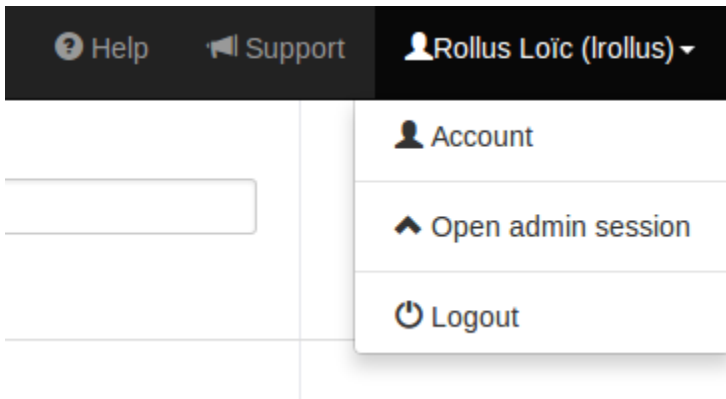
//Close admin session for user
def closeAdminSession(SecUser user)

//Get all active roles
Set<SecRole> findCurrentRole(SecUser user)

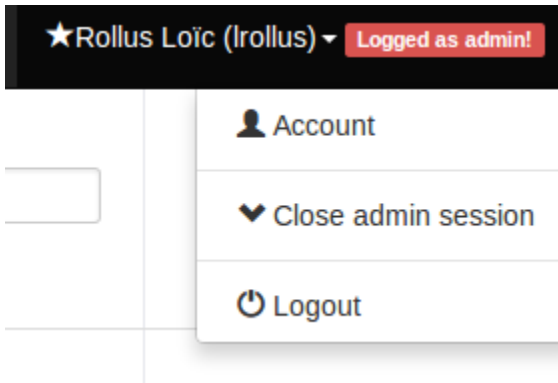
//Check if user is admin (with admin session opened)
boolean isAdminByNow(SecUser user) //isUserByNow and isGuestByNow too

//Check if user is admin (with admin session closed or opened)
boolean isAdmin(SecUser user) //isUser and isGuest too
```

A user with `ROLE_ADMIN` may open the Admin session on the web app. This service calls the `activeAdminSession` method.



This user will now be logged as an admin. He may the close its admin session. This service calls the closeAdminSession.



Connect As

An admin user is able to be connected as another user. This functionality simply calls the `/j_spring_security_switch_user` service (with POST param: `j_username:"xxx"`).