Comment gérer les droits avec sfGuard

En effet, la différenciation entre les droits, les groupes et les utilisateurs n'est pas toujours évidente. Je vous propose donc de découvrir comment gérer au mieux les droits avec sfGuard

Les utilisateurs

La pierre de base de l'accès à votre application. C'est l'utilisateur qui se connecte. Il est possible d'étendre la table de base de sfGuard (sfGuardUser), nous en reparlerons dans un autre article.

Un utilisateur est définit par un nom d'utilisateur (ou de login), un nom, un prénom. Il possède un mot de passe qui est haché dans la table.

Un utilisateur peut appartenir à un (ou plusieurs) groupe.
Un utilisateur peut se voir attribuer un (ou plusieurs) droit.

Le groupes

Fondamentalement, les groupes ne servent à rien (!). En effet on pourrait parfaitement ce passer des groupes et gérer les droits d'accès des utilisateurs. Mais sans eux, quel galère. En effet, si les droits sont correctement différenciés et associé à des actions, plus qu'à des modules, définir un profil d'utilisateur peut englober de nombreux droits. Et les définir individuellement pour chaque utilisateurs va poser rapidement un problème lourd de gestion. C'est ici que l'utilisation des groupes va se révéler payante, il nous suffit de créer des groupes qui reprennent des profils de droits, groupe auquel nous associerons des utilisateurs.

Les droits

Aussi appelé "crédentials" c'est ce que nous allons tester au niveau unitaire du code. C'est ce qui permet de dire si une action est possible ou pas. Un utilisateur qui possède un droit, qu'il l'ai reçu en directe ou par l'intermédiaire d'un groupe se verra attribuer l'autorisation d'effectuer une

action donnée.

Comment définir les droits d'un utilisateur ?

Les droits d'un utilisateurs sont le cumul de ces droits individuels et des droits obtenus par l'ensemble des droits des groupes des quels il est membre.

Et le droit super administrateur ?

sfGaurd permet d'attribuer un droit de super administrateur. Tout utilisateur qui possède ce droit va systèmatiquement répondre oui à toutes demandes de "a-t-il le droit"

Attention ! Si vous utiliser un droit pour interdire plutôt que pour utiliser, votre super admin n'aura jamais la possibilité d'effectuer l'action.

Mon changement d'authorisation n'est pas pris en compte

Les droits définis dans la base sont lu lors du loggin de l'utilisateur. Ils sont ensuite stocké dans la session utilisateur. Si vous modifier les droits d'un utilisateurs la nouvelle définition ne sera pas prise en compte avant le loggin suivant.

Comment définir les droits

D'expérience, n'hésitez pas à avoir des droits très précis (modification article, vue article, effacer article, modifier mon article,...). Vous avez alors la possibilité de regrouper des niveaux de droits au seins de groupes : utilisateur (qui pourra voir les article et modifier les siens) et rédacteur, (qui pourra modifier tous les articles).

Comment tester un droit dans le code d'une action

La méthode fait partie du sfGuardSecurityUseret étend la fonction de base de sfBasicSecurityUser. C'est la méthode ->hasCredential(DroitsATester, [ForceAnd]).

- DroitATester est soit un droit, le cas courant, soit un array des droits possibles, il suffit qu'un des droits soit valide pour que la fonction retourne true.
- ForceAnd est un paramètre optionnel qui n'a de sens que dans le cas où le premier

paramètre est un array, il permet de préciser que l'utilisateur doit avoir tous les droits définis dans le tableau pour que la fonction retourne true.

Exemple de vérification d'un droit pour une action d'édition :

```
  public function executeEdit ( sfWebRequest $request )
{
   ...
   $this->foward404until( $this->getUser()->hasCredential('EditArticle') );
   ...
}
```

Qui va envoyer un utilisateur tentant de modifier un article et sans les droits sur la page d'erreur 404 de l'application.

Test en utilisant le fichier de configuration security.yml

Il est possible de configurer les droits dans le fichier security.yml de chaque module. Cette fonction est dépréciée depuis la 1.3/1.4 mais encore parfaitement fonctionnel.

Pour la même configuration que ci-dessus on va mettre dans le fichier security.yml du module article

```
  # apps//modules/articles/security.yml

...
edit:
   credentials: EditArticle

default:
  is_secure: true
```

A noter qu'ici, on demande en plus que l'utilisateur soit identifié pour tous le module.

Voilà un rapide résumé des fonctions de gestion des droits de sfGuard.
Écris sur la version 1.4 de symfony et la version 5.0 de sfGuard.