Gestion des roles dynamiquement :

(ref : <http://stackoverflow.com/questions/11602337/how-to-change-role-hierarchy-storage-in-symfony2>)

The solution was simple. First I created a Role entity.

class Role

{

/\*\*

\* @var integer $id

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

private $id;

/\*\*

\* @var string $name

\*

\* @ORM\Column(name="name", type="string", length=255)

\*/

private $name;

/\*\*

\* @ORM\ManyToOne(targetEntity="Role")

\* @ORM\JoinColumn(name="parent\_id", referencedColumnName="id")

\*\*/

private $parent;

...

}

after that created a RoleHierarchy service, extended from the Symfony native one. I inherited the constructor, added an EntityManager there and provided an original constructor with a new roles array instead of the old one:

class RoleHierarchy extends Symfony\Component\Security\Core\Role\RoleHierarchy

{

private $em;

/\*\*

\* @param array $hierarchy

\*/

public function \_\_construct(array $hierarchy, EntityManager $em)

{

$this->em = $em;

parent::\_\_construct($this->buildRolesTree());

}

/\*\*

\* Here we build an array with roles. It looks like a two-levelled tree - just

\* like original Symfony roles are stored in security.yml

\* @return array

\*/

private function buildRolesTree()

{

$hierarchy = array();

$roles = $this->em->createQuery('select r from UserBundle:Role r')->execute();

foreach ($roles as $role) {

/\*\* @var $role Role \*/

if ($role->getParent()) {

if (!isset($hierarchy[$role->getParent()->getName()])) {

$hierarchy[$role->getParent()->getName()] = array();

}

$hierarchy[$role->getParent()->getName()][] = $role->getName();

} else {

if (!isset($hierarchy[$role->getName()])) {

$hierarchy[$role->getName()] = array();

}

}

}

return $hierarchy;

}

}

... and redefined it as a service:

<services>

<service id="security.role\_hierarchy" class="Acme\UserBundle\Security\Role\RoleHierarchy" public="false">

<argument>%security.role\_hierarchy.roles%</argument>

<argument type="service" id="doctrine.orm.default\_entity\_manager"/>

</service>

</services>

<services>

<service id="security.role\_hierarchy" class="Acme\UserBundle\Security\Role\RoleHierarchy" public="false">

<argument>%security.role\_hierarchy.roles%</argument>

<argument type="service" id="doctrine.orm.default\_entity\_manager"/>

</service>

</services>

That's all. Maybe, there is something unnecessary in my code. Maybe it is possible to write better. But I think, that main idea is evident now.