Tp 06 Sécurité et site administration

AMONA Birewa Audrey

27 novembre 2023

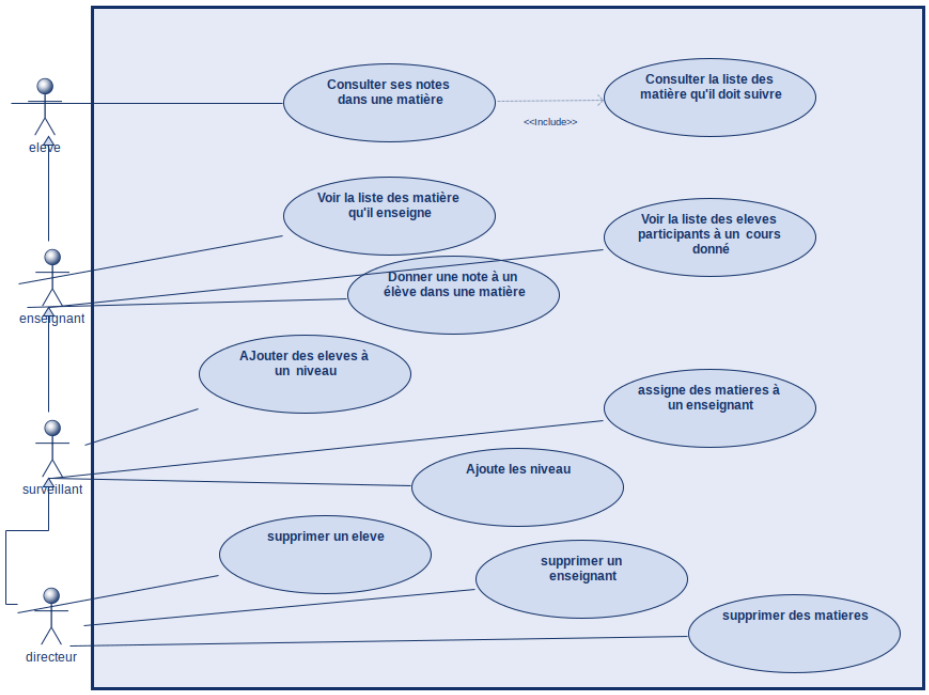
Table des matières

Résumé

Ce TP contient 3 parties indépendantes. D'abord nous nous pencherons sur l'authentification. Ensuite nous explorerons les problèmes de sécurité : CSRF, HTTPS, clé de chiffrement et détournement de clic. Enfin nous verrons comment customiser le site d'administration.

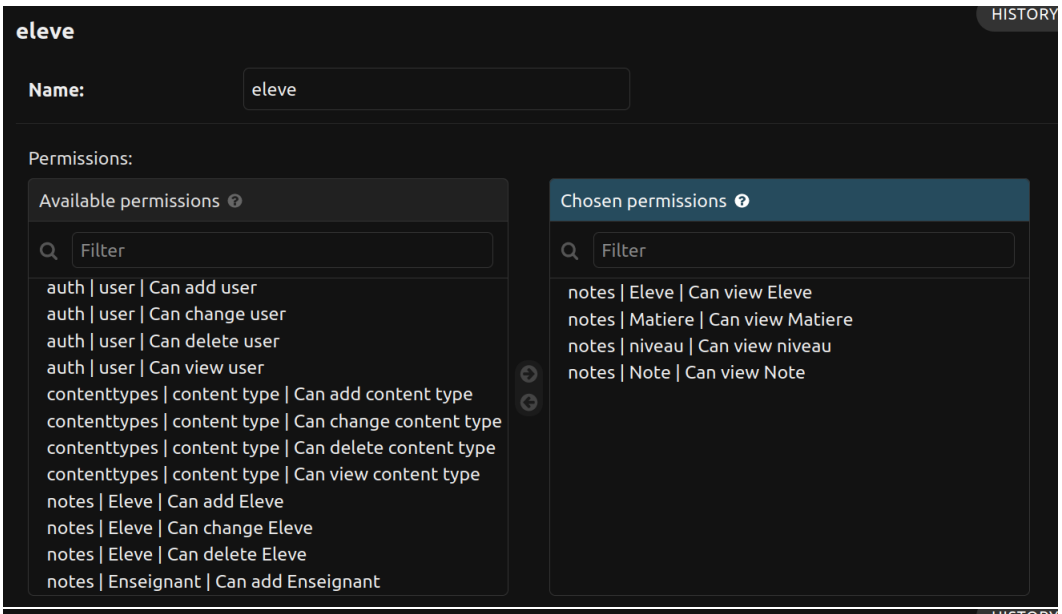
1 Authentification

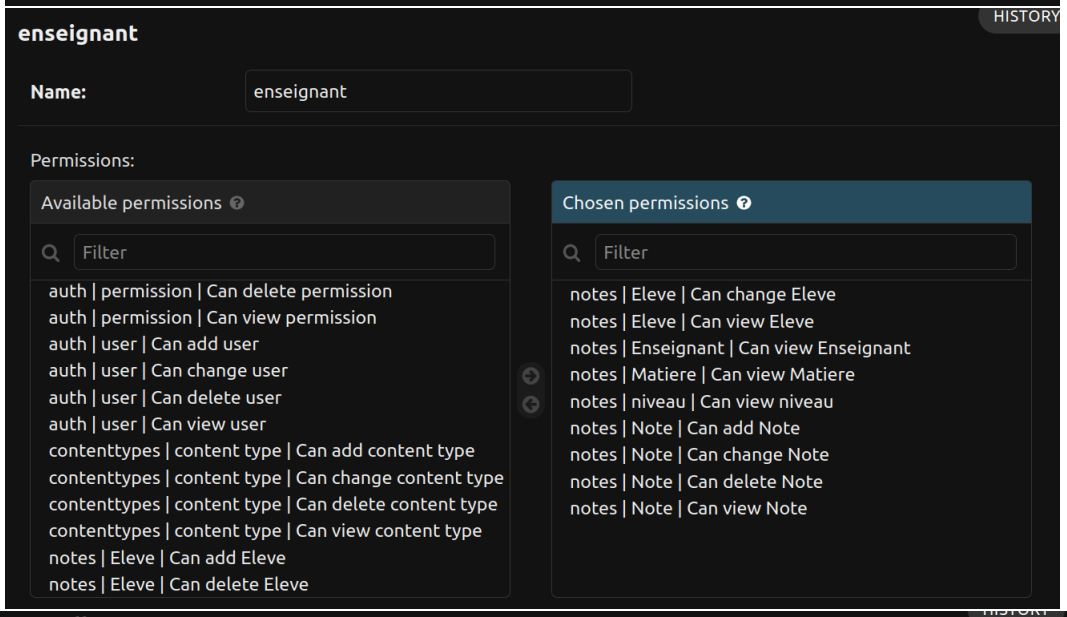
1. Dans le Tp précédent, nous avions spécifié les différentes fonctionnalités en fonction des utilisateurs du système :

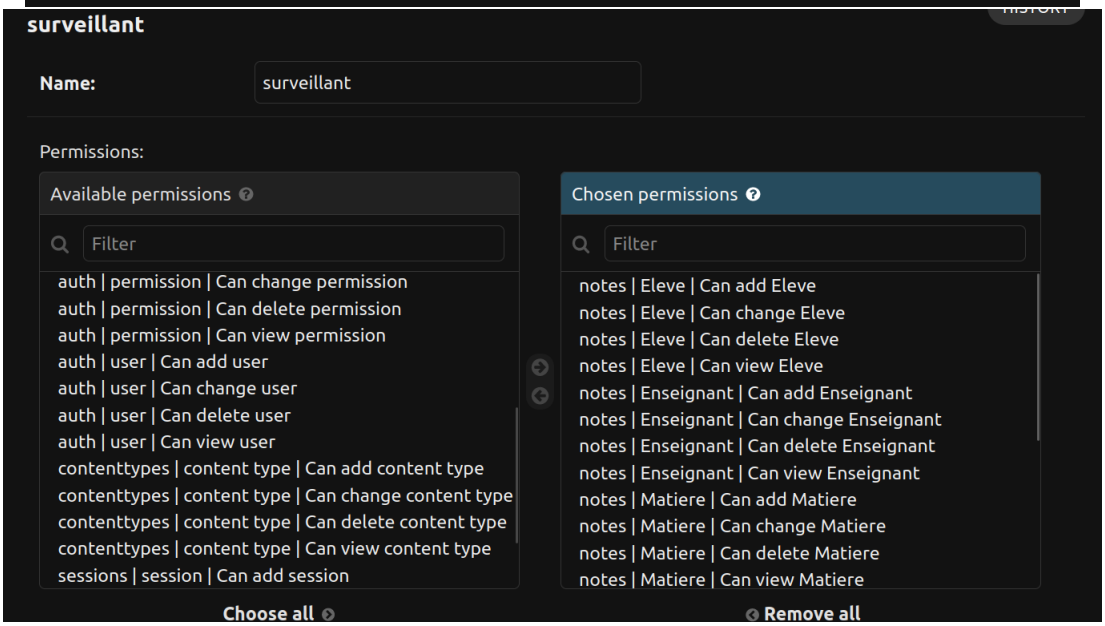


   Il étais convenu que seul les enseignant, les surveillants ou encore le directeur peuvent avoir accès au formulaire d'ajout de note.

2. Dans le site administration, nous remarquons qu'il y'a deux modèles de plus **Users** et **Groups**, que nous n'avons pas ajouté nous même. Le modèle **Users** possède les attributs suivant : username, lastname, firstname, email adress, staff, status. Ilpossède un enregistrement qui a pour username le nom de notre superutilisateur créé.
3. Nous allons créer des groupes avec les différents roles du diagramme de cas d'utilisation précédent.



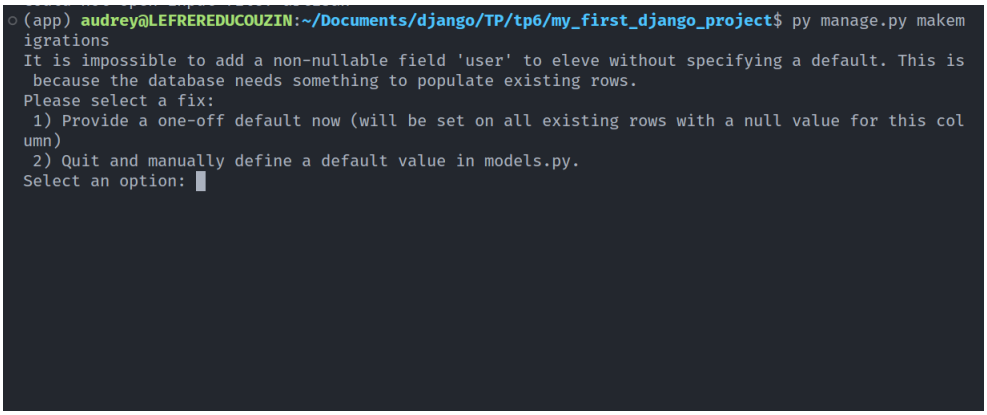




4. Django fournit une classe User qui contient l'ensemble des informations nécessaires à l'authentification et aux autorisations Nous pouvons étendre ce modèle sans modifier nos modèles utilisateurs par défaut soit :

— En créant une relation d'héritage entre notre modèle et le modèle par défaut **Users**
— En créant une relation OneToOne entre notre modèle et le modèle par défaut **Users**

5. Nous allons procéder à la création OneToOne entre les deux modèles, lorsqu'on lance **py manage.py makemigrations**, on a l'erreur suivante :

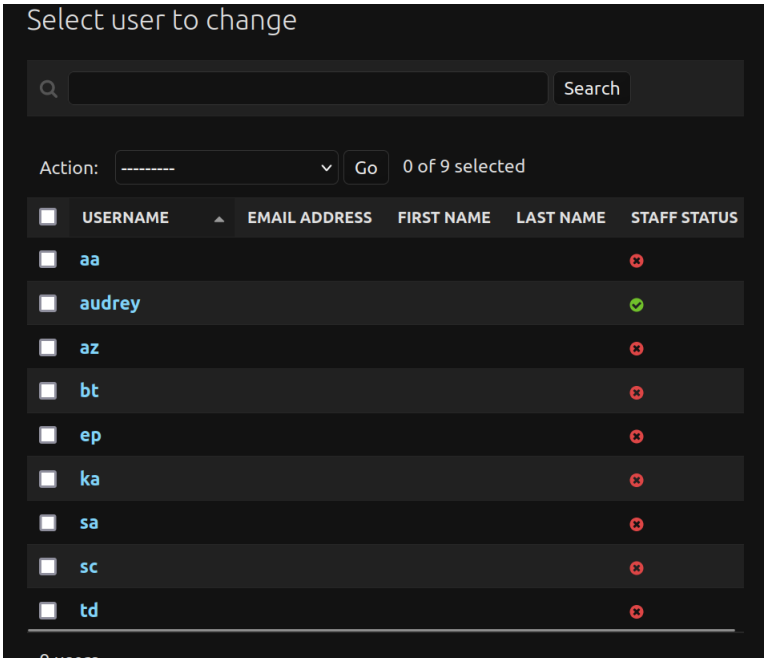


Ensuite on corrige le modèle en rendant l'attribut **user** nullable :

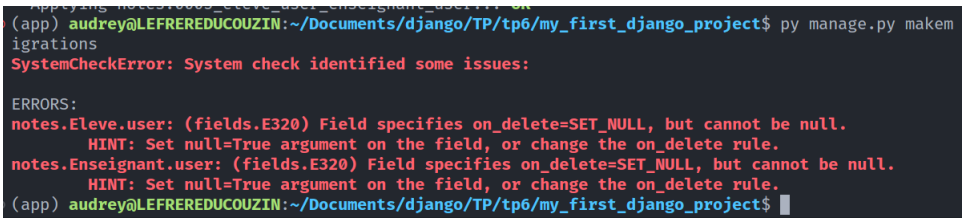
```python
class Personne(models.Model):
    nom = models.CharField(max_length=50)
    prenom = models.CharField(max_length=50)
    sexe = models.CharField(choices=(("M", "Masculin"), ("F","Feminin")),max_length=20)
    date_naissance = models.DateField()
    user = models.OneToOneField(User, on_delete=models.CASCADE, null=True)

    class Meta:
        abstract = True
```

6. Nous allons maintenant créer une instance de **User** pour chaque personne que nous avions créé :



7. Nous allons maintenant enlever la propriété nullable sur l'attribut précédemment ajouter. Losqu'on essaie de relancer les migration, on a le message suivant au terminal :


(app) audrey@LEFREREDUCOUZIN:~/Documents/django/TP/tp6/my_first_django_project$ py manage.py makem
igrations
SystemCheckError: System check identified some issues:
ERRORS:
notes.Eleve.user: (fields.E320) Field specifies on_delete=SET_NULL, but cannot be null.
HINT: Set null=True argument on the field, or change the on_delete rule.
notes.Enseignant.user: (fields.E320) Field specifies on_delete=SET_NULL, but cannot be null.
HINT: Set null=True argument on the field, or change the on_delete rule.
(app) audrey@LEFREREDUCOUZIN:~/Documents/django/TP/tp6/my_first_django_project$