

# ACCÈS RH POUR LES MANAGERS DU DÉPARTEMENT DRH

## 🎯 OBJECTIF

Permettre aux **managers du département DRH** d'accéder aux pages de validation RH, en plus des staff/superusers.

## 👥 QUI A ACCÈS AUX PAGES RH ?

Après cette modification :

1.  **Staff** (utilisateurs avec `(is_staff=True)`)
2.  **Superusers** (utilisateurs avec `(is_superuser=True)`)
3.  **Managers du département DRH** (nouveauté)

## 📦 INSTALLATION

### Étape 1 : Créer le fichier `decorators.py`

```
bash
```

```
cp /mnt/user-data/outputs/decorators_absence.py absence/decorators.py
```

### Étape 2 : Mettre à jour le context processor

```
bash
```

```
cp /mnt/user-data/outputs/context_processors_avec_drh.py absence/context_processors.py
```

### Étape 3 : Mettre à jour les vues dans `absence/views.py`

Ajouter l'import en haut du fichier :

```
python
```

```
from .decorators import rh_or_drh_manager_required, manager_required
```

Modifier la vue `rh_validation` :

```
python
```

```
# AVANT
@login_required
def rh_validation(request):
    if not (request.user.is_staff or request.user.is_superuser):
        messages.error(request, "Accès refusé.")
        return redirect('dashboard')
    # ...


```

```
# APRÈS
@rh_or_drh_manager_required
def rh_validation(request):
    # Le décorateur gère l'accès automatiquement
    # ...


```

Faire de même pour toutes les vues RH :

- `rh_validation`
- `rh_valider_demande`
- `rh_refuser_demande`
- Toute autre vue RH

#### Étape 4 : Vérifier la structure de votre département DRH

Le département DRH doit avoir :

- **CODE** = `'DRH'` ou `'RH'`
- **OU LIBELLE** contenant `'RH'`

Vérifier dans l'admin ou la base de données :

```
sql
SELECT CODE, LIBELLE FROM ZDDE WHERE CODE IN ('DRH', 'RH') OR LIBELLE LIKE '%RH%';
```

#### Étape 5 : Assigner un manager au département DRH

Dans l'admin Django ou via le shell :

```
python
```

```

from employee.models import ZY00
from departement.models import ZDDE, ZYMA
from datetime import date

# Récupérer le département DRH
dept_drh = ZDDE.objects.get(CODE='DRH')

# Récupérer l'employé qui sera manager
manager = ZY00.objects.get(nom='MARTIN') # Exemple

# Créer l'affectation manager
ZYMA.objects.create(
    manager=manager,
    departement=dept_drh,
    date_debut=date.today(),
    statut='actif'
)

```

## Étape 6 : Relancer le serveur

```

bash
python manage.py runserver

```

## 🔍 COMMENT ÇA FONCTIONNE ?

### Le décorateur `@rh_or_drh_manager_required`

```

python
@rh_or_drh_manager_required
def rh_validation(request):
    # Cette vue est accessible si :
    # 1. L'utilisateur est staff/superuser
    # 2. OU l'utilisateur est manager du département DRH
    pass

```

## Logique de vérification

1. Utilisateur connecté ?

- └ NON → Redirection vers login
- └ OUI → Continuer

2. Staff ou Superuser ?

- └ OUI →  Accès autorisé
- └ NON → Continuer

3. A un profil employé ?

- └ NON → X Accès refusé
- └ OUI → Continuer

4. Affectation active ?

- └ NON → X Accès refusé
- └ OUI → Continuer

5. Département = DRH ?

- └ NON → X Accès refusé
- └ OUI → Continuer

6. Est manager de ce département ?

- └ NON → X Accès refusé
- └ OUI → ✓ Accès autorisé

## TESTS

### Test 1 : Manager DRH peut accéder

1. Créer un utilisateur (ex: martin@example.com)
2. Créer un employé associé (Martin DUPONT)
3. Affecter au département DRH (ZYAF)
4. Définir comme manager du département (ZYMA)
5. Se connecter avec ce compte
6. Aller sur /absence/rh/validation/
  - ✓ Accès autorisé
  - ✓ Badge de notification RH visible

### Test 2 : Manager autre département ne peut pas

1. Créer un manager du département Comptabilité
2. Se connecter avec ce compte
3. Aller sur /absence/rh/validation/
  - X Message d'erreur
  - X Redirection vers dashboard

### Test 3 : Staff a toujours accès

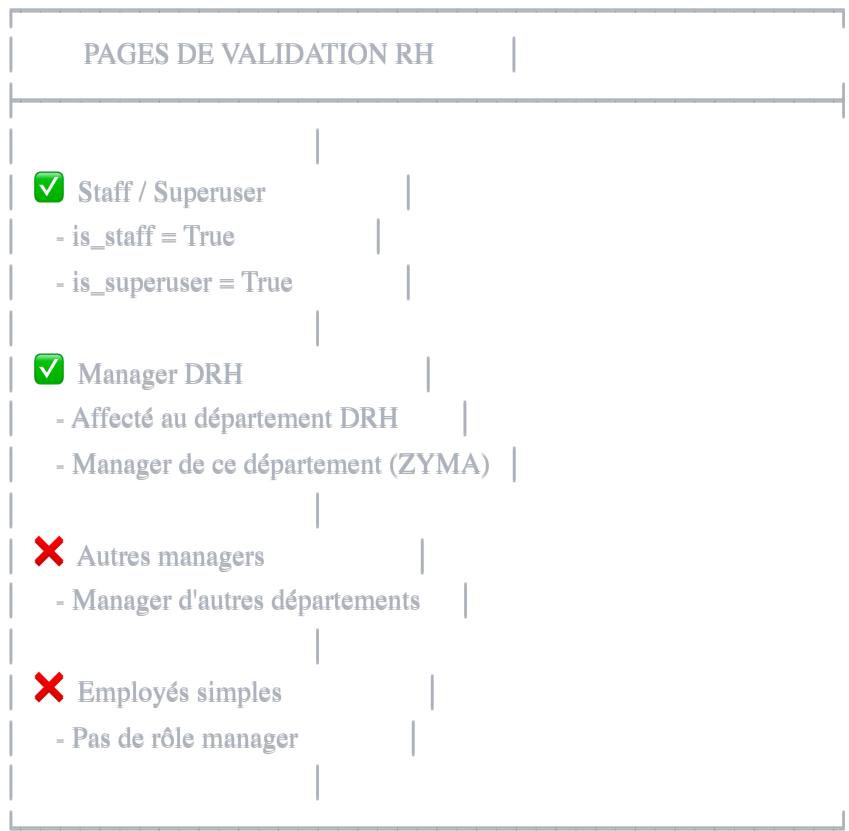
1. Se connecter avec un compte staff
2. Aller sur /absence/rh/validation/
  - ✓ Accès autorisé (comme avant)

### Test 4 : Notifications RH apparaissent

1. Créer une demande et la valider en tant que manager
2. Se connecter en tant que manager DRH

- Badge de notification (1)
- Section "En attente de validation RH"

## SCHÉMA DES DROITS



## PERSONNALISATION

### Changer le code du département

Si votre département RH a un autre code :

```

python

# Dans decorators.py, ligne ~40
if (departement.CODE and departement.CODE.upper() in ['DRH', 'RH', 'VOTRE_CODE']) or \
(departement.LIBELLE and 'RH' in departement.LIBELLE.upper()):

```

### Ajouter d'autres départements avec accès RH

```

python

# Exemple : Donner accès RH aux managers du département "Direction"
if (departement.CODE and departement.CODE.upper() in ['DRH', 'RH', 'DIR']) or \
(departement.LIBELLE and any(word in departement.LIBELLE.upper() for word in ['RH', 'DIRECTION'])):

```

### Vérifier l'accès dans les templates

```
django
```

```
{% if is_rh_user %}  
    <a href="{% url 'absence:rh_validation' %}" class="btn btn-primary">  
        Validation RH  
    </a>  
{% endif %}
```

## 🔧 FONCTIONS UTILITAIRES

### Vérifier si un utilisateur est RH

python

```
from absence.decorators import is_rh_or_drh_manager  
  
if is_rh_or_drh_manager(request.user):  
    # L'utilisateur a les droits RH  
    pass
```

### Dans les templates

Le context processor ajoute automatiquement :

django

```
{% if is_rh_user %}  
    <!-- Contenu visible uniquement pour RH -->  
    <div class="rh-section">  
        ...  
    </div>  
{% endif %}
```

## 🔒 SÉCURITÉ

### Plusieurs niveaux de protection

1. **Décorateur sur la vue** : Empêche l'accès direct à l'URL
2. **Context processor** : Cache les notifications
3. **Template** : Cache les liens (optionnel)
4. **Middleware** : Protection Django par défaut

### Messages d'erreur

python

# Si accès refusé

"Accès refusé. Cette page est réservée aux RH et aux managers du département DRH."

# Si pas d'employé associé

"Accès refusé. Vous n'avez pas de profil employé associé."



## EXAMPLE COMPLET

### Structure de la base de données

ZDDE (Département)

```
|-- CODE: 'DRH'  
|-- LIBELLE: 'Direction des Ressources Humaines'
```

ZY00 (Employé)

```
|-- nom: 'MARTIN'  
|-- prenoms: 'Sophie'
```

ZYAF (Affectation)

```
|-- employe: Sophie MARTIN  
|-- departement: DRH  
|-- statut: 'actif'
```

ZYMA (Manager)

```
|-- manager: Sophie MARTIN  
|-- departement: DRH  
|-- statut: 'actif'  
|-- date_fin: NULL
```

### Code de la vue

```
python
```

```

from .decorators import rh_or_drh_manager_required

@rh_or_drh_manager_required
def rh_validation(request):
    """
    Page de validation RH
    Accessible par staff, superuser ET managers DRH
    """

    demandes = ZDDA.objects.filter(
        statut='VALIDEE_MANAGER'
    ).select_related('employe', 'type_absence')

    return render(request, 'absence/rh_validation.html', {
        'demandes': demandes
    })

```

## CHECKLIST D'INSTALLATION

- Fichier decorators.py créé
- Context processor mis à jour
- Import ajouté dans views.py
- Décorateurs appliqués aux vues RH
- Département DRH vérifié (CODE ou LIBELLE)
- Manager DRH créé dans ZYMA
- Serveur relancé
- Test : Manager DRH → Accès autorisé
- Test : Autre manager → Accès refusé
- Test : Staff → Accès autorisé
- Test : Notifications RH visibles

## DÉPANNAGE

**Problème : Manager DRH n'a pas accès**

**Vérifications :**

```
python
```

```

# Dans le shell Django
from employee.models import ZY00
from departement.models import ZYAF, ZYMA

manager = ZY00.objects.get(nom='MARTIN')

# 1. Vérifier l'affectation
aff = ZYAF.objects.filter(employe=manager, statut='actif').first()
print(f"Département : {aff.departement.CODE if aff else 'Aucun'}")

# 2. Vérifier le rôle manager
manager_role = ZYMA.objects.filter(
    manager=manager,
    statut='actif',
    date_fin__isnull=True
).first()
print(f"Manager de : {manager_role.departement.CODE if manager_role else 'Aucun'}")

# 3. Les deux doivent être DRH

```

## Problème : Aucune notification RH

**Cause :** Le context processor n'identifie pas l'utilisateur comme RH

**Solution :** Vérifier que `context['is_rh_user']` est True dans le template

## RÉSULTAT FINAL

Après installation :

- **Staff/Superuser** : Accès RH (comme avant)
- **Manager DRH** : Accès RH (nouveau)
- **Notifications** : Badge pour les managers DRH
- **Sécurité** : Vérifications multi-niveaux
- **Flexible** : Facile d'ajouter d'autres départements
- **Messages clairs** : Erreurs explicites

**Système RH 100% fonctionnel avec support DRH ! 🚀**