GUIDE DE PARAMÉTRAGE DES RÔLES

Système de Gestion des Absences

Application:	ONIAN EasyM
Module:	Gestion des Absences
Version:	1.2
Date:	01/01/2026

TABLE DES MATIÈRES

1. INTRODUCTION

Le système de gestion des absences d'ONIAN EasyM repose sur un système de rôles permettant de contrôler finement les accès et les permissions de chaque utilisateur. Ce guide vous explique comment configurer et gérer ces rôles.

Objectifs de ce guide :

• Comprendre l'architecture des rôles

• Installer et configurer les rôles

• Attribuer les rôles aux employés

• Gérer les permissions associées

• Résoudre les problèmes courants

■■ **Note importante :** La gestion des rôles nécessite des droits administrateur. Assurez-vous d'avoir les permissions nécessaires avant de procéder aux modifications.

2. ARCHITECTURE DES RÔLES

Le système utilise 5 rôles principaux, chacun correspondant à un niveau de responsabilité différent dans le processus de gestion des absences :

Rôle	Code	Niveau	Description
Gestionnaire Application	GESTION_APP	Admin	Paramétrage complet
RH Validation	RH_VALIDATION_ABS	RH	Validation finale des absences
Manager	MANAGER_ABS	Manager	Validation niveau 1
Assistant RH	ASSISTANT_RH	RH	Consultation uniquement
Employé Standard	EMPLOYE_STD	Employé	Déclaration d'absences

2.1 Hiérarchie des Permissions

Les rôles suivent une hiérarchie stricte :

Niveau 5	GESTION_APP	Toutes les permissions
Niveau 4	RH_VALIDATION_ABS	Validation RH + consultation
Niveau 3	ASSISTANT_RH	Consultation uniquement (lecture seule)
Niveau 2	MANAGER_ABS	Validation manager + ses équipes
Niveau 1	EMPLOYE_STD	Ses propres absences uniquement

3. DESCRIPTION DÉTAILLÉE DES RÔLES

3.1 Rôle GESTION_APP (Gestionnaire Application)

Code : GESTION_APP
Groupe Django : GESTION_APP
Niveau : Administrateur

Description :

Ce rôle offre un accès complet à tous les paramétrages de l'application. Il est destiné aux administrateurs système et aux responsables de la configuration de l'application.

Permissions :

✓ Gestion complète des types d'absence

✓ Configuration des jours fériés

✓ Paramétrage des conventions de congés

✓ Configuration des paramètres de calcul

✓ Gestion des acquisitions de congés

✓ Paramétrage de l'entreprise

✓ Validation des absences (tous niveaux)

✓ Export et rapports complets

> ■■ **Attention :** Ce rôle donne un accès total au paramétrage. Ne l'attribuez qu'aux personnes de confiance ayant les compétences techniques nécessaires.

3.2 Rôle RH_VALIDATION_ABS (RH Validation)

Code : RH_VALIDATION_ABS
Groupe Django : RH_VALIDATION_ABS
Niveau : Ressources Humaines

Description :

Ce rôle permet de valider les absences après approbation du manager. Il représente la validation finale dans le workflow des absences.

Permissions :

✓ Validation finale des absences (niveau RH)

✓ Consultation de toutes les absences de l'entreprise

✓ Export des données d'absence

✓ Consultation des acquisitions de congés

✓ Accès aux rapports RH

Workflow :

1. L'employé soumet une demande
2. Le manager valide (niveau 1)
3. **Le RH valide (niveau 2)** ← **CE RÔLE**
4. L'absence est confirmée

3.3 Rôle MANAGER_ABS (Manager)

Code : MANAGER_ABS
Groupe Django : MANAGER_ABS
Niveau : Manager

Description :

Ce rôle permet aux managers de valider les demandes d'absence de leurs équipes. Il représente la première étape du processus de validation.

Permissions :

✓ Validation des absences de son équipe (niveau 1)

✓ Consultation des absences de son département

✓ Consultation de ses propres absences

✓ Création d'absences pour lui-même

> ■■ **Note :** Un manager ne peut valider que les absences des employés de son département. Cette restriction est gérée automatiquement par le système.

3.4 Rôle EMPLOYE_STD (Employé Standard)

Code : EMPLOYE_STD
Groupe Django : EMPLOYE_STD
Niveau : Employé

Description :

Ce rôle est attribué à tous les employés standards. Il permet de gérer ses propres absences uniquement.

Permissions :

✓ Création de demandes d'absence

✓ Consultation de ses propres absences

✓ Modification de ses absences (statut brouillon)

✓ Annulation de ses absences

✓ Consultation de son solde de congés

3.5 Rôle ASSISTANT_RH (Assistant RH)

Code : ASSISTANT_RH
Groupe Django : ASSISTANT_RH
Niveau : Ressources Humaines (Consultation)

Description :

Ce rôle permet de consulter toutes les absences de l'entreprise en mode lecture seule, sans pouvoir les valider ou les modifier. Il est destiné aux assistants RH qui ont besoin de visibilité sur les absences pour des tâches administratives.

Permissions :

✓ Consultation de toutes les absences (lecture seule)

✓ Consultation des types d'absence

✓ Consultation des acquisitions de congés

✓ Consultation des jours fériés

✓ Consultation des conventions de congés

✓ Accès aux rapports et statistiques

Restrictions :

✗ Ne peut PAS valider les absences

✗ Ne peut PAS modifier les absences

✗ Ne peut PAS créer d'absences pour d'autres employés

✗ Ne peut PAS gérer les paramètres de l'application

> ■■ **Note :** Ce rôle est idéal pour les assistants RH qui ont besoin de consulter les absences pour établir des rapports, suivre les plannings ou répondre aux questions des employés, sans avoir le pouvoir de validation.

4. INSTALLATION DES RÔLES

L'installation des rôles se fait via une commande Django personnalisée. Cette commande crée automatiquement tous les rôles et leurs permissions associées.

4.1 Prérequis

• Accès au serveur (SSH ou console locale)

• Droits administrateur sur la base de données

• Django installé et configuré

• Application 'employee' et 'absence' migrées

4.2 Procédure d'Installation

Étape 1 : Accéder au serveur

Connectez-vous au serveur via SSH ou ouvrez un terminal sur le serveur local :

```
$ ssh utilisateur@serveur
```

Étape 2 : Naviguer vers le projet

```
$ cd /chemin/vers/votre/projet
```

Étape 3 : Activer l'environnement virtuel

```
$ source venv/bin/activate
```

Étape 4 : Exécuter la commande

```
$ python manage.py create_absence_roles
```

Étape 5 : Vérifier le résultat

Vous devriez voir un message de confirmation similaire à :

```
■ Rôle GESTION_APP créé
■ Rôle RH_VALIDATION_ABS créé
■ Rôle MANAGER_ABS créé
■ Rôle EMPLOYE_STD créé
■ Rôle ASSISTANT_RH créé

■ Configuration des rôles terminée !
```

5. ATTRIBUTION DES RÔLES AUX EMPLOYÉS

Une fois les rôles créés, vous devez les attribuer aux employés. Plusieurs méthodes sont disponibles :

5.1 Via l'Interface d'Administration Django

1. Connectez-vous à l'admin Django : http://votresite.com/admin/
2. Naviguez vers **Employee** → **Attribution de rôle (ZYRE)**
3. Cliquez sur **Ajouter attribution de rôle**
4. Sélectionnez l'**employé**
5. Sélectionnez le **rôle** à attribuer
6. Définissez la **date de début**
7. Cochez **Actif**
8. Cliquez sur **Enregistrer**

5.2 Via le Shell Django (Méthode Avancée)

Pour les administrateurs avancés, vous pouvez utiliser le shell Django :

```
$ python manage.py shell

from employee.models import ZYRO, ZYRE, ZY00
from django.utils import timezone

# Récupérer le rôle
role = ZYRO.objects.get(CODE='ASSISTANT_RH')

# Récupérer l'employé
employe = ZY00.objects.get(matricule='MT000001')

# Créer l'attribution
ZYRE.objects.create(
    employe=employe,
    role=role,
    date_debut=timezone.now().date(),
    actif=True
)

print('■ Rôle attribué avec succès')
```

6. GESTION DES PERMISSIONS

6.1 Tableau Récapitulatif des Permissions

Action	GESTION_APP	RH	MANAGER	ASSISTANT_RH	EMPLOYÉ
Créer absence (soi)	✓	✓	✓	✓	✓
Voir ses absences	✓	✓	✓	✓	✓
Voir toutes absences	✓	✓	Équipe	✓	Soi
Valider (Manager)	✓	✗	✓	✗	✗
Valider (RH)	✓	✓	✗	✗	✗
Gérer types absence	✓	✗	✗	✗	✗
Gérer jours fériés	✓	✗	✗	✗	✗
Gérer conventions	✓	✗	✗	✗	✗
Paramètres entreprise	✓	✗	✗	✗	✗
Export données	✓	✓	✗	✓	✗

6.2 Cumul de Rôles

Un employé peut avoir plusieurs rôles simultanément. Dans ce cas, le système applique le principe du **cumul des permissions** :

Exemple 1 : Marie est EMPLOYÉ + ASSISTANT_RH

→ Elle peut créer ses absences ET consulter toutes les absences de l'entreprise

Exemple 2 : Jean est EMPLOYÉ + MANAGER + RH

→ Il peut créer ses absences, valider en tant que manager ET en tant que RH

Exemple 3 : Sophie est EMPLOYÉ + ASSISTANT_RH

→ Elle voit toutes les absences mais ne peut rien valider

■■ **Note :** Le système gère automatiquement les notifications pour éviter les doublons. Un employé avec plusieurs rôles recevra une notification pour chaque contexte.

7. DÉCORATEURS ET MIXINS

Pour protéger vos vues et contrôler l'accès selon les rôles, ONIAN EasyM fournit des décorateurs (pour les vues fonctions) et des mixins (pour les vues classes).

7.1 Décorateurs pour Vues Fonctions

Les décorateurs s'appliquent sur les vues définies avec `def`. Ils permettent de restreindre l'accès à certaines vues en fonction des rôles.

Décorateur	Rôles Autorisés	Usage Typique
`@drh_or_admin_required`	DRH, PDG, Admin	Fonctions DRH
`@gestion_app_required`	GESTION_APP, Admin	Paramétrage, Gestion rôles
`@assistant_rh_required`	ASSISTANT_RH, RH_VALIDATION_ABS, DRH, GESTION_APP, Admin	Gestion RH
`@rh_required`	RH_VALIDATION_ABS, DRH, GESTION_APP, Admin	Validation RH
`@manager_required`	MANAGER_ABS, Managers, DRH, GESTION_APP, Admin	Validation équipe
`@manager_or_rh_required`	Managers OU RH, Admin	Validation mixte
`@role_required(...)`	Personnalisable	Rôles multiples custom

Exemple d'utilisation :

```
@login_required
@drh_or_admin_required
def embauche_agent(request):
    # Code de la vue...
    pass
```

7.2 Mixins pour Vues Classes (CBV)

Les mixins s'appliquent sur les vues classes (héritant de ListView, UpdateView, etc.). Ils offrent une approche orientée objet pour la gestion des permissions.

Mixin	Rôles Autorisés	Usage Typique
`DRHOrAdminRequiredMixin`	DRH, GESTION_APP, Admin	CRUD employés
`GestionAppRequiredMixin`	GESTION_APP, Admin	Paramétrage application
`AssistantRHRequiredMixin`	ASSISTANT_RH, RH_VALIDATION_ABS, DRH, GESTION_APP, Admin	Gestion RH
`DRHOrAssistantRHRequiredMixin`	Tous rôles RH, Admin	Accès RH général
`ManagerRequiredMixin`	MANAGER_ABS, Managers, DRH, GESTION_APP, Admin	Gestion équipe

Exemple d'utilisation :

```
class EmployeListView(LoginRequiredMixin,
                      DRHOrAdminRequiredMixin,
                      ListView):
    model = ZY00
```

```
    template_name = 'employee/list.html'
```

7.3 Ordre des Mixins (Critique)

■■ **L'ordre des mixins est crucial pour le bon fonctionnement !** Respectez toujours cette hiérarchie :

```
class MaVue(LoginRequiredMixin,    # 1. Vérification connexion
            DRHOrAdminRequiredMixin,  # 2. Vérification permission
            ListView):               # 3. Type de vue Django
    pass
```

Règle d'or : Les mixins de permission viennent toujours APRÈS LoginRequiredMixin mais AVANT la vue de base (ListView, CreateView, UpdateView, etc.)

7.4 Décorateurs vs Mixins : Quand Utiliser Quoi ?

Critère	Décorateurs (@)	Mixins (Classe)
Type de vue	Vues fonctions (def)	Vues classes (CBV)
Syntaxe	@decorateur def ma_vue(request):	class MaVue(Mixin, View):
Complexité	Simple et direct	Plus structuré, POO
Réutilisabilité	Moyenne	Élevée (héritage)
Performance	Légèrement plus rapide	Optimisé pour CBV
Exemple	@drh_or_admin_required	DRHOrAdminRequiredMixin

■ **Recommandation :** Utilisez les mixins pour les vues classes (CBV) car ils offrent une meilleure intégration avec Django et facilitent la maintenance du code.

8. EXEMPLES D'UTILISATION

8.1 Cas d'Usage Typiques

Cas 1 : Employé Standard

Profil : Sophie, assistante administrative
Rôle attribué : EMPLOYE_STD
Permissions : Créer et consulter ses absences uniquement

Cas 2 : Manager de Département

Profil : Nicolas, chef du département IT
Rôles attribués : EMPLOYE_STD + MANAGER_ABS
Permissions : Créer ses absences + Valider les absences de son équipe IT

Cas 3 : Responsable RH

Profil : Nathalie, DRH
Rôles attribués : EMPLOYE_STD + RH_VALIDATION_ABS
Permissions : Créer ses absences + Valider toutes les absences (niveau RH)

Cas 4 : Assistant RH

Profil : Marie, assistante RH
Rôles attribués : EMPLOYE_STD + ASSISTANT_RH
Permissions : Créer ses absences + Consulter toutes les absences (lecture seule)

Cas 5 : Administrateur Système

Profil : Marc, responsable IT
Rôle attribué : GESTION_APP
Permissions : Accès complet au paramétrage de l'application

8.2 Workflow Complet

Étape	Acteur	Rôle Requis	Action
1	Sophie	EMPLOYE_STD	Crée une demande d'absence
2	Nicolas	MANAGER_ABS	Valide la demande (niveau 1)
3	Nathalie	RH_VALIDATION_ABS	Valide la demande (niveau 2)
4	Sophie	EMPLOYE_STD	Reçoit la confirmation
5	Marie	ASSISTANT_RH	Consulte pour établir un rapport

9. DÉPANNAGE

9.1 Problèmes Courants

Problème : L'employé ne peut pas valider les absences en tant que manager

Solutions :

• Vérifier que le rôle MANAGER_ABS est bien attribué

• Vérifier que l'attribution est active (date_fin = NULL)

• Vérifier que l'employé est bien défini comme manager du département dans ZYMA

• Vérifier que l'employé de la demande est bien dans le département du manager

Problème : L'employé ne voit pas les menus de paramétrage

Solutions :

• Vérifier que le rôle GESTION_APP est attribué

• Vider le cache du navigateur

• Vérifier les templates (condition `{% if user.employe.peut_gerer_parametrage_app %}`)

• Se déconnecter et se reconnecter

Problème : L'assistant RH peut valider des absences

Solutions :

• Vérifier que SEUL le rôle ASSISTANT_RH est attribué (pas RH_VALIDATION_ABS)

• Vérifier que les boutons de validation sont masqués dans le template

• Vérifier que les vues de validation utilisent bien les décorateurs @rh_required

• Re-exécuter la commande create_absence_roles pour corriger les permissions

9.2 Commandes de Diagnostic

Utilisez ces commandes pour diagnostiquer les problèmes :

```
# Vérifier les rôles existants
python manage.py shell
>>> from employee.models import ZYRO
>>> ZYRO.objects.all()

# Vérifier les attributions d'un employé
>>> from employee.models import ZY00
>>> emp = ZY00.objects.get(matricule='MT000001')
>>> emp.get_roles()

# Vérifier si un employé est assistant RH
>>> emp.est_assistant_rh()
```

```
# Vérifier les permissions d'un rôle
>>> role = ZYRO.objects.get(CODE='ASSISTANT_RH')
>>> role.django_group.permissions.all()
```

9.3 Contact Support

Si le problème persiste après avoir essayé ces solutions :

• **Email :** support@onian-easym.com

• **Téléphone :** +228 XX XX XX XX

• **Horaires :** Lundi - Vendredi, 8h - 17h