

# GUIDE DE BONNES PRATIQUES

## Gestion Électronique de Documents

### 1. Introduction à la GED et à HeelonGed

**GED (Gestion Électronique de Documents) :**

Une solution pour **stocker, organiser, partager et sécuriser** vos documents numériques. HeelonGed est une **API sécurisée** (développée en Rust/Axum) qui permet de gérer ces documents avec des fonctionnalités avancées comme le **versionnage, l'audit, et la conformité RGPD**.

---

### 2. Bonnes Pratiques Générales pour une GED

#### Organisation des Documents

- **Structure claire :**  
Utilisez une **arborescence logique** (ex: /Clients/Projet\_X/Contrats/2026). Évitez les dossiers trop profonds (> 5 niveaux).
- **Nommage des fichiers :**  
Adoptez une **convention de nommage** cohérente :
  - AAAA-MM-JJ\_NomDuDocument\_Version.ext (ex: 2026-05-05\_ContratClient\_v2.pdf).
  - Évitez les caractères spéciaux (!@#\$\$%^&\*) et les espaces (utilisez \_ ou -).
- **Métadonnées :**  
Renseignez les **tags, catégories et descriptions** pour faciliter les recherches.

#### Sécurité et Accès

- **Droits d'accès :**  
Appliquez le **principe du moindre privilège** : donnez uniquement les droits nécessaires (lecture, écriture, suppression).  
Utilisez les **rôles** (ex: administrateur, éditeur, lecteur) pour gérer les permissions.
- **Authentification :**  
Activez l'**authentification forte** (2FA) si disponible.  
Utilisez des **clés API sécurisées** (comme dans HeelonGed) pour les accès programmatiques.
- **Chiffrement :**  
Vérifiez que les documents sensibles sont **chiffrés au repos** (HeelonGed utilise SQLite avec des sauvegardes sécurisées).

## Versionnage et Historique

- **Versionnage automatique :**  
HeelonGed gère les versions des documents. **Ne modifiez jamais un fichier directement :** utilisez toujours l'API pour créer une nouvelle version.

- **Audit :**  
Consultez régulièrement les **logs d'audit** pour suivre les actions (création, modification, suppression) :

```
bash  
Copier
```

```
podman exec -it heelonged-backend sh -c "sqlite3 /app/data/db/ged.db  
'SELECT action, actor, document_id, version_id, created_at FROM audit_logs  
ORDER BY created_at DESC LIMIT 20;'"
```

## Suppression et Archivage

- **Suppression irréversible :**  
Dans HeelonGed, la suppression d'un document efface **toutes ses versions et fichiers associés**. Utilisez cette fonction avec prudence.  
Exemple de commande :

```
bash  
Copier
```

```
curl -s -X DELETE "$BASE/api/v1/dpo/docs/politique-rgpd" -H "X-DPO-API-  
Key: $DPO_KEY"
```

- **Archivage :**  
Archivez les documents obsolètes au lieu de les supprimer. Utilisez des **tags** comme archived ou deprecated.

## Recherche et Récupération

- **Moteur de recherche :**  
Utilisez les **métadonnées et mots-clés** pour optimiser les recherches.  
Exemple : Recherchez par slug, version\_id, ou created\_at.
- **Backup :**  
Effectuez des **sauvegardes régulières** de la base de données et des fichiers :

```
bash  
Copier
```

```
cp /srv/heelonged-backend/db/ged.db /srv/heelonged-backend/db/ged.db.$  
(date +%Y%m%d)
```

### 3. Spécificités et Bonnes Pratiques pour HeelonGed

#### Configuration et Déploiement

- **Environnement :**

Configurez correctement les variables dans `prod.env` :

```
text  
Copier
```

```
DATABASE_URL=sqlite:///app/data/db/ged.db
```

```
STORAGE_PATH=/app/data/files
```

```
MAX_UPLOAD_MB=10 # Limitez la taille des uploads pour éviter les abus
```

```
LOG_DIR=/app/logs
```

- **Déploiement :**

Utilisez le **pipeline GitHub Actions** pour les mises à jour automatiques.

Pour un déploiement manuel, utilisez :

```
bash  
Copier
```

```
bash deploy.sh --image ghcr.io/ppaperso/heelonged-backend@sha256:<digest>
```

Le script `deploy.sh` effectue :

1. Une **validation des prérequis**.
2. Un **pull de l'image cible**.
3. Une **sauvegarde automatique** de la base et des fichiers.
4. Un **redémarrage du conteneur**.
5. Un **test de santé** (healthcheck) et un **rollback automatique** en cas d'échec.

#### Gestion des Versions

- **Rollback :**

Pour revenir à une version précédente, mettez à jour `current_version_id` en base :

```
bash  
Copier
```

```
podman exec -it heelonged-backend sh -c "sqlite3 /app/data/db/ged.db  
\nUPDATE documents SET current_version_id = '$PREV_VERSION_ID' WHERE slug  
= '$DOC_SLUG';\n"
```

*(Un endpoint API pour cette opération pourrait être ajouté ultérieurement.)*

#### Gestion des Clés API

- **Création et Révocation :**

Les clés API doivent être **sécurisées** et **révoquées** en cas de compromission.

Exemple de révocation :

```
bash
Copier
```

```
curl -s -X DELETE "$BASE/api/v1/dpo/api-keys/$KEY_ID/revoke" -H "X-DPO-API-Key: $DPO_KEY"
```

Réponse attendue :

```
json
Copier
```

```
{
  "key_id": "550e8400-e29b-41d4-a716-446655440000",
  "revoked_at": "2026-04-22T14:00:00Z"
}
```

## Conformité et Audit

- **Vérification de la conformité :**

Exécutez régulièrement l'audit intégré :

```
bash
Copier
```

```
sudo bash /usr/local/sbin/heelonged-backend-audit
```

- **Logs :**

Surveillez les logs en temps réel :

```
bash
Copier
```

```
sudo podman logs -f heelonged-backend
```

## Sauvegardes

- **Automatisez les sauvegardes :**

Sauvegardez la base (`ged.db`) et le dossier `files` quotidiennement.

Exemple :

```
bash
Copier
```

```
cp /srv/heelonged-backend/db/ged.db /srv/heelonged-backend/backups/ged.db.
$(date +%Y%m%d)
```

## 4. Checklist pour les Utilisateurs

Action	Fréquence	Responsable
Vérifier les droits d'accès	Mensuelle	Administrateur
Sauvegarder la base et les fichiers	Quotidienne	Automatique + Admin
Consulter les logs d'audit	Hebdomadaire	Administrateur
Mettre à jour les clés API	Selon besoin	Administrateur
Tester le rollback d'une version	Avant mise en prod	Développeur

## 5. Ressources et Support

- **Documentation technique** : Consultez le README .md pour les détails de déploiement et de configuration.
- **Support** : En cas de problème, vérifiez les logs (podman logs) ou contactez l'équipe technique.
- **Formations** : Organisez des sessions pour former les utilisateurs aux bonnes pratiques.

---

## Conseils Supplémentaires

- **Testez toujours** les mises à jour en environnement de **staging** avant de déployer en production.
- **Limitez les uploads** (MAX\_UPLOAD\_MB=10) pour éviter les abus et saturer le stockage.
- **Utilisez des conteneurs** (Podman) pour isoler HeelonGed et faciliter les mises à jour.