

Guide de Démarrage Rapide - Prisma Workflow

Ce guide vous permet de démarrer rapidement avec le système de gestion d'interventions techniques.

Installation Express (5 minutes)

Prérequis

- Node.js 20+ installé
- PostgreSQL 14+ installé et en cours d'exécution
- Git installé







Installation Automatique

Bash

```
# 1. Cloner le projet
git clone <repository-url>
cd prisma-workflow

# 2. Exécuter le script de configuration automatique
./scripts/database-setup.sh
```

Le script configure automatiquement :

-  Base de données PostgreSQL
-  Variables d'environnement
-  Dépendances npm
-  Client Prisma
-  Migrations de base de données
-  Données d'exemple

Vérification de l'Installation

Bash

```
# Démarrer l'application
npm run dev
```

```
# Ouvrir Prisma Studio (dans un autre terminal)
npm run db:studio
```

Premiers Pas

1. Explorer les Données

Ouvrez Prisma Studio dans votre navigateur (<http://localhost:5555>) pour explorer :

- **Utilisateurs** : Administrateurs, managers, techniciens
- **Clients** : Entreprises clientes
- **Missions** : Interventions planifiées
- **Matériel** : Inventaire et sorties

2. Tester les Services

Bash

```
# Exécuter les exemples d'utilisation
npm run dev
```

L'application affiche automatiquement :

- Statistiques de la base de données
- Liste des utilisateurs avec pagination
- Détails des missions et interventions
- Exemples de notifications et d'audit

3. Créer Votre Première Mission

TypeScript

```
import { MissionService } from './src/services/mission.service';

const nouvelleMission = await MissionService.createMission({
  numIntervention: "INT-2024-TEST",
  natureIntervention: "Test d'installation",
  objectifDuContrat: "Test du système",
  priorite: "normale",
  dateSortieFicheIntervention: new Date(),
```

```
  clientId: 1 // ID du client existant
});
```

Commandes Essentielles

Base de Données

Bash

```
# Générer le client Prisma après modification du schéma
npm run db:generate

# Créer une nouvelle migration
npm run db:migrate

# Réinitialiser la base (développement uniquement)
npm run db:reset

# Ouvrir l'interface graphique
npm run db:studio
```

Sauvegarde

Bash

```
# Créer une sauvegarde
./scripts/database-backup.sh backup

# Lister les sauvegardes
./scripts/database-backup.sh list

# Restaurer une sauvegarde
./scripts/database-backup.sh restore nom-fichier.sql.gz
```

Développement

Bash

```
# Mode développement avec rechargement automatique
npm run dev

# Compiler TypeScript
npm run build
```

```
# Démarrer en production  
npm start
```

Configuration Rapide

Variables d'Environnement

Modifiez `.env` selon vos besoins :

Plain Text

```
# Base de données  
DATABASE_URL="postgresql://user:password@localhost:5432/intervention_dev"  
  
# Application  
NODE_ENV=development  
PORT=3000  
  
# Sécurité  
JWT_SECRET=your-secret-key  
BCRYPT_ROUNDS=12
```

Personnalisation du Schéma

1. Modifiez `prisma/schema.prisma`
2. Créez la migration : `npm run db:migrate`
3. Générez le client : `npm run db:generate`

Prochaines Étapes

1. **Explorez la Documentation** : Lisez `README.md` pour une compréhension complète
2. **Personnalisez les Services** : Adaptez les services à vos besoins métier
3. **Ajoutez des Tests** : Implémentez des tests pour vos modifications
4. **Configurez le Déploiement** : Préparez l'environnement de production

Résolution de Problèmes

PostgreSQL ne démarre pas

Bash

Ubuntu/Debian

```
sudo systemctl start postgresql
```

macOS

```
brew services start postgresql
```

Erreur de connexion à la base

1. Vérifiez que PostgreSQL est démarré
2. Vérifiez l'URL dans `.env`
3. Vérifiez les permissions utilisateur

Erreur de migration

Bash

Réinitialiser les migrations (développement uniquement)

```
npm run db:migrate:reset
```

Client Prisma obsolète

Bash

Régénérer le client

```
npm run db:generate
```



Ressources

- [Documentation Complète](#)
- [Exemples d'Utilisation](#)
- [Services Disponibles](#)
- [Scripts Utilitaires](#)

Besoin d'aide ? Consultez la documentation complète ou ouvrez une issue sur GitHub.