

# Utiliser Ruby on Rails

Pendant mon temps chez ShowMeBug, j'ai contribué au projet d'intégration avec WeChat Entreprise. Cela impliquait d'intégrer ShowMeBug avec WeChat Entreprise, offrant un accès sans couture aux outils d'entretien technique au sein de l'écosystème WeChat Entreprise. J'ai utilisé des technologies telles que Ruby, Ruby on Rails, PostgreSQL et le SDK WeChat pour créer une expérience utilisateur fluide pour les interviewers et les candidats.

Cet article de blog a été composé avec l'assistance de l'IA autour de février 2025.

---

Ruby on Rails (souvent simplement "Rails") est un puissant framework de développement web construit sur le langage de programmation Ruby. Il est conçu pour rendre la construction d'applications web rapide et agréable en mettant l'accent sur les conventions plutôt que sur la configuration et les principes DRY (Don't Repeat Yourself). Suivons les étapes pour l'installer et créer une application simple.

**Étape 1: Installer Ruby et Rails** Tout d'abord, vous aurez besoin d'installer Ruby, car Rails est une gemme (bibliothèque) Ruby. La plupart des systèmes ne viennent pas avec Ruby préinstallé, voici comment le configurer :

- **Sur macOS/Linux :**

- Utilisez un gestionnaire de versions comme `rbenv` ou `rvm` pour la flexibilité. Installez-le via Homebrew (`brew install rbenv`), puis exécutez :

```
rbenv install 3.2.2 # Une version stable de Ruby en 2025  
rbenv global 3.2.2
```

- Installez Rails :

```
gem install rails
```

- **Sur Windows :**

- Utilisez RubyInstaller (téléchargez depuis rubyinstaller.org). Choisissez une version comme 3.2.2 avec DevKit.
- Après avoir installé Ruby, ouvrez une invite de commande et exécutez :

```
gem install rails
```

Vérifiez l'installation :

```
ruby -v # Doit afficher quelque chose comme ruby 3.2.2  
rails -v # Doit afficher la dernière version de Rails, par exemple 7.1.x
```

**Étape 2: Créez un Nouveau Projet Rails** Une fois Rails installé, générez une nouvelle application :

```
rails new myapp --database=sqlite3  
cd myapp
```

Cela crée un dossier appelé `myapp` avec une structure Rails complète, utilisant SQLite comme base de données par défaut (idéal pour le développement).

**Étape 3: Démarrer le Serveur** Exécutez le serveur Rails intégré :

```
rails server
```

Ouvrez votre navigateur à `http://localhost:3000`. Vous verrez une page de bienvenue. Félicitations, votre application Rails est en cours d'exécution !

**Étape 4: Construire Quelque Chose de Simple** Créez une fonctionnalité de base "Posts" pour comprendre le modèle MVC (Modèle-Vue-Contrôleur) de Rails.

- **Générer un Modèle et un Contrôleur :**

```
rails generate scaffold Post title:string body:text
```

Cela crée un modèle `Post`, une migration de base de données, un contrôleur et des vues—tout interconnecté.

- **Exécuter la Migration :**

```
rails db:migrate
```

Cela configure la table de base de données pour les posts.

- **Vérifiez-le :** Redémarrez le serveur (`rails server`) et visitez `http://localhost:3000/posts`. Vous verrez une interface CRUD pour créer, lire, mettre à jour et supprimer des posts.

**Étape 5: Explorer les Concepts Clés**

- **Routes :** Ouvrez `config/routes.rb`. Vous verrez `resources :posts`, qui génère automatiquement des routes RESTful comme `/posts/new` ou `/posts/1/edit`.
- **Contrôleurs :** Regardez `app/controllers/posts_controller.rb`. Il gère les requêtes et les réponses.
- **Vues :** Vérifiez `app/views/posts/`. Ce sont des modèles ERB (HTML avec Ruby intégré).
- **Modèles :** Consultez `app/models/post.rb`. Il est lié à la base de données et peut inclure des validations (par exemple, `validates :title, presence: true`).

## Étape 6: Personnaliser et Déployer

- Ajoutez un peu de style avec du CSS dans `app/assets/stylesheets/`.
- Pour la production, passez à une base de données comme PostgreSQL (`rails new myapp --database=postgresql`) et déployez sur une plateforme comme Render ou Heroku. Mettez à jour `Gemfile` avec `gem "pg"` et exéutez `bundle install`.

### Conseils Pro

- Utilisez `rails console` pour expérimenter avec vos modèles en temps réel.
- Exécutez `rails generate --help` pour voir tous les raccourcis offerts par Rails.
- Utilisez des gems comme `devise` pour l'authentification ou `pundit` pour l'autorisation—ajoutez-les à votre `Gemfile` et configurez-les selon vos besoins.

C'est tout ! Vous avez une application Rails de base en cours d'exécution. À partir de là, explorez les guides officiels Rails ([guides.rubyonrails.org](http://guides.rubyonrails.org)) ou construisez quelque chose de réel pour consolider vos compétences. Quel genre de projet envisagez-vous ?