Ruby on Rails verwenden

Während meiner Zeit bei ShowMeBug habe ich am Enterprise WeChat Integration Projekt mitgearbeitet. Dies umfasste die Integration von ShowMeBug mit Enterprise WeChat, wodurch ein nahtloser Zugriff auf technische Interview-Tools innerhalb des Enterprise WeChat-Ökosystems ermöglicht wurde. Ich nutzte Technologien wie Ruby, Ruby on Rails, PostgreSQL und die WeChat SDK, um sowohl für Interviewer als auch für Kandidaten ein reibungsloses Benutzererlebnis zu schaffen.

Dieser Blogbeitrag wurde mit Unterstützung von KI im Februar 2025 erstellt.

Ruby on Rails (oft einfach nur "Rails") ist ein leistungsstarkes Webentwicklung-Framework, das auf der Ruby-Programmiersprache basiert. Es ist darauf ausgelegt, das Erstellen von Webanwendungen schnell und angenehm zu gestalten, indem es Konventionen vor Konfiguration und DRY (Don't Repeat Yourself) Prinzipien betont. Lassen Sie uns durch die Einrichtung und das Erstellen einer einfachen App gehen.

Schritt 1: Ruby und Rails installieren Zunächst benötigen Sie eine Ruby-Installation, da Rails ein Ruby-Gem (Bibliothek) ist. Die meisten Systeme haben Ruby nicht vorinstalliert, daher hier die Einrichtung:

- **Auf macOS/Linux:**
  - Verwenden Sie einen Versionsmanager wie `rbenv` oder `rvm` für Flexibilität. Installieren Sie es über Homebrew (`brew install rbenv`), dann führen Sie aus:

    ```
    rbenv install 3.2.2  # Eine stabile Ruby-Version ab 2025
    rbenv global 3.2.2
    ```

  - Installieren Sie Rails:

    ```
    gem install rails
    ```

- **Auf Windows:**
  - Verwenden Sie RubyInstaller (herunterladen von rubyinstaller.org). Wählen Sie eine Version wie 3.2.2 mit DevKit.
  - Nach der Installation von Ruby öffnen Sie eine Eingabeaufforderung und führen aus:

    ```
    gem install rails
    ```

Überprüfen Sie die Installation:

```
ruby -v  # Sollte etwas wie ruby 3.2.2 anzeigen
rails -v # Sollte die neueste Rails-Version anzeigen, z.B. 7.1.x
```

Schritt 2: Ein neues Rails-Projekt erstellen Sobald Rails installiert ist, erstellen Sie eine neue App:

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

Dies erstellt einen Ordner namens `myapp` mit einer vollständigen Rails-Struktur, die SQLite als Standarddatenbank verwendet (ideal für die Entwicklung).

Schritt 3: Den Server starten Starten Sie den integrierten Rails-Server:

```
rails server
```

Öffnen Sie Ihren Browser unter `http://localhost:3000`. Sie sehen eine Willkommensseite. Herzlichen Glückwunsch, Ihre Rails-App läuft!

Schritt 4: Etwas Einfaches erstellen Lassen Sie uns ein grundlegendes "Posts"-Feature erstellen, um das MVC (Model-View-Controller) Muster von Rails zu verstehen.

- **Ein Modell und eine Steuerung generieren:**

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

  Dies erstellt ein `Post`-Modell, eine Datenbankmigration, eine Steuerung und Ansichten—alles miteinander verbunden.

- **Die Migration ausführen:**

  ```
  rails db:migrate
  ```

  Dies richtet die Datenbanktabelle für Beiträge ein.

- **Es ansehen:** Starten Sie den Server neu (`rails server`) und besuchen Sie `http://localhost:3000/posts`. Sie sehen eine CRUD-Schnittstelle zum Erstellen, Lesen, Aktualisieren und Löschen von Beiträgen.

Schritt 5: Wichtige Konzepte erkunden

- **Routen:** Öffnen Sie `config/routes.rb`. Sie sehen `resources :posts`, was RESTful-Routen wie `/posts/new` oder `/posts/1/edit` automatisch generiert.
- **Steuerungen:** Schauen Sie sich `app/controllers/posts_controller.rb` an. Es verarbeitet Anfragen und Antworten.
- **Ansichten:** Überprüfen Sie `app/views/posts/`. Dies sind ERB-Vorlagen (HTML mit eingebettetem Ruby).
- **Modelle:** Sehen Sie sich `app/models/post.rb` an. Es bindet an die Datenbank und kann Validierungen enthalten (z.B. `validates :title, presence: true`).

Schritt 6: Anpassen und Bereitstellen

- Fügen Sie etwas Stil mit CSS in `app/assets/stylesheets/` hinzu.
- Für die Produktion wechseln Sie zu einer Datenbank wie PostgreSQL (`rails new myapp --database=postgresql`) und stellen Sie auf eine Plattform wie Render oder Heroku bereit. Aktualisieren Sie `Gemfile` mit `gem "pg"` und führen Sie `bundle install` aus.

Pro-Tipps

- Verwenden Sie `rails console`, um mit Ihren Modellen in Echtzeit zu experimentieren.
- Führen Sie `rails generate --help` aus, um alle Abkürzungen zu sehen, die Rails bietet.
- Nutzen Sie Gems wie `devise` für die Authentifizierung oder `pundit` für die Autorisierung—fügen Sie sie Ihrem `Gemfile` hinzu und konfigurieren Sie sie nach Bedarf.

Das war's! Sie haben eine grundlegende Rails-App zum Laufen gebracht. Von hier aus erkunden Sie die offiziellen Rails Guides (guides.rubyonrails.org) oder erstellen Sie etwas Reales, um Ihre Fähigkeiten zu festigen. An welchem Projekt denken Sie?