

Broxigar's RoR Guide

Where Convention overthrows Configuration



GLAUBER B. @cancelei

WAR is not an EASY endeavor.

So Broxigar, after breaking the chains, decided to study Ruby on Rails and bring what you need to know for the next Battle.

Awesome!

Let's embark on Broxigar's journey in building an app using Ruby on Rails. We'll create a WoW-themed app that can be both educational and entertaining for new and intermediate Ruby on Rails developers. This will include features like character management, quest tracking, and item collections.

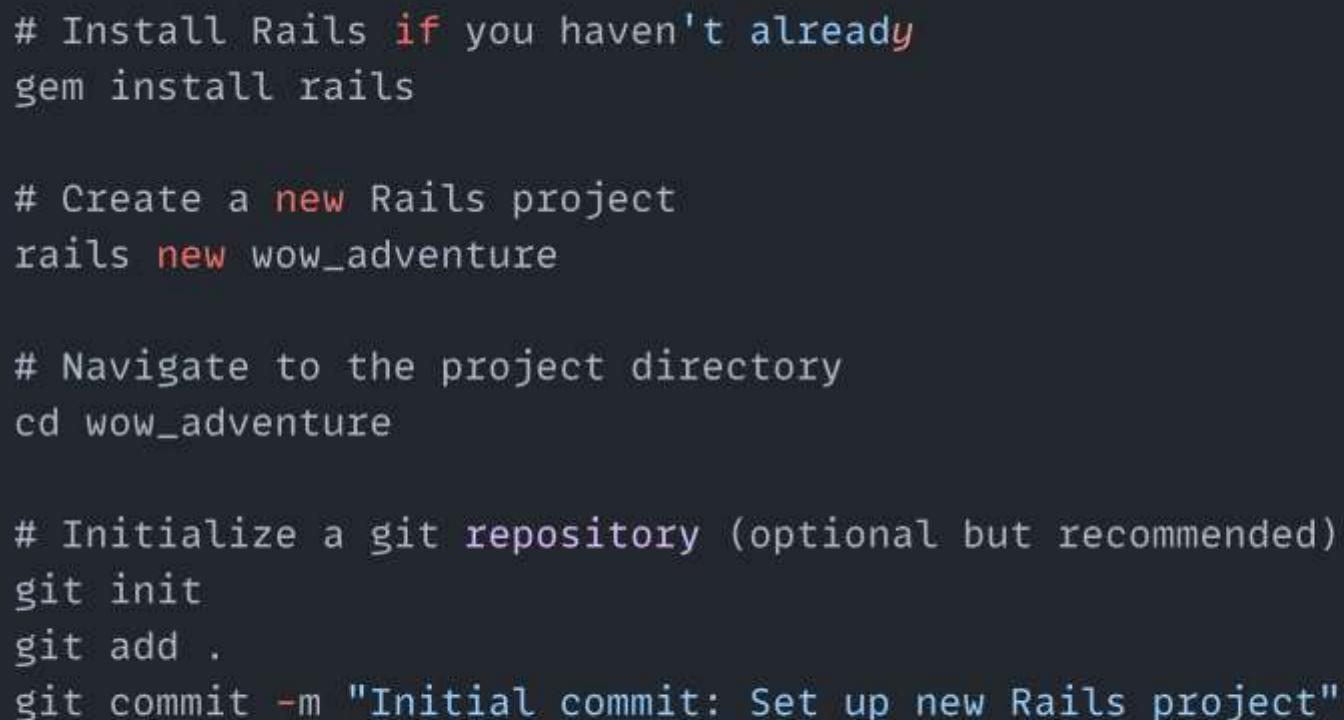
01

PROJECT SETUP & CREATING MODELS

Use the command line (CLI) to run commands before battle.

Step 1: Project Setup

We'll start by setting up a new Rails project. Open your terminal and run the following commands:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains a series of terminal commands for setting up a new Rails project, with some words highlighted in color to match the original image.

```
# Install Rails if you haven't already
gem install rails

# Create a new Rails project
rails new wow_adventure

# Navigate to the project directory
cd wow_adventure

# Initialize a git repository (optional but recommended)
git init
git add .
git commit -m "Initial commit: Set up new Rails project"
```

Step 2: Creating Models

We'll create models for Characters, Quests, and Items. These models will form the backbone of our app.

```
# Generate the Character model
rails generate model Character name:string race:string class:string level:integer

# Generate the Quest model
rails generate model Quest title:string description:text reward:string

# Generate the Item model
rails generate model Item name:string item_type:string rarity:string

# Run migrations to create the tables in the database
rails db:migrate
```

02

Setting Up Associations & Building Controllers

Building an army requires not just plans but build associations and having controllers ready to perform actions.

Setting Up Associations

Characters will have many quests and items. Let's set up these associations.

```
# app/models/character.rb
class Character < ApplicationRecord
  has_many :quests
  has_many :items
end

# app/models/quest.rb
class Quest < ApplicationRecord
  belongs_to :character
end

# app/models/item.rb
class Item < ApplicationRecord
  belongs_to :character
end
```

Next, we'll create controllers to manage our characters, quests, and items.

```
# Generate controllers for Characters, Quests, and Items
rails generate controller Characters
rails generate controller Quests
rails generate controller Items
```

Step 4: Building Controllers

Let us also create methods in each controller that appeared in Controllers folder.

We'll implement the basic CRUD operations in our controllers.

```
# app/controllers/characters_controller.rb
class CharactersController < ApplicationController
  def index
    @characters = Character.all
  end

  def show
    @character = Character.find(params[:id])
  end

  def new
    @character = Character.new
  end

  def create
    @character = Character.new(character_params)
    if @character.save
      redirect_to @character
    else
      render :new
    end
  end

  private

  def character_params
    params.require(:character).permit(:name, :race, :class, :level)
  end
end
```


03

Creating Views & Adding Routes

The views are the front end, like a war that also needs routes to swarm the adversaries.

Step 5: Adding Routes

Let's add routes for our resources.

```
# config/routes.rb
Rails.application.routes.draw do
  resources :characters do
    resources :quests
    resources :items
  end
  root 'characters#index'
end
```

We'll create basic views for listing, creating, and showing characters, quests, and items.

```
←!— app/views/characters/index.html.erb —→
<h1>Characters</h1>
<%= link_to 'New Character', new_character_path %>
<ul>
  <% @characters.each do |character| %>
    <li>
      <%= link_to character.name, character_path(character) %>
      (Level: <%= character.level %>, Race: <%= character.race %>, Class: <%= character.class %>)
    </li>
  <% end %>
</ul>
```

Step 6: Creating Views

We'll create basic views for listing, creating, and showing characters, quests, and items.

```
←!— app/views/characters/show.html.erb →  
<h1><%= @character.name %></h1>  
<p>Level: <%= @character.level %></p>  
<p>Race: <%= @character.race %></p>  
<p>Class: <%= @character.class %></p>
```

```
←!— app/views/characters/new.html.erb →  
<h1>New Character</h1>  
<%= form_with model: @character, local: true do |form| %>  
  <div>  
    <%= form.label :name %>  
    <%= form.text_field :name %>  
  </div>  
  <div>  
    <%= form.label :race %>  
    <%= form.text_field :race %>  
  </div>  
  <div>  
    <%= form.label :class %>  
    <%= form.text_field :class %>  
  </div>  
  <div>  
    <%= form.label :level %>  
    <%= form.number_field :level %>  
  </div>  
  <div>  
    <%= form.submit %>  
  </div>  
<% end %>  
<%= link_to 'Back', characters_path %>
```

**Good luck on your
programming journey!**

Hope this Presentation inspired you to sharpen up your weapons.



<https://github.com/cancelei>