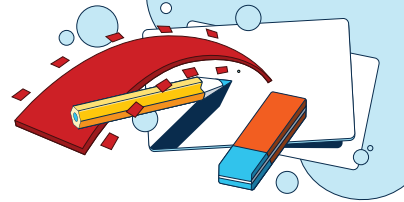


RUBY ON RAILS

presented by Tower - the best Git client for Mac and Windows



Rails Commands

gem install rails

installs the Rails gem.

rails new app_name --database=postgres

creates a new Rails application with the specified name and Postgres support.

bundle install

installs all gems specified in the [Gemfile](#).

rails console

opens a console to interact with the Rails environment.

rails server

starts the Rails development server.

gem 'gem_name', '~> version_number'

specifies a gem and its version in the Gemfile.

rails routes

displays all the routes in an application.

rails dev:cache

toggles Rails caching.

Rails Databases

rails db:create

creates a new database specified in the [config/database.yml](#) file.

rails db:seed

runs the seed data specified in the [db/seeds.rb](#) file.

rails db:migrate

runs any pending database migrations.

rails db:rollback

rolls back a database migration one step at a time.

rails db:drop

drops the database specified in the [config/database.yml](#) file.

rails generate migration AddColumnToTable column_name:datatype

adds a new column to an existing database table.

rails generate migration RemoveColumnFromTable column_name:datatype

removes a column from an existing database table.

rails generate migration RenameColumnInTable old_column_name:new_column_name

renames a column in an existing database table.

rails generate migration CreateJoinTable table1 table2

creates a new join table to associate two existing tables.

Rails Routing (config/routes.rb)

resources :models

generates RESTful routes for the specified model.

get 'path(/:optional_parameter)'

defines a route with an optional parameter.

resources :models, except: [:edit, :update]

generates RESTful routes for a model, excluding certain actions.

root 'controller#action'

sets the root path to the specified controller and action.

get 'path' => 'controller#action'

maps a GET request to the specified path to the specified controller and action.

post 'path' => 'controller#action'

maps a POST request to the specified path to the specified controller and action.

Rails Models

rails generate model ModelName attribute1:type attribute2:type

generates a new model with the specified attributes and their data types.

validates :attribute, presence: true

ensures that the specified attribute is not blank.model.

validates :attribute, uniqueness: true

ensures that the specified attribute is unique.

validates :attribute, length: { minimum: 2, maximum: 20 }

ensures that the specified attribute is within a certain length range.

belongs_to :model

sets up a one-to-one association with the specified model.

has_many :models

sets up a one-to-many association with the specified model.

has_and_belongs_to_many :models

sets up a many-to-many association with the specified model.

before_save :method_name

runs a specified method before saving a model instance to the database.

Rails Views

<% ruby_code %>

executes Ruby code in a view.

<%= ruby_code %>

embeds Ruby code in a view.

<% if condition %>...<% else %>...<% end %>

executes Ruby code conditionally in a view.

<%= render 'partial_name', variable: value %>

renders a specified partial view with a local variable.

<%= image_tag "image.png" %>

links to image asset specified in the [app/assets/images](#) folder.

<%= stylesheet_link_tag "application" %>

links to CSS asset.

<%= javascript_include_tag "application" %>

links to JavaScript asset.

<%= link_to 'Link Text', path, method: :delete %>

generates a link to delete a resource with the specified text.

Rails Controllers

rails generate controller ControllerName index show2

generates a new controller with the specified actions, routes and views.

def action

defines a new controller action.

redirect_to path

redirects the user to the specified path.

render :action

renders the specified view template.

params[:parameter]

accesses a parameter passed from a form or URL query string.

render partial: 'partial_name', locals: { variable: value }

renders a specified partial view with local variables.

render json: @model_instance

renders a model instance as JSON format.

flash[:notice] = 'Message'

stores a message in the flash hash to be displayed to the user after a redirect.

redirect_to :back

redirects the user to the previous page.

respond_to do |format|

specifies how to respond to different formats (e.g., HTML, JSON) in a controller action.