Ruby on Rails

Ruby web framework

CONCEPTS

Inside an app use bin/rails: it runs the version pinned in the Gemfile, not a global one.

generate files + migration
↓ edit
migration db/migrate
↓ db:migrate
schema db/schema.rb
↓ routes + test
server localhost:3000

Short aliases: s server, c console, g generate, d destroy, db dbconsole, t test.

CREATE & RUN

`rails new <app>`
Create a new app

`rails new <app> --database=postgresql`
New app using PostgreSQL

`bin/rails server`
Start the server (Puma)

`bin/rails server -p <port>`
Use another port

`bin/rails about`
Rails, Ruby and DB versions

GENERATORS

`bin/rails generate`
List available generators

`bin/rails generate scaffold <Name> <field:type>`
Model, controller, views, routes

`bin/rails generate model <Name> <field:type>`
Model and migration

`bin/rails generate controller <Name> <action>`
Controller with views and routes

`bin/rails generate migration <Name>`
Empty migration

GENERATORS (cont.)

`bin/rails destroy <generator> <args>`
Remove what generate made

DATABASE

`bin/rails db:migrate`
Run pending migrations

`bin/rails db:rollback`
Undo the last migration

`bin/rails db:migrate:status`
Show each migration up or down

`bin/rails db:create`
Create the database

`bin/rails db:seed`
Load db/seeds.rb

`bin/rails db:prepare`
Set up, or migrate if it exists

`bin/rails dbconsole`
Open psql, mysql or sqlite3

CONSOLE & RUNNER

`bin/rails console`
Interactive shell with the app loaded

`bin/rails console --sandbox`
Roll back all changes on exit

`bin/rails runner "<code>"`
Run Ruby in the app context

INSPECT

`bin/rails routes`
List all routes

`bin/rails routes -g <pattern>`
Filter by name, path or action

TEST

`bin/rails test`
Run tests, not system tests

`bin/rails test <file>:<line>`
Run the test at a line

HOUSEKEEPING

`bin/rails credentials:edit`
Edit encrypted credentials