

Q1. What does 'Convention over Configuration' mean in Rails?

A: Rails derives behavior from naming conventions. A User model maps to the users table. UserController handles /users routes. This means less config: no mapping files are needed. Follow the convention and Rails wires everything automatically; deviate and you configure explicitly.

Q2. What is the Rails request lifecycle?

A: A request hits the Router (config/routes.rb) which dispatches to a controller action. The action interacts with models, then renders a view or returns JSON. Rack middleware (CSRF, session, logger) wraps the entire stack and runs before/after the router.

Q3. What is Active Record and how do associations work?

A: Active Record implements the active record pattern: model instances map to rows and class methods generate SQL. has_many :orders in User and belongs_to :user in Order define the association. user.orders loads associated rows. user.orders.create(total:100) inserts and associates.

Q4. How do Rails migrations work and how do you roll one back?

A: rails generate migration AddAgeToUsers age:integer creates a timestamped migration file. rails db:migrate runs all pending Up methods. rails db:rollback runs the most recent Down method, reversing the change. db:migrate VERSION=20230101 migrates to a specific version.

Q5. How does Rails routing work with resources?

A: resources :users in routes.rb generates seven RESTful routes (index, show, new, create, edit, update, destroy). namespace :api {resources :users} prefixes routes and expects Api::UsersController. match '/about', via: :get defines a custom route.

Q6. What are Rails concerns and what problems do they solve?

A: A concern is a module in app/models/concerns or app/controllers/concerns using ActiveSupport::Concern. Include them in models/controllers to share reusable behavior (Taggable, Auditable) without multiple inheritance. included do ... end runs in the including class's context.

Q7. How does Active Job work with Sidekiq?

A: Define a job class extending ApplicationJob with a perform(*args) method. Set queue_adapter :sidekiq in ApplicationJob. Enqueue with MyJob.perform_later(user_id). Sidekiq workers pick up jobs from Redis, call perform, and handle retries on failure with exponential backoff.

Q8. What is Action Cable in Rails?

A: Action Cable integrates WebSockets into Rails using channels. A channel class (ApplicationCable::Channel) defines subscribed and receive methods. The client subscribes via App.cable.subscriptions.create('ChatChannel', callbacks). Broadcast from

Q9. How does Rails asset management work with Propshaft?

A: Propshaft (Rails 7+) is a simpler asset pipeline. It fingerprints files for cache busting and serves them from /assets. Unlike Sprockets, it does not transpile or concatenate use importmap-rails for ES modules or a bundler (esbuild, Vite) separately for complex JS.

Q10. How do you test a Rails model with RSpec and FactoryBot?

A: FactoryBot.define { factory :user { name 'Alice'; email 'alice@ex.com' } }. In the spec: let(:user) { create(:user) }. Test associations, validations, and methods with expect(user.email).to eq('alice@ex.com') and expect(user).to be_valid. Run with bundle exec rspec.