

## Advanced Techniques in Ruby on Rails (code: Rails Advanced)

### Overview

Ruby programming language was created by Yukihiro Matsumoto (Matz) and firstly published in 1995. It's a very modern, interpreted programming language available on many platforms. It's pure, object-oriented language which adopted many of its properties from the Smalltalk language. Ruby on Rails is advanced environment (framework) used for creating modern and advanced services / web applications (websites) in a quick, non-stressful way.

During the course we will introduce advanced programming techniques of Ruby language, that is to say metaprogramming (technique which allows us to generate application code automatically in real time), hook methods (which allow application to affect the current code execution of Ruby interpreter) and other solutions to create very elastic and elegant code. We will discuss briefly some good practices of object-oriented programming in Ruby language, introducing also examples of how these techniques can be used in Ruby environment itself.

Participants will see many advanced programming mechanisms using Ruby on Rails environment. We will lay particular stress on TDD techniques and creating application tests, way of optimizing application performance through advanced cache usage on different levels of processing data, creating interactive applications and exchanging information in real time. Also, we will present closely the methods of putting tasks to the background and we will discuss asset pipeline, a way of managing Javascript styles and scripts inside the application, which was introduced in Rails 3.1.

### Duration

3 days

### Agenda

1. Advanced programming in Ruby
  - Object-oriented programming
    - Using modules and mixins
    - Advanced use cases for closures – blocks, procs and lambdas
  - Metaprogramming with Ruby
    - Creating methods with `define_method`
    - Interpreting code through `eval`, `instance_eval` and `class_eval`
  - Hook methods
    - Tracking callbacks (`method_added`, `inherited`, `included`, `extended` etc.)
    - Using `method_missing` and `const_missing`
2. Advanced techniques in Ruby on Rails
  - Good practices while designing models relations and data logic
  - Advanced associations: `polymorphic`, `has_and_belongs_to_many` etc.
  - Standard data serialization and hstore (PostgreSQL)
  - Asset pipeline and complicated views
  - Background tasks
    - Selecting tasks appropriate for background processing
    - When to use full environment for supporting tasks, alternative methods
    - Review of available libraries and techniques (`resque`, `sidekiq`, `beanstalkd`, etc)
  - Best practices while creating REST API

### Ask for details

Phone +44 203 608 6289

info@alx.training

- Profiling applications, searching for bottlenecks
  - MiniProfiler
  - Bullet
  - NewRelic
- Creating Gems and Engines
- 3. Effective code testing and TDD practices
  - Best practices
  - Refactoring
  - Guard, RSpec, MiniTest, Cucumber, FactoryGirl
- 4. Exchanging information by application in real time
  - Review of available techniques: long pulling, websockets, streaming
  - ActionController: Live (Rails 4+)
  - Faye messaging server
  - Cache in Rails applications – advanced techniques
    - Page caching
    - Actions and fragments caching
    - Caching data from any application component
    - Expiring cache (Sweepers, Observers)
    - Auto-expiring cache keys

**Ask for details**

Phone +44 203 608 6289

info@alx.training

**Target audience and prerequisites**

We assume that the participant knows Ruby language fairly well and has at least basic knowledge on Ruby on Rails environment. One should be able to create a simple application in Ruby on Rails environment. It's also necessary to be familiar with technologies related to creating websites. At least basic knowledge of CSS, Javascript language and JQuery library are required.

For people having no experience in Rails environment we would recommend other training: Web Applications with Ruby on Rails, as it introduces that framework from the very ground up.

Below you will find the summary of requirements/recommendations for the initial knowledge and skills of training participants:

- solid knowledge of basics concerning programming in Ruby language
- general knowledge of programming in Ruby on Rails environment (at least 3+ version recommended)
- comfortable with working in terminal environment (bash) of Linux/BSD/OSX system
- understanding of HTTP protocol and related topics
- good knowledge of HTML and CSS
- basic knowledge of Javascript language (and basics of JQuery)
- knowledge of SQL and basic knowledge of MySQL or Postgres database

**Certificates**

Course participants receive completion certificates signed by ALX. Upon completing the course, participants will have the following skills/ knowledge:

- creating sophisticated and advanced WWW applications
- advanced programming techniques in Ruby language
- creating advanced application logic through ActiveRecord relations
- advanced techniques of optimizing application effectiveness
- ability to search for problems and bottlenecks
- knowledge of TDD techniques and ability to create effective test for applications

- ability to create one's own gems
- ability to connect different applications using Engines
- understanding processing background tasks
- ability to effectively organise styles and Javascript code inside Rails application

### **Ask for details**

Phone +44 203 608 6289  
info@alx.training

## **Locations**

- Warsaw (English) – Jasna 14/16A
- Online (English) – your home, office or wherever you want
- any other location (London, UK, EU) on request

## **Price**

1290 EUR

The price includes:

- course materials,
- snacks, coffee, tea and soft drinks,
- course completion certificate,
- one-time consultation with the instructor after course completion.