

# Ruby on Rails Job Interview Questions And Answers



**Interview Questions Answers**

**<https://interviewquestionsanswers.org/>**

## About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Ruby on Rails will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Ruby on Rails Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Ruby on Rails category. To ensure quality, each submission is checked by our team, before it becomes live. This [Ruby on Rails Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.  
[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter for latest Jobs and interview preparation guides.  
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**  
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



## Ruby on Rails Interview Questions And Answers Guide.

### Question - 1:

Why Ruby on Rails?

#### Ans:

Super productive new way to develop web applications because:

1. Dynamically typed programming language similar to Python, Smalltalk, and Perl(Templating language)
2. Follows Model-View-Controller (MVC) architecture
3. Strives for simplicity
4. Less coding.
5. Minimum Configuration.
6. Design patterns admire
7. Light webserver
8. ORM

[View All Answers](#)

### Question - 2:

What is session and cookies?

#### Ans:

Session is used for maintaining the particular value throughout the session..(browser closed)

e.x: session[:value]="poornimad"

this value is not changed until the browser closed..

Cookies:

\*\*\*\*\*

This is used to store the values in the Browser..this value is not delete till the cookies are deleted..This cookies concept is same as Java cookies.The concept is same.

[View All Answers](#)

### Question - 3:

Whats the difference between symbol and string?

#### Ans:

Rails makes extensive use of symbols. A symbol looks like a variable symbols name, but it's prefixed with a colon.

Examples of symbols include :action,:line\_items, and :id.

Rails uses symbols to identify things. In particular, it uses them as keys when naming method parameters and looking things up in hashes. For example:

```
redirect_to :action => "edit", :id => params[:id]
```

[View All Answers](#)

### Question - 4:

What the difference between static scaffolding and Dynamic scaffolding?

#### Ans:

With Rails 2.0, you may have noticed that dynamic scaffolding breaks-that is, if you have a controller with scaffold :model\_name in it, all the scaffolded actions-new, delete, index-no longer exist! In Rails 2.0, you can only generate static scaffolding-that is, you can use the scaffold to generate the files for controllers, models, and views.

What's more, Rails 2.0 allows you to specify the model



attributes inside the scaffold. This then creates views with all the appropriate fields, and it also creates the migration with all the fields in it! Excellent!

As an example, say we wanted to create a blog-post model. We could generate it like so:

```
script/generate scaffold Post title:string content:text
category_id:integer
```

You'll notice Rails will generate, among other things:

- \* A post.rb model file
- \* A posts\_controller.rb controller file
- \* A posts view folder containing views for the index, show, new, and edit actions
- \* A DB migration called xxx\_create\_posts
- \* A unit-test, fixtures file, and helper

Everything you need-indeed, everything the dynamic scaffolding provided-is included, albeit as static content.

All you need to do is migrate your DB and you're up and flying!

So the main difference is, with dynamic scaffolding you can generate new, edit and delete methods but with static scaffolding you can't

[View All Answers](#)

### Question - 5:

Why do we use request.xhr? in rails?

#### Ans:

Conventional web application transmit information to and from the sever using synchronous requests. This means you fill out a form, hit submit, and get directed to a new page with new information from the server.

When you interact with an Ajax-powered web page, it loads an Ajax engine in the background. In response to an event web application passes asynchronous request (XMLHttpRequest or xhr). In fact it is javaScript object/method that performs asynchronous interaction with the server, JavaScript object that performs asynchronous interaction with the server and behind the scene fetches data. and behind the scene fetches data.

We do request.xhr? only to check the request type, either its AJAX request or others(post, get).

[View All Answers](#)

### Question - 6:

What is difference between form\_for and form\_tag?

#### Ans:

form\_for and form\_tag both are used to submit the form in ruby on rails.

but the way of handling objects related to model is different.

form\_for:

you should use form\_for for a specific model i.e while crating an new row in database. form\_for will perform the standard http post which is having fields related to active record objects.

here is the example for using form\_for in ruby on rails:

```
<% form_for :user, @user, :url => { :action => "update" }
do |f| %>
```

then in here you can use the f object to create input field.

```
First name: <%= f.text_field :firstname %>
```

```
Last name : <%= f.text_field :lastname %>
```

```
Biography : <%= f.text_area :biography %>
```

```
<% end %>
```

form\_tag:

form\_tag just creates an form as an normal form. form\_for will perform the standard http post without any model backed and has normal fields. this is used mainly when specific things need to be submitted via form

here is the example for using form\_tag in ruby on rails:

```
<% form_tag '/posts' do -%>
```

```
<%= text_field_tag "post", "firstname" %>
```

```
<% end -%>
```

[View All Answers](#)

### Question - 7:

How will be the future for ruby on rails?

#### Ans:

Ruby On Rails developed in 1992.But now it is spreading all



over. Coding using ROR is easy we can easily develop web applications. We can develop webservers and Ajax also can be implemented in ROR. Similarly we have some disadvantages in it like the User Interface and easily can do the wrong things. Finally It is easy to learn and easy to implement also. Learn it, It has good future.

[View All Answers](#)

### Question - 8:

What is Ruby software and where and when it is useful?  
Is it useful to a database developer, if yes...where?

#### Ans:

Ruby is a dynamic object-oriented programming language. We can use it for any application like Java, or we can use it for web programming language.

[View All Answers](#)

### Question - 9:

What are filters? and how many types of filters are there in Ruby?

#### Ans:

Filters enable controllers to run shared pre and post processing code for its actions.  
Filter methods are macro-style, that is, they appear at the top of your controller method, inside the class context, before method definitions.  
The below types of filters in Ruby:  
before\_filter,  
after\_filter,  
prepend\_before\_filter,  
prepend\_after\_filter,  
around\_filter

[View All Answers](#)

### Question - 10:

What is the difference between sessions and flash?

#### Ans:

Session is only for storing the small amount of user data, which data is not stored permanently that's when we logged out from our application.. which session value will be removed.  
But Flash is used to display some messages like error message and useful information messages in view pages...

[View All Answers](#)

### Question - 11:

What is the functionality of Model, views and controllers?

#### Ans:

Model: Model is responsible for maintaining the database.  
views: View is responsible for generating user interface.  
controllers: Getting input request from browser and generate outgoing response.

[View All Answers](#)

### Question - 12:

How are Model, views and controllers related?

#### Ans:

Model, View and Controller (MVC) is a 3-tier architecture.  
View consists of all HTML, JavaScript pages. Its client-side template. While as Model connects to the database describing all data fields and the relation between them and all business calculations are controlled by controller.

[View All Answers](#)

### Question - 13:

Is Ruby a scripting language or a compiled language?

#### Ans:

In general, programming languages fall into one of two categories: they're either compiled languages or scripting languages. Let's explore what each of those terms means, and understand the differences between them.



**Compiled Languages:** The language in which you write an application is not actually something that your computer understands. Your code needs to be translated into bits and bytes that can be executed by your computer. This process of translation is called compilation, and any language that requires compilation is referred to as a compiled language. Examples of compiled languages include C, C#, and Java. For a compiled language, the actual compilation is the final step in the development process. You invoke a compiler -- the software program that translates your final hand-written, human-readable code into machine-readable code -- and the compiler creates an executable file. This final product is then able to execute independently of the original source code.

Thus, if you make changes to your code, and you want those changes to be incorporated into the application, you must stop the running application, recompile it, then start the application again.

**Scripting Languages:** On the other hand, a scripting language such as Ruby, PHP, or Python, relies upon an application's source code all of the time. Scripting languages don't have a compiler or a compilation phase per se; instead, they use an interpreter -- a program that runs on the web server -- to translate hand-written code into machine-executable code on the fly. The link between the running application and your hand-crafted code is never severed, because that scripting code is translated every time it is invoked -- in other words, for every web page that your application renders.

As you might have gathered from the name, the use of an interpreter rather than a compiler is the major difference between a scripting language and a compiled language. **The Great Performance Debate:** If you've come from a compiled-language background, you might be concerned by all this talk of translating code on the fly -- how does it affect the application's performance?

These concerns are valid -- translating code on the web server every time it's needed is certainly more expensive, performance-wise, than executing pre-compiled code, as it requires more effort on the part of your machine's processor. The good news is that there are ways to speed up scripted languages, including techniques such as code caching and persistent interpreters. However, both topics are beyond the scope of this book.

There's also an upside to scripted languages in terms of performance -- namely, your performance while developing an application.

Imagine that you've just compiled a shiny new Java application, and launched it for the first time ... and then you notice a typo on the welcome screen. To fix it, you have to stop your application, go back to the source code, fix the typo, wait for the code to recompile, and restart your application to confirm that it is fixed. And if you find another typo, you'll need to repeat that process again.

Lather, rinse, repeat.

In a scripting language, you can fix the typo and just reload the page in your browser -- no restart, no recompile, no nothing. It's as simple as that.

[View All Answers](#)

### Question - 14:

What i have to do to view my project always on browser?

**Ans:**

if u run the server (ruby script/server -d) then the command line will close ,but server is running.The server will run until the machine is restart. you can start different port using ruby script/server -p 3001 -d

[View All Answers](#)

### Question - 15:

Is it possible to build a 50% bespoke e-commerce platform hence having the ability to customize everything down the line? For example would it make sense to start coding an application on the Ruby on Rails framework but where the most complex/time consuming code pieces (e.g. shopping cart, etc) can be initially bolted on (hence diminishing development time and cost) having the ability to change them completely further down the line?

**Ans:**

Yes 100% we can build application for e-commerce platform for example shopify and we have easily integrate the payment transactions like paypal and authorize.net



[View All Answers](#)

### Question - 16:

What are the differences between Rails 2.x and Rails 3?

#### Ans:

- (1) Introduction of bundler (New way to manage your gem dependencies)
- \* (2) Gemfile and Gemfile.lock (Where all your gem dependencies lies, instead of environment.rb)
- \* (3) A new .rb file in config/ folder, named as application.rb (Which has everything that previously environment.rb had)
- \* (4) Change in SQL Structure: Model.where(:activated => true)
- \* (5) All the mailer script will now be in app/mailers folder, earlier we kept inside app/models.
- \* (6) Rails3-UJS support. for links and forms to work as AJAX, instead of writing complex lines of code, we write :remote => true
- \* (7) HTML 5 support.
- \* (8) Changes in the model based validation syntax: validates :name, :presence => true
- \* (9) Ability to install windows/ruby/jruby/development/production specific gems to Gemfile.  
group :production do  
gem 'will\_paginate'  
end

[View All Answers](#)

# Computer Programming Most Popular & Related Interview Guides

- 1 : [Python Interview Questions and Answers.](#)
- 2 : [OOP Interview Questions and Answers.](#)
- 3 : [Software engineering Interview Questions and Answers.](#)
- 4 : [PHP Interview Questions and Answers.](#)
- 5 : [VBA \(Visual Basic for Applications\) Interview Questions and Answers.](#)
- 6 : [Visual Basic \(VB\) Interview Questions and Answers.](#)
- 7 : [Node.js Interview Questions and Answers.](#)
- 8 : [CMMI Interview Questions and Answers.](#)
- 9 : [Microsoft Foundation Class \(MFC\) Interview Questions and Answers.](#)
- 10 : [Lotus Notes Interview Questions and Answers.](#)

Follow us on FaceBook

[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/  
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)