

105 Ruby on Rails interview questions to hire top developers

Questions

1. What is Rails? Explain it like I'm five.
2. What does MVC stand for in Rails, and why is it important?
3. Can you describe the purpose of a Rails migration?
4. What is a Rails model, and what does it do?
5. Explain the purpose of a Rails controller.
6. What is a Rails view, and how does it relate to the controller?
7. What's the difference between 'rails new' and 'rails generate'?
8. What is the role of the 'routes.rb' file in a Rails application?
9. How do you create a new controller in Rails using the command line?
10. What is ActiveRecord in Rails?
11. How do you define a model with attributes in Rails?
12. How can you perform CRUD (Create, Read, Update, Delete) operations in Rails?
13. What is a Rails form, and how do you create one?
14. Explain the purpose of Rails helpers.
15. What is an asset pipeline in Rails? What problem does it solve?
16. How do you add CSS and JavaScript to your Rails application?
17. What is a Rails partial, and why would you use one?
18. Explain what Rails environments are (development, test, production).
19. How do you start a Rails server?
20. What is the purpose of the 'Gemfile' in a Rails project?
21. How do you install gems in a Rails project?
22. What is a 'rake' task in Rails, and give an example?
23. How do you run database migrations in Rails?
24. Describe the difference between 'GET' and 'POST' HTTP methods.
25. What are Rails sessions and cookies used for?
26. Explain the concept of RESTful routes in Rails.
27. What are the common data types in Rails migrations?
28. How do you debug a Rails application?
29. What is Rails? Explain it like you're talking to a friend who knows nothing about web development.
30. Can you describe the MVC pattern and how it's used in Rails? Imagine you're explaining it to a kid with building blocks.
31. What is a Rails migration? Why would you use one?
32. What's the difference between 'rails new' and 'rails generate'?
33. What is ActiveRecord? What does it help us do?
34. How would you define a model in Rails? Can you give a simple example?
35. What are Rails routes? Can you give an example of a simple route?
36. What is a Rails controller? What is its main job?
37. What are views in Rails? What's their purpose?
38. What is a 'gem' in Ruby on Rails? Give an example of a gem and what it does.
39. How do you start a Rails server?
40. What is the purpose of the 'rails console'?
41. Explain the difference between 'GET' and 'POST' requests.
42. What is a 'form' in HTML, and how does it relate to a Rails application?
43. What is a database? What is a common database used with Rails?
44. What does CRUD stand for? Give examples of how CRUD actions relate to a 'User' model.
45. What is an 'association' in Rails? Give an example like 'has_many'.
46. How would you create a new database in Rails?
47. What is the asset pipeline in Rails? What is its purpose?
48. What is a partial in Rails views? Why would you use one?
49. How would you display a variable from a controller in a Rails view?
50. What is the purpose of a 'flash' message in Rails?
51. What are some common data types you might use in a Rails model (e.g., string, integer)?
52. What is a 'rake' task? Can you describe a common rake task?
53. What is testing in the context of Ruby on Rails? Why is it important?
54. How does Rails handle database migrations, and what's the process for rolling back a migration?
55. Explain the concept of concerns in Rails. When and why would you use them?
56. Describe how you would implement a custom validator in Rails. Provide an example.
57. What are the different types of Rails associations (e.g., has_many, belongs_to), and how do you choose the right one?
58. Explain how you would optimize a slow-performing Rails application. What tools and techniques would you use?
59. What are the benefits of using ActiveJob in Rails, and how does it work with different queueing backends?
60. Describe your understanding of polymorphic associations in Rails. Give a practical use case.
61. How can you implement authentication and authorization in a Rails application? What gems or techniques would you consider?
62. Explain the difference between render and redirect_to in a Rails controller.
63. How do you handle form submissions with nested attributes in Rails?
64. Describe how you would implement API versioning in a Rails application.
65. What is the purpose of Rack in the context of a Rails application?
66. Explain how you would implement caching in a Rails application. What different caching strategies are available?
67. How would you handle background tasks in a Rails application? What are some options, and what are the trade-offs?
68. Describe how you would debug a Rails application. What tools and techniques do you find most helpful?
69. What are the benefits of using Service Objects in Rails applications? Explain with example.
70. Explain the use of scopes in Rails models. How do they improve code readability and maintainability?
71. How do you handle file uploads in Rails? What gems or techniques would you use for image processing?
72. What are the different ways to handle internationalization (i18n) in a Rails application?
73. Describe the process of deploying a Rails application to a production environment.
74. Explain how you would implement a search functionality in a Rails application. What are some performance considerations?
75. How would you implement real-time features (e.g., chat, notifications) in a Rails application?
76. Describe the use of ActiveModel::Serializers in Rails. When should you use it?
77. How do you handle different environments (e.g., development, test, production) in a Rails application?
78. What is the purpose of using gems like Pundit or CanCanCan in Rails? How do they help with authorization?
79. Explain the concept of eager loading in Rails. Why is it important and how does it work?
80. How would you optimize a slow Rails application, focusing on database query optimization and caching strategies?
81. Describe your experience with different Rails testing frameworks (e.g., RSpec, Minitest) and your preferred testing strategies.
82. Explain how you would implement a complex search functionality with filtering, sorting, and pagination in a Rails application.
83. How do you handle background jobs in Rails, and what are the pros and cons of different background processing libraries (e.g., Sidekiq, Resque)?
84. Describe a time you had to debug a complex issue in a Rails application. What steps did you take to identify and resolve the problem?
85. Explain your approach to securing a Rails application against common web vulnerabilities like CSRF, XSS, and SQL injection.
86. How would you implement a multi-tenancy architecture in a Rails application, and what are the considerations for data isolation and scalability?
87. Describe your experience with different deployment strategies for Rails applications (e.g., Capistrano, Docker, Kubernetes).
88. Explain how you would implement a RESTful API in Rails, including versioning, authentication, and documentation.
89. How do you approach code reviews in a Rails project, and what are the key things you look for in a code review?
90. Describe your experience with different Rails caching techniques (e.g., fragment caching, Russian doll caching) and when you would use each one.
91. Explain how you would handle internationalization (i18n) and localization (l10n) in a Rails application.
92. How do you stay up-to-date with the latest trends and best practices in the Rails ecosystem?
93. Describe a challenging Rails project you worked on and the lessons you learned from it.
94. Explain how you would implement a real-time feature in a Rails application (e.g., using Action Cable or WebSockets).
95. How do you handle database migrations in a Rails project, and what are some best practices for writing and running migrations?
96. Describe your experience with different Rails ORM's (e.g., ActiveRecord, Sequel) and your preferred approach to interacting with databases.
97. Explain how you would implement a role-based access control (RBAC) system in a Rails application.
98. How do you handle performance monitoring and alerting in a Rails application, and what tools do you use for this purpose?
99. Describe your experience with different Rails engines and when you would use an engine in a Rails project.
100. Explain how you would implement a payment gateway integration in a Rails application.
101. How do you handle file uploads and storage in a Rails application, and what are some considerations for security and scalability?
102. Describe your experience with different Rails code analysis tools (e.g., RuboCop, Brakeman) and how they help improve code quality.
103. Explain how you would implement a search engine optimization (SEO) strategy in a Rails application.