

# 97 Symfony Interview Questions to Hire Top Engineers

## Questions

---

1. What is Symfony, in simple terms?
2. Can you explain what a framework is, like I'm five?
3. What are some advantages of using Symfony over writing code from scratch?
4. What is a bundle in Symfony, and can you give an example?
5. What's the difference between the `src/` and `vendor/` directories?
6. What is routing in Symfony? Explain its purpose.
7. How do you create a simple route in Symfony?
8. What is a controller in Symfony, and what does it do?
9. How do you pass data from a controller to a template?
10. What are templates in Symfony used for?
11. What is Twig, and how is it used in Symfony?
12. How would you display a variable in a Twig template?
13. What is a service in Symfony, and why are they useful?
14. Can you name a few commonly used services in Symfony?
15. What is dependency injection? (Explain like I'm five)
16. How does Symfony handle forms?
17. What is a form builder in Symfony?
18. What is the purpose of CSRF protection in forms?
19. How do you validate form data in Symfony?
20. What is Doctrine, and how is it used with Symfony?
21. How do you define a database entity in Symfony using Doctrine?
22. What is a migration in Doctrine, and why is it important?
23. How can you perform database queries using Doctrine in Symfony?
24. What is the Symfony command-line tool (bin/console) used for?
25. Can you give some examples of useful Symfony console commands?
26. What is the purpose of environment variables in Symfony?
27. How would you define different configurations for development and production environments?
28. What is the Symfony profiler, and how can it help with debugging?
29. Explain the basic structure of a Symfony project directory.
30. If you encountered an error in Symfony, how would you typically go about debugging it?
31. What's a Symfony bundle, like a LEGO brick for your app?
32. Can you name a few Symfony components you've used?
33. How does Symfony help keep your code organized?
34. What's the difference between `app_dev.php` and `app.php`?
35. Explain what a service container is in Symfony.
36. What is the purpose of a YAML file in Symfony configuration?
37. What's the role of the `composer.json` file in a Symfony project?
38. How would you define a route in Symfony, and what does it do?
39. What are controllers in Symfony, and what do they handle?
40. What's the role of templates in a Symfony application?
41. How do you pass data from a controller to a template?
42. What is a form in Symfony, and why do we use them?
43. How would you handle form validation in Symfony?
44. What's the Doctrine ORM, and how does Symfony use it?
45. How do you connect to a database in Symfony?
46. What's the purpose of migrations in Symfony, and how do you run one?
47. Explain how you would debug a simple error in a Symfony application.
48. What are environment variables, and why are they useful in Symfony?
49. How would you clear the Symfony cache?
50. What's the difference between `dump()` and `dd()` in Symfony?
51. Describe a simple task you've done with Symfony and what you learned.
52. Explain the lifecycle of a Symfony request. What happens from the moment a user sends a request to when they receive a response?
53. How can you implement custom validation logic in Symfony, beyond the built-in constraints?
54. Describe the role of the Symfony event dispatcher and how you would create and dispatch a custom event.
55. Explain the difference between Symfony services, parameters, and configurations. When would you use each?
56. How do you handle file uploads in Symfony, including validation and storage?
57. What are the advantages and disadvantages of using Doctrine ORM in a Symfony project? What are some alternatives?
58. How would you optimize a slow-performing Symfony application? What tools and techniques would you use?
59. Describe how you would implement a custom security voter in Symfony to control access to specific resources.
60. Explain the purpose of the Symfony profiler and how you can use it to debug and optimize your application.
61. How can you implement a custom console command in Symfony? What are some use cases for custom commands?
62. What is the role of the Symfony serializer component? How can you customize the serialization and deserialization process?
63. How would you implement a multi-language website in Symfony? What are the key considerations?
64. Explain the difference between Symfony environments (dev, prod, test) and how they affect application behavior.
65. How can you use Symfony's cache component to improve performance? What are different caching strategies?
66. Describe how you would implement a RESTful API in Symfony, including routing, request handling, and response formatting.
67. What are data transformers in Symfony forms and when might you use them?
68. Explain the purpose of using the `bin/console` command in Symfony and give some common examples.
69. How do you handle different types of relationships when using Doctrine in Symfony (e.g., one-to-many, many-to-many)?
70. What are some strategies for writing unit and functional tests in a Symfony project? What tools would you use?
71. How can you use the Symfony Messenger component to handle asynchronous tasks and message queues?
72. Describe how you would implement a custom Twig filter or function in Symfony.
73. How would you approach optimizing a slow-performing Symfony application, and what tools would you use?
74. Describe your experience with different Symfony environments (dev, prod, test) and how you manage configurations across them.
75. Explain the concept of service decoration in Symfony and provide a practical example of when you would use it.
76. How do you handle complex form scenarios in Symfony, such as dynamic forms or forms with multiple entities?
77. Discuss your experience with implementing API authentication in Symfony, including different methods like OAuth or JWT.
78. How would you implement a custom event dispatcher in Symfony, and what are the benefits of using event dispatching?
79. Explain the role of the Symfony profiler and how you use it to identify and resolve performance bottlenecks.
80. Describe your experience with using Symfony Messenger for asynchronous task processing and how it integrates with different transports.
81. How do you ensure code quality and maintainability in a large Symfony project, including code reviews and automated testing?
82. Explain the difference between using a custom repository method and a DQL query in Symfony, and when you would choose one over the other.
83. How do you handle database migrations in a Symfony project, especially when working in a team environment?
84. Describe your experience with integrating third-party libraries or services into a Symfony application.
85. How would you implement a custom validator in Symfony to enforce specific business rules?
86. Explain the purpose of the Symfony security component and how you would configure it for different user roles and permissions.
87. How do you approach debugging complex issues in a Symfony application, and what strategies do you use to identify the root cause?
88. Describe your experience with implementing caching strategies in Symfony to improve application performance.
89. How would you implement a custom console command in Symfony to perform a specific task?
90. Explain the concept of dependency injection in Symfony and how it promotes loose coupling and testability.
91. How do you handle file uploads in Symfony, including validation and storage considerations?
92. Describe your experience with using Symfony's built-in testing tools, such as PHPUnit and functional testing.
93. How do you approach internationalization (i18n) and localization (l10n) in a Symfony application?
94. Explain the role of the Symfony router and how you define custom routes for different application features.
95. How would you implement a custom Twig filter or function in Symfony to extend the templating engine's functionality?
96. Describe your experience with deploying Symfony applications to different environments, such as cloud platforms or traditional servers.