

# 102 Magento 2 Interview Questions to Hire Top Developers

## Questions

1. What is Magento 2, in simple terms?
2. Can you name some key benefits of using Magento 2?
3. What are the different Magento 2 editions, and who are they for?
4. What is the purpose of the di.xml file in Magento 2?
5. Explain what a module is in Magento 2.
6. What's the difference between an area code and a scope in Magento 2?
7. How would you enable or disable a module in Magento 2?
8. What are the different types of events in Magento 2?
9. How do you create a custom theme in Magento 2?
10. Explain the purpose of the setup:upgrade command.
11. What is the difference between a block, a template, and a layout in Magento 2?
12. How do you clear the cache in Magento 2?
13. What is the purpose of the pub/static directory?
14. How does Magento 2 handle indexing?
15. What is the purpose of the object manager?
16. Explain how you would override a block in Magento 2.
17. Describe the role of observers in Magento 2.
18. What is the purpose of the composer.json file in a Magento 2 module?
19. How would you add a custom JavaScript file to a Magento 2 theme?
20. What is the difference between a virtual type and a preference in di.xml?
21. How do you create a new product attribute in Magento 2?
22. Explain the purpose of UI components in Magento 2.
23. Describe the process of creating a simple module that displays 'Hello World' on a page.
24. Describe the process of creating a custom module in Magento 2, including the necessary files and their functions.
25. Explain the role of the di.xml file in Magento 2 and how it is used for dependency injection.
26. What are interceptors in Magento 2, and how can they be used to modify the behavior of existing classes?
27. How do you create a custom layout update in Magento 2 and apply it to a specific page?
28. Explain the difference between a virtual type and a preference in Magento 2's di.xml.
29. Describe the different types of events in Magento 2 (e.g., global, area-specific) and how to dispatch and observe them.
30. How do you create a custom product attribute in Magento 2, and what are the different input types available?
31. Explain the purpose of the Magento 2 UI component library and how it is used to build admin grids and forms.
32. How do you create a custom command-line interface (CLI) command in Magento 2?
33. Describe the different cache types in Magento 2 and how to invalidate them programmatically.
34. How can you override a core Magento 2 block, model, or helper?
35. Explain how to create a custom shipping method in Magento 2.
36. What are plugins in Magento 2, and what are the different plugin types (before, after, around)?
37. Describe the process of creating a custom theme in Magento 2, including the fallback mechanism.
38. How do you use the Magento 2 service contracts to create a web API?
39. Explain how to use the Magento 2 logging system to debug and troubleshoot issues.
40. How do you create a custom widget in Magento 2 and configure its parameters?
41. Describe the purpose of the setup scripts (InstallData, UpgradeData, InstallSchema, UpgradeSchema) in Magento 2 modules.
42. How do you implement a custom cron job in Magento 2?
43. Explain how to use the Magento 2 message queue system for asynchronous processing.
44. How do you create a custom payment gateway integration in Magento 2?
45. Describe how you would debug a performance issue in Magento 2.
46. How can you extend the Magento 2 REST API to add custom endpoints or modify existing ones?
47. Explain how to work with the Magento 2 customer session.
48. How can you add a custom system configuration setting in Magento 2?
49. Describe your experience with automated testing in Magento 2 (e.g., unit tests, integration tests).
50. How do you handle database transactions in Magento 2?
51. Explain the concept of the Magento 2 object manager.
52. How do you implement a custom validator for a form field in Magento 2?
53. Describe how you would optimize a Magento 2 store for SEO.
54. Explain the implications of using the same event observer for multiple events in Magento 2. What are the potential drawbacks and how can they be mitigated?
55. Describe a scenario where you would implement a plugin on a public interface versus rewriting a class in Magento 2. What are the trade-offs?
56. How does Magento 2's dependency injection container handle circular dependencies? What strategies can you employ to resolve them?
57. Explain the process of creating and using a custom indexer in Magento 2. What are the key considerations for performance and scalability?
58. Describe how you would implement a custom cache type in Magento 2. What configuration is required, and how would you invalidate the cache?
59. What are the different types of virtual types in Magento 2, and how do they impact performance and maintainability?
60. Explain the role of message queues (e.g., RabbitMQ) in Magento 2, and describe a use case where they would be beneficial.
61. How does Magento 2 handle versioning of database schema? Explain the process of creating and applying upgrade scripts.
62. Describe the process of creating a custom admin grid with mass actions and filters in Magento 2. What are the key components involved?
63. Explain how you would implement a multi-tenant architecture in Magento 2, where multiple stores share the same codebase and database.
64. What are the different types of collectors in Magento 2's sales rules, and how do they impact the cart price rules application process?
65. Describe the implications of using different session storage options (e.g., file, database, Redis) in Magento 2. What are the performance and security considerations?
66. How does Magento 2's code generation framework work? Explain the role of interceptors and factories in the process.
67. Explain the difference between using before, after, and around plugins in Magento 2. Provide a specific example for each plugin type.
68. Describe how you would implement a custom payment gateway in Magento 2, including the steps involved in processing payments and handling callbacks.
69. What are the different types of events in Magento 2 (e.g., global, area-specific), and how do they differ in terms of scope and availability?
70. Explain how you would optimize Magento 2's performance for a high-traffic e-commerce website, including caching strategies, database optimization, and code profiling.
71. Describe the process of creating a custom widget in Magento 2, including the configuration options and template rendering.
72. How does Magento 2 handle customer segments? Explain the different types of conditions and actions that can be used to define segments.
73. Explain the role of the service contracts in Magento 2. How can you ensure that your custom modules adhere to best practices for service contracts?
74. Describe how you would implement a custom shipping method in Magento 2, including the steps involved in calculating shipping rates and displaying options to the customer.
75. What are the key considerations for securing a Magento 2 website against common vulnerabilities such as SQL injection and cross-site scripting (XSS)?
76. Explain the nuances between interceptors and plugins in Magento 2.
77. Describe advanced techniques for optimizing Magento 2 performance in high-traffic scenarios.
78. Discuss your experience with implementing complex data import/export processes in Magento 2.
79. How would you approach debugging a complex performance bottleneck in a Magento 2 application?
80. Describe your experience with different caching strategies in Magento 2 and when to use them.
81. Explain the architectural differences between Magento 1 and Magento 2 and the reasons for the changes.
82. Describe how you would implement a complex custom shipping module with advanced rate calculation logic in Magento 2.
83. Discuss your experience with using message queues for asynchronous processing in Magento 2.
84. Explain how you would implement a custom payment gateway integration with complex transaction processing requirements in Magento 2.
85. Describe your approach to securing a Magento 2 application against common web vulnerabilities.
86. How do you handle complex dependency injection scenarios involving circular dependencies in Magento 2?
87. Explain the role and implementation of Elasticsearch in a Magento 2 environment.
88. Describe your experience with contributing to open-source Magento 2 projects or developing custom modules.
89. How would you approach load testing a Magento 2 application to identify performance bottlenecks?
90. Explain the concepts of Domain-Driven Design (DDD) and how they can be applied in a Magento 2 project.
91. Describe the process of creating and managing API integrations (REST or GraphQL) in Magento 2.
92. How do you stay up-to-date with the latest Magento 2 releases, security patches, and best practices?
93. Explain how you would implement a multi-website or multi-store setup with complex shared catalog and customer management rules.
94. Describe your experience with using tools for automated testing (unit, integration, functional) in Magento 2 projects.
95. Explain the process of customizing the Magento 2 checkout flow with complex validation and data manipulation requirements.
96. Describe the differences between a ViewModel and a Block in Magento 2 and when you would use each.
97. Explain the nuances between different types of events in Magento 2 (e.g., global, area-specific).
98. Describe your approach to handling complex data migrations from older versions of Magento or other platforms to Magento 2.
99. How would you approach integrating Magento 2 with a third-party ERP or CRM system with complex data synchronization requirements?
100. Explain the role of service contracts and data interfaces in Magento 2 module development.
101. Describe the steps involved in creating a custom indexer in Magento 2 for optimizing search performance.