Top 25 API Interview Questions and Answers

Introduction

API Interview Questions and Answers help you prepare for developer hiring rounds. Many companies build web apps, mobile apps, and cloud services. These systems exchange data through APIs. A developer role often requires clear knowledge of API structure, request methods, response formats, and security methods.

Interview panels test both concept knowledge and real work experience. You face questions about REST APIs, HTTP methods, authentication, and error handling. Strong preparation improves performance during technical rounds. You answer with clarity and confidence.

This article presents common API Interview Questions and Answers used in developer interviews. Each question focuses on practical knowledge used during software development projects. Use these questions to review key topics before your next interview.

List of API Interview Questions for Freshers

1. What's an API?

2. What advantages can RESTful web services offer?

3. What is Web API?

4. What are the uses of web APIs?

5. Which are the primary return types that the Web API supports?

6. What is Web API 2.0?

7. What are JSON and XML?

8. Who can use Web API?

9. In terms of an API, what is CRUD?

10. What distinguishes web services from APIs?

11. Which styles are available for web API creation?

12. What does API testing entail?

13. Which tools do software engineers use for API testing?

API Technical Interview Questions and Answers for Freshers

1. What's an API?

Application Programming Interface is referred to as an API. It's a kind of software that enables data or functionality sharing and communication between two apps. It is made up of several components, such as subroutines, which enable it to carry out the intended tasks.

2. What advantages can RESTful web services offer?

Because they are simple to maintain, software developers who create APIs for web-based applications frequently use RESTful web services.

They are adaptable, lightweight, and simple to scale, should a software engineer see advancements they'd like to see made.

3. What is Web API?

As the name implies, a Web API (Application Programming Interface) is an API that can be accessed via the HTTP protocol over the Internet. It's a mechanism that server code can use to read and push data to a server. It can be created or developed with various technologies, including ASP.NET, Java, and others.

4. What are the uses of web APIs?

Uses for Web APIs:

- It has extra layers that do nothing more than standardize communications and offer various input and output formatting options.
- It is compatible with several web application formats, including ASP.NET WebForms, and ASP.NET MVC.
- Web API services are seen as the best option for creating resource-oriented services.
- It also aids in the development of SOAP-based and RESTful services.

5. Which are the primary return types that the Web API supports?

There is no particular data type for it. Any kind of data can be returned, based on what the business needs. Numerous HTTP methods, such as GET, POST, PUT, and others, can return data in various formats based on the use case.

6. What is Web API 2.0?

It is essentially an improved and changed Web API. This version supports the add-on features listed below:

- New Routing Property
- ASP.NET Web API security with OAuth 2.0
- Assistance with Cross-Origin requests through CORS
- Return type for IHttpActionResult
- OData Service Support for $expand, $select

Web API 2.0 is regarded as the best option and a suitable development model that makes it easier to develop because of all its new capabilities.

RESTful services provide interfaces to a range of clients operating on different hardware.

Route configuration at the Web API method or controller level is also supported.

7. What are JSON and XML?

Extensible Markup Language, or XML:

- It is specifically made to move and store data.
- It is comparable to HTML, but because it lets users design unique tags, it is more flexible than HTML.
- It represents structured data, which includes documents, data, configuration, etc.

JavaScript Object Notation, or JSON:

- It is a compact format made for data storage and transportation.
- It is a common text-based format that is simpler to comprehend and uses JavaScript object syntax to describe structured data.
- It is simpler to use and faster.

8. Who can use Web API?

A wide variety of clients use or use web APIs, including browsers, mobile devices, iPhones, etc. Additionally, it works well with native apps that need

web services but don’t support SOAP. Every client that is capable of utilizing HTTP verbs like GET, DELETE, POST, and PUT can also utilize it.

9. In terms of an API, what is CRUD?

Create, read, update, and delete is referred to as CRUD. It acts as a stickable acronym to help developers think through and incorporate the four components of a viable, usable application.

10. What distinguishes web services from APIs?

Software engineers use three different communication styles—SOAP, REST, and XML-RPC—to deliver web services over the internet. They can never function without a network. On the other hand, APIs support a variety of communication channels. They don’t require a network to function.

11. Which styles are available for web API creation?

XML/JSON formatting, stateless communication, a basic URI as the service address, and HTTP client-server communication are common architectural paradigms for Web APIs.

12. What does API testing entail?

- Testing APIs makes the application accessible without requiring a user interface.
- This enables you to find little bugs during graphical user interface (GUI) testing before they become major ones.
- Additionally, because API testing requires less code than GUI testing, it usually takes less time to complete.
- It consequently provides more economical and effective test coverage.

- The fact that XML or JSON, is used for data transport is another advantage of API testing.
- When selecting automation testing services, users can choose any coding language due to these language-independent forms of interaction.

Check out: API Course Syllabus PDF

13. Which tools do software engineers use for API testing?

Among the well-liked tools are Apigee, SoapUI Pro, Katalon Studio, Postman, and Tricentis Tosca.

List of API Interview Questions for Experienced

14. Describe SOAP.

15. What distinguishes API testing from user interface testing?

16. What methods are there for simulating user input?

17. Describe the API documentation.

18. How do idempotent procedures work?

19. In terms of RESTful web services, what are payloads?

20. What is the largest payload size that POST methods allow sending?

21. Distinguish between PUT and POST techniques.

22. Which HTTP response status codes do you encounter most frequently while using REST APIs?

23. Describe a resource.

24. What is a real-world REST API example?

25. How should resources be represented in a RESTful web service design?

API Technical Interview Questions and Answers for Experienced

14. Describe SOAP.

The messaging protocol known as SOAP, or Simple Object Access Protocol, is based on XML. It facilitates information transfer between computers. A SOAP API can be used to create, locate, remove, or edit records.

15. What distinguishes API testing from user interface testing?

User interface, or UI, testing is the process of analyzing an application's graphical user interface, including how the user may interact with its features. API testing establishes a channel of communication that permits two software systems to exchange functions and subroutines.

16. What methods are there for simulating user input?

To achieve input injection, you can use low-level input, robots, device drivers, and so on.

17. Describe the API documentation.

The procedure depends on having quality API documentation. When utilizing a program, it provides a fast reference. The plan and delivery layout are provided. It also provides sources for the content and describes each system function in depth.

18. How do idempotent procedures work?

Idempotent techniques prevent problems brought on by duplicate requests on the client side and are known to deliver the same result even after the same request has been made numerous times.

19. In terms of RESTful web services, what are payloads?

In RESTful web services, payloads are the request data that is contained in the message body of an HTTP request sent through the POST or GET method.

20. What is the largest payload size that POST methods allow sending?

The maximum payload size that can be supplied using POST methods is theoretically unlimited. Larger payloads, however, may require more bandwidth. As a result, it can take the server longer to process the request.

21. Distinguish between PUT and POST techniques.

POST Method

- On the server, a resource may be created via POST.
- POST lacks idempotence.
- Cacheable responses to POSTs exist.

PUT Method

- PUT is used to substitute a different resource for an existing one at a given URI.
- PUT is idempotent, meaning that even after several calls, it will always return a single resource.
- PUT responses are not cacheable.

22. Which HTTP response status codes do you encounter most frequently while using REST APIs?

The most frequent response status codes are 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error, 502 Bad Gateway, 503 Service Unavailable, and 200 OK.

23. Describe a resource.

A resource in REST is an item on the server that has a label and can be accessed. Resources include linked data, a set of procedures, and connections to other resources on the server.

24. What is a real-world REST API example?

- Weather apps use public REST APIs to display weather data and exchange associated information.
- Airlines provide flight schedules and prices using APIs, which enable travel and ticketing websites for businesses.

- Public transportation companies employ APIs to open up their data, allowing real-time mapping and navigation app access.

25. How should resources be represented in a RESTful web service design?

- Both the client and the server should be able to grasp it easily.
- It must be comprehensive, regardless of its arrangement.
- It must properly handle and take into account how the resources relate to one another.

Conclusion

API knowledge holds strong value during developer interviews. Recruiters look for clear understanding of API structure, request methods, response formats, authentication, and error handling. When you explain these topics with clarity, interview discussions become easier and more confident.

Spend time reviewing common **API interview questions and answers**. Practice answers with examples from projects or training sessions. This habit improves technical communication and helps you respond with accuracy during hiring rounds.

If you want structured guidance and hands on practice, join our API Training in Chennai. You learn API development through real project work and expert guidance. This training supports strong technical skills and better preparation for developer interviews.