# Hypertext Transfer Protocol (HTTP) 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 Hypertext Transfer Protocol (HTTP) 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 Hypertext Transfer Protocol (HTTP) 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 Hypertext Transfer Protocol (HTTP) category. To ensure quality, each submission is checked by our team, before it becomes live. This Hypertext Transfer Protocol (HTTP) 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

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**

# Hypertext Transfer Protocol (HTTP) Interview Questions And Answers Guide.

**Question - 1:**

Describe the different roles of HTTP?

**Ans:**

In HTTP, there are two different roles: server and client. In general, the client always initiates the conversation; the server replies. HTTP is text based; that is, messages are essentially bits of text, although the message body can also contain other media. Text usage makes it easy to monitor an HTTP exchange.

View All Answers

**Question - 2:**

What is the mean of GET?

**Ans:**

GET is the simplest type of HTTP request method; the one that browsers use each time you click a link or type a URL into the address bar. It instructs the server to transmit the data identified by the URL to the client. Data should never be modified on the server side as a result of a GET request. In this sense, a GET request is read-only, but of course, once the client receives the data, it is free to do any operation with it on its own side - for instance, format it for display.

View All Answers

**Question - 3:**

What are response codes in HTTP?

**Ans:**

HTTP response codes standardize a way of informing the client about the result of its request.
You might have noticed that the example application uses the PHP header(), passing some strange looking strings as arguments. The header() function prints the HTTP headers and ensures that they are formatted appropriately. Headers should be the first thing on the response, so you shouldn't output anything else before you are done with the headers. Sometimes, your HTTP server may be configured to add other headers, in addition to those you specify in your code.

View All Answers

**Question - 4:**

What are status codes in HTTP?

**Ans:**

In HTTP/1.0 and since, the first line of the HTTP response is called the status line and includes a numeric status code (such as "404") and a textual reason phrase (such as "Not Found"). The way the user agent handles the response primarily depends on the code and secondarily on the response headers. Custom status codes can be used since, if the user agent encounters a code it does not recognize, it can use the first digit of the code to determine the general class of the response.

View All Answers

**Question - 5:**

Explain about persistent connections?

**Ans:**

In HTTP/0.9 and 1.0, the connection is closed after a single request/response pair. In HTTP/1.1 a keep-alive-mechanism was introduced, where a connection could be reused for more than one request.

View All Answers

**Question - 6:**

Explain secure HTTP?

**Ans:**

There are currently two methods of establishing a secure HTTP connection: the https URI scheme and the HTTP 1.1 Upgrade header, introduced by RFC 2817. Browser support for the Upgrade header is, however, nearly non-existent, so HTTPS is still the dominant method of establishing a secure HTTP connection. Secure

HTTP is notated by the prefix https:// instead of http:// on web URIs.

View All Answers

**Question - 7:**

Explain request message in HTTP?

**Ans:**

The request message consists of the following:
* Request line, such as GET /images/logo.png HTTP/1.1, which requests a resource called /images/logo.png from server
* Headers, such as Accept-Language: en
* An empty line
* An optional message body
The request line and headers must all end with <CR><LF> (that is, a carriage return followed by a line feed). The empty line must consist of only <CR><LF> and no other white-space. In the HTTP/1.1 protocol, all headers except Host are optional.
A request line containing only the path name is accepted by servers to maintain compatibility with HTTP clients before the HTTP/1.0 specification in RFC1945.

View All Answers

**Question - 8:**

Tell me what is HTTP session state?

**Ans:**

HTTP is a stateless protocol. A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests. For example, when a web server is required to customize the content of a web page for a user, the web application may have to track the user's progress from page to page. A common solution is the use of HTTP cookies. Other methods include server side sessions, hidden variables (when the current page is a form), and URL-rewriting using URI-encoded parameters, e.g., /index.php?session_id=some_unique_session_code.

View All Answers

**Question - 9:**

What is Idempotent methods and web applications?

**Ans:**

Methods PUT and DELETE are defined to be idempotent, meaning that multiple identical requests should have the same effect as a single request. Methods GET, HEAD, OPTIONS and TRACE, being prescribed as safe, should also be idempotent, as HTTP is a stateless protocol.

View All Answers

**Question - 10:**

Tell me what happens to an undeliverable datagram?

**Ans:**

An undeliverable datagram is discarded and an ICMP error message is sent to the source host.

View All Answers

**Question - 11:**

Explain IP is a best-effort protocol in HTTP?

**Ans:**

IP is a best-effort protocol, because it will make every effort to always transmit a datagram and also datagrams will not be just discarded. However, the delivery of the datagram to the destination is not guaranteed.

View All Answers

**Question - 12:**

Tell me what do you mean by IP is an unreliable protocol?

**Ans:**

IP is a unreliable protocol because it does not guarantee the delivery of a data-gram to its destination. The reliability must be provided by the upper layer protocols like TCP. IP does not support flow control, re-transmission, acknowledgement and error recovery.

View All Answers

**Question - 13:**

Tell me to which OSI layer does IP belong?

**Ans:**

IP belongs to the Network Layer (layer 3) in the OSI model.
internet protocol is working in network layer of osi model in congection with tcp tx layer protocol.

View All Answers

**Question - 14:**

What is the mean of PUT?

**Ans:**

A PUT request is used when you wish to create or update the resource identified by the URL. For example,
1 PUT /clients/robin

might create a client, called Robin on the server. You will notice that REST is completely backend agnostic; there is nothing in the request that informs the server how the data should be created - just that it should. This allows you to easily swap the backend technology if the need should arise. PUT requests contain the data to use in updating or creating the resource in the body. In cURL, you can add data to the request with the -d switch.
1 curl -v -X PUT -d "some text"

View All Answers

**Question - 15:**
What is the mean of 500 Internal Server Error HTTP response codes?

**Ans:**
When all else fails; generally, a 500 response is used when processing fails due to unanticipated circumstances on the server side, which causes the server to error out.

View All Answers

**Question - 16:**
What is the mean of 409 Conflict HTTP response codes?

**Ans:**
This indicates a conflict. For instance, you are using a PUT request to create the same resource twice.

View All Answers

**Question - 17:**
What is the mean of 405 Method Not Allowed HTTP response codes?

**Ans:**
The HTTP method used is not supported for this resource.

View All Answers

**Question - 18:**
What is the mean of 401 Unauthorized HTTP response codes?

**Ans:**
This error indicates that you need to perform authentication before accessing the resource.

View All Answers

**Question - 19:**
What is the mean of 404 Not Found HTTP response codes?

**Ans:**
This response indicates that the required resource could not be found. This is generally returned to all requests which point to a URL with no corresponding resource.

View All Answers

**Question - 20:**
What is the mean of 400 Bad Request HTTP response codes?

**Ans:**
The request was malformed. This happens especially with POST and PUT requests, when the data does not pass validation, or is in the wrong format.

View All Answers

**Question - 21:**
What is the mean of 201 Created HTTP response codes?

**Ans:**
This indicates the request was successful and a resource was created. It is used to confirm success of a PUT or POST request.

View All Answers

**Question - 22:**
What is 200 OK HTTP response codes?

**Ans:**
This response code indicates that the request was successful.

View All Answers

**Question - 23:**
What is HTTP Client Libraries?

**Ans:**
cURL is, more often than not, the HTTP client solution of choice for PHP developers.
To experiment with the different request methods, you need a client, which allows you to specify which method to use. Unfortunately, HTML forms do not fit the bill, as they only allow you to make GET and POST requests. In real life, APIs are accessed programmatically through a separate client application, or through JavaScript in the browser.
This is the reason why, in addition to the server, it is essential to have good HTTP client capabilities available in your programming language of choice.

A very popular HTTP client library is, again, cURL. You've already been familiarized with the cURL command from earlier in this tutorial. cURL includes both a standalone command line program, and a library that can be used by various programming languages. In particular, cURL is, more often than not, the HTTP client solution of choice for PHP developers. Other languages, such as Python, offer more native HTTP client libraries.

View All Answers

### Question - 24:

What is Representations in HTTP?

### Ans:

The HTTP client and HTTP server exchange information about resources identified by URLs.
We can sum up what we have learned so far in the following way: the HTTP client and HTTP server exchange information about resources identified by URLs.
We say that the request and response contain a representation of the resource. By representation, we mean information, in a certain format, about the state of the resource or how that state should be in the future. Both the header and the body are pieces of the representation.
The HTTP headers, which contain metadata, are tightly defined by the HTTP spec; they can only contain plain text, and must be formatted in a certain manner.
The body can contain data in any format, and this is where the power of HTTP truly shines. You know that you can send plain text, pictures, HTML, and XML in any human language. Through request metadata or different URLs, you can choose between different representations for the same resource. For example, you might send a webpage to browsers and JSON to applications.

View All Answers

### Question - 25:

What is Idempotent methods in HTTP?

### Ans:

These methods achieve the same result, no matter how many times the request is repeated: they are GET, PUT, and DELETE. The only non idempotent method is POST. PUT and DELETE being considered idempotent might be surprising, though, it, in fact, is quite easy to explain: repeating a PUT method with exactly the same body should modify a resource in a way that it remains identical to the one described in the previous PUT request: nothing will change! Similarly, it makes no sense to delete a resource twice. It follows that no matter how many times a PUT or DELETE request is repeated, the result should be the same as if it had been done only once.

View All Answers

### Question - 26:

What are the safe and unsafe methods of HTTP?

### Ans:

safe methods are those that never modify resources. The only safe methods, from the four listed above, is GET. The others are unsafe, because they may result in a modification of the resources.

View All Answers

### Question - 27:

For what purposes POST is used?

### Ans:

POST is used when the processing you wish to happen on the server should be repeated, if the POST request is repeated (that is, they are not idempotent; more on that below). In addition, POST requests should cause processing of the request body as a subordinate of the URL you are posting to.
In plain words:
1 POST /clients/

View All Answers

### Question - 28:

Described the most important HTTP verbs?

### Ans:

If you've ever created HTML forms, you'll be familiar with two of the most important HTTP verbs: GET and POST. But there are far more HTTP verbs available. The most important ones for building Restful API are GET, POST, PUT and DELETE. Other methods are available, such as HEAD and OPTIONS, but they are more rare (if you want to know about all other HTTP methods, the official source is IETF).

View All Answers

### Question - 29:

What are the mean of HTTP verbs?

### Ans:

HTTP verbs tell the server what to do with the data identified by the URL.
HTTP verbs tell the server what to do with the data identified by the URL. The request can optionally contain additional information in its body, which might be required to perform the operation - for instance, data you want to store with the resource. You can supply this data in cURL with the -d option.

View All Answers

### Question - 30:

What is the mean of URLS in HTTP?

### Ans:

URLs are how you identify the things that you want to operate on. We say that each URL identifies a resource. These are exactly the same URLs which are assigned to web pages. In fact, a web page is a type of resource.

View All Answers

**Question - 31:**

What is cURL in HTTP?

**Ans:**

cURL is a command line tool that is available on all major operating systems.

View All Answers

**Question - 32:**

From what HTTP messages are made?

**Ans:**

HTTP messages are made of a header and a body. The body can often remain empty; it contains data that you want to transmit over the network, in order to use it according to the instructions in the header. The header contains metadata, such as encoding information; but, in the case of a request, it also contains the important HTTP methods. In the REST style, you will find that header data is often more significant than the body.

View All Answers

**Question - 33:**

What is https URI scheme?

**Ans:**

https is a URI scheme that is, aside from the scheme token, syntactically identical to the http scheme used for normal HTTP connections, but which signals the browser to use an added encryption layer of SSL/TLS to protect the traffic. SSL is especially suited for HTTP since it can provide some protection even if only one side of the communication is authenticated. This is the case with HTTP transactions over the Internet, where typically only the server is authenticated

View All Answers

**Question - 34:**

What is Secure HTTP?

**Ans:**

There are currently two methods of establishing a secure HTTP connection: the https URI scheme and the HTTP 1.1 Upgrade header, introduced by RFC 2817. Browser support for the Upgrade header is, however, nearly non-existent, so HTTPS is still the dominant method of establishing a secure HTTP connection. Secure HTTP is notated by the prefix https:// instead of http:// on web URIs.

View All Answers

**Question - 35:**

What is HTTP session state?

**Ans:**

HTTP is a stateless protocol. A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests. For example, when a web server is required to customize the content of a web page for a user, the web application may have to track the user's progress from page to page. A common solution is the use of HTTP cookies. Other methods include server side sessions, hidden variables (when the current page is a form), and URL-rewriting using URI-encoded parameters, e.g., /index.php?session_id=some_unique_session_code.

View All Answers

**Question - 36:**

What is Persistent connections?

**Ans:**

In HTTP/0.9 and 1.0, the connection is closed after a single request/response pair. In HTTP/1.1 a keep-alive-mechanism was introduced, where a connection could be reused for more than one request.

View All Answers

**Question - 37:**

What is Status codes?

**Ans:**

In HTTP/1.0 and since, the first line of the HTTP response is called the status line and includes a numeric status code (such as "404") and a textual reason phrase (such as "Not Found"). The way the user agent handles the response primarily depends on the code and secondarily on the response headers. Custom status codes can be used since, if the user agent encounters a code it does not recognize, it can use the first digit of the code to determine the general class of the response.

View All Answers

**Question - 38:**

Explain Idempotent methods and web applications?

**Ans:**

Methods PUT and DELETE are defined to be idempotent, meaning that multiple identical requests should have the same effect as a single request. Methods GET, HEAD, OPTIONS and TRACE, being prescribed as safe, should also be idempotent, as HTTP is a stateless protocol.

View All Answers

**Question - 39:**

Explain Safe methods?

**Ans:**

Some methods (for example, HEAD, GET, OPTIONS and TRACE) are defined as safe, which means they are intended only for information retrieval and should not change the state of the server. In other words, they should not have side effects, beyond relatively harmless effects such as logging, caching, the serving of banner advertisements or incrementing a web counter. Making arbitrary GET requests without regard to the context of the application's state should therefore be considered safe.
By contrast, methods such as POST, PUT and DELETE are intended for actions which may cause side effects either on the server, or external side effects such as financial transactions or transmission of email. Such methods are therefore not usually used by conforming web robots or web crawlers, which tend to make requests without regard to context or consequences.

View All Answers

**Question - 40:**

What are request methods?

**Ans:**

HEAD
Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.
GET
Requests a representation of the specified resource. Requests using GET (and a few other HTTP methods) "SHOULD NOT have the significance of taking an action other than retrieval". The W3C has published guidance principles on this distinction, saying, "Web application design should be informed by the above principles, but also by the relevant limitations." See safe methods below.
POST
Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.
PUT
Uploads a representation of the specified resource.
DELETE
Deletes the specified resource.
TRACE
Echoes back the received request, so that a client can see what (if any) changes or additions have been made by intermediate servers.
OPTIONS
Returns the HTTP methods that the server supports for specified URL. This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.
CONNECT
Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.
PATCH
Is used to apply partial modification

View All Answers

**Question - 41:**

What is a Request message?

**Ans:**

The request message consists of the following:
* Request line, such as GET /images/logo.png HTTP/1.1, which requests a resource called /images/logo.png from server
* Headers, such as Accept-Language: en
* An empty line
* An optional message body
The request line and headers must all end with <CR><LF> (that is, a carriage return followed by a line feed). The empty line must consist of only <CR><LF> and no other whitespace. In the HTTP/1.1 protocol, all headers except Host are optional.
A request line containing only the path name is accepted by servers to maintain compatibility with HTTP clients before the HTTP/1.0 specification in RFC1945

View All Answers

**Question - 42:**

What is HTTP session?

**Ans:**

HTTP session is a sequence of network request-response transactions. An HTTP client initiates a request. It establishes a Transmission Control Protocol (TCP) connection to a particular port on a host (typically port 80; see List of TCP and UDP port numbers). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own, the body of which is perhaps the requested resource, an error message, or some other information.

View All Answers

**Question - 43:**

What is Hypertext Transfer Protocol (HTTP)?

**Ans:**

Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.
The standards development of HTTP has been coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium, culminating in the publication of a series of Requests for Comments (RFCs), most notably RFC 2616 (June 1999), which defines HTTP/1.1, the version of HTTP in common use

View All Answers

# Protocols Most Popular & Related Interview Guides

1 : **TCP/IP Interview Questions and Answers.**

**2 : Internet Protocol (IP) Address Interview Questions and Answers.**

**3 : SS7 Interview Questions and Answers.**

**4 : Spanning Tree Protocol (STP) Interview Questions and Answers.**

**5 : Secured Socket Layer (SSL) Interview Questions and Answers.**

**6 : UDP Interview Questions and Answers.**

**7 : SNMP Interview Questions and Answers.**

**8 : FTP Interview Questions and Answers.**

**9 : Networking Protocols Interview Questions and Answers.**

**10 : SMTP Interview Questions and Answers.**

**Follow us on FaceBook**
**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**