

# Understanding REST API status codes

**Estimated time needed:** 4 minutes

## Learning objectives:

- Explain the significance of REST API status codes in web development
- Identify and categorize different HTTP status codes
- Understand common status codes and their use cases
- Differentiate between client-side and server-side errors

In modern web development, Representational State Transfer (REST) APIs are fundamental for communication between software systems. These APIs rely on HTTP status codes to provide feedback on request outcomes, helping developers understand whether their requests were successful, redirected, encountered client errors, or suffered from server issues. Mastering these status codes is essential for building and maintaining reliable API-driven applications.

This reading provides an overview of the REST API status codes, classifications, and use cases.

## Classification of REST API codes

REST API codes are classified into five categories based on their purpose and meaning. These categories are:

- **Informational codes (1xx):** Indicates that the request has been received and processing continues.
- **Success (2xx):** Confirms that the server successfully processed the request.
- **Redirection (3xx):** Signals that further action is needed to complete the request, often involving URL changes.
- **Client errors (4xx):** Indicates issues with the request made by the client, such as invalid data or unauthorized access.
- **Server errors (5xx):** Represents failures on the server side, where the request is valid, but the server cannot process it.

Understanding these categories helps developers troubleshoot and implement APIs effectively. The following table lists and explains the REST API codes for easy reference.

Code Category	Code	Indication
<b>Informational codes (1xx)</b>	100 Continue	The server received the request headers, implying the client can send the request body.
<b>Success codes (2xx)</b>	200 OK	The request was successful, and the response depends on the request method used.
	201 Created	A new resource was created, which is often the result of a successful POST request.
	204 No Content	The request was processed successfully. However, there is no content to return, commonly used for DELETE operations.
<b>Redirection codes (3xx)</b>	301 Moved Permanently	The requested resource's location has been permanently moved to a new URL.
	302 Found	The requested resource is temporarily available under a different URL.

Code Category	Code	Indication
	304 Not Modified	The requested resource has not changed since the last retrieval, allowing the client to use a cached version.
<b>Client error codes (4xx)</b>	400 Bad Request	The request either contains invalid data or is malformed.
	401 Unauthorized	Authentication is required to access the resource.
	403 Forbidden	The server refuses to authorize the request despite understanding it.
	404 Not Found	The requested resource does not exist on the server.
	429 Too Many Requests	The client has sent too many requests in a given time period. Used in rate-limiting scenarios.
	451 Unavailable for Legal Reasons	Resource is unavailable due to legal restrictions.
<b>Server error codes (5xx)</b>	500 Internal Server Error	A general error indicating something went wrong on the server's side.
	502 Bad Gateway	An invalid response from an upstream server was received by the server.
	503 Service Unavailable	The server is experiencing temporary overload or maintenance and is, therefore, unable to handle the request.

## Summary

- REST API codes are standardized HTTP status codes that indicate the outcome or result of an API request.
- These codes help the client and server understand the status and actions required.
- REST API codes are classified into five categories.
  - **Informational codes** indicate that the request has been received and the server is processing it.
  - **Success codes** confirm that the server processed the request.
  - **Redirection codes** suggest the client needs further action, typically involving URL redirection.
  - **Client error codes** indicate that the request contains errors on the client's side and needs correction.
  - **Server error codes** show the server encountered an issue while processing a valid request.