

HTTP Status Codes

When REST API requests are sent to carryout various actions, various HTTP status codes will be returned based on the state of the action (success or failure) and the HTTP method (`POST`, `GET`, `PUT`, `DELETE`) executed. The following are the definitions of the various HTTP status codes that are returned:

- [Success HTTP status codes](#)
- [Error HTTP status codes](#)

Success HTTP status codes

Code	Code Summary	Description
100	Continue	The server has received the request headers, and the client should proceed to send the request body. This is applicable in the case of a request for which a body needs to be sent (e.g., <code>POST</code> request). This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server. The server will send a final response after the request has been completed.
200	Ok	HTTP request was successful. The output corresponding to the HTTP request will be returned. Generally used as a response to a successful <code>GET</code> and <code>PUT</code> REST API HTTP methods.
201	Created	HTTP request was successfully processed and a new resource was created. Generally used as a response to a successful <code>POST</code> REST API HTTP method.
202	Accepted	HTTP request was accepted for processing, but the processing has not been completed. This generally occurs when your successful in trying to deploy or undeploy an application.

Error HTTP status codes

Code	Code Summary	Description
400	Bad request	The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid JSON format).
403	Forbidden	The request was a valid request, but the server is refusing to respond to it. This status code appears when executing a metadata service REST API with an incorrect token.
404	Not found	Requested resource not found. Generally used as a response for unsuccessful <code>GET</code> and <code>PUT</code> REST API HTTP methods. This code will also be returned when trying to delete (<code>DELETE</code>) an artifact that does not exist.
409	Conflict	Request could not be processed because of conflict in the request. This generally occurs when you are trying to add a resource that already exists. For example, when trying to add an auto-scaling policy that has an already existing ID.
500	Internal server error	Server error occurred.