**HTTP Status codes**

HTTP response status codes indicate whether a specific [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) request has been successfully completed. Responses are grouped in five classes:

1. [Informational responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#information_responses) (100–199)
2. [Successful responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#successful_responses) (200–299)
3. [Redirection messages](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages) (300–399)
4. [Client error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#client_error_responses) (400–499)
5. [Server error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#server_error_responses) (500–599)

To automate API—first steps

//Create apache HTTP client

//Create a request message and send to the service

//Get the response

//validate the response

Get Method

|  |
| --- |
| public class CustomerGETApiTest {  public static void main(String[] args) throws ClientProtocolException, IOException {    //Create apache HTTP client  HttpClient client = HttpClientBuilder.create().build();    //Create a request message and send to the service  HttpGet resquestMsg= new HttpGet("https://gorest.co.in/public/v2/users/2643");    //Get the response  HttpResponse response = client.execute(resquestMsg);    //validate the response  System.out.println("StatusLine = "+ response.getStatusLine() +  "\nStatuusCode = " +response.getStatusLine().getStatusCode()  +"\nReasonPhrase = " +response.getStatusLine().getReasonPhrase()  +"\nProtocolVersion = " +response.getStatusLine().getProtocolVersion()  +"\nToString = " +response.getStatusLine().toString()  );  //get the response entity  HttpEntity entity = response.getEntity();  String result = "";  if (entity != null)  {  result = EntityUtils.toString(entity);  System.out.println("Entity = " + entity);  System.out.println("Result = " +result);  }    System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println(" Test Results");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");    //validate status code  int statusCode=response.getStatusLine().getStatusCode();  String statusMessage = response.getStatusLine().getReasonPhrase();  if (statusCode==200 && statusMessage.equals("OK")) {  System.out.println("StatusCode Test passed");  }  else {  System.out.println("Statuscode Test failed");  }    //Validate user\  boolean testPassed = result.contains("Chapal Nambeesan");  if(testPassed) {  System.out.println("Valid User");  }  else {  System.out.println("Test failed");  }  }  } |
|  |

Post Method with bearer authorization

|  |
| --- |
| Test  public class CustomerPOSTApiTest {  public static void main(String[] args) throws ClientProtocolException, IOException {  HttpClient client = HttpClientBuilder.create().build();  HttpPost request = new HttpPost("https://gorest.co.in/public/v2/users");    request.setHeader("Authorization", "Bearer ac008d3b46143c71c281ca6d326232c1a863981fde6004cebf805d509decdb88");    List <NameValuePair> urlParameters = new ArrayList<>();    urlParameters.add(new BasicNameValuePair("name", "Karunambika"));  urlParameters.add(new BasicNameValuePair("email", "karuna@gmmmm.com"));  urlParameters.add(new BasicNameValuePair("gender", "female"));  urlParameters.add(new BasicNameValuePair("status", "active"));    request.setEntity(new UrlEncodedFormEntity(urlParameters));  HttpResponse response = client.execute(request);  System.out.println("StatusCode" + response.getStatusLine().getStatusCode()  +response.getStatusLine().toString()  );  }  } |

PUT Method with bearer authorization

|  |
| --- |
| Test  public class CustomerPUTApiTest {  public static void main(String[] args) throws ClientProtocolException, IOException {  HttpClient client = HttpClientBuilder.create().build();  HttpPut request = new HttpPut("https://gorest.co.in/public/v2/users/4377");    request.setHeader("Authorization", "Bearer ac008d3b46143c71c281ca6d326232c1a863981fde6004cebf805d509decdb88");    List <NameValuePair> urlParameters = new ArrayList<>();  urlParameters.add(new BasicNameValuePair("email", "change\_karuna@yahoo.com"));  urlParameters.add(new BasicNameValuePair("gender", "female"));  urlParameters.add(new BasicNameValuePair("status", "active"));    request.setEntity(new UrlEncodedFormEntity(urlParameters));  HttpResponse response = client.execute(request);    System.out.println("Response = " + response.getStatusLine());  }  } |

DELETE Method with bearer authorization

|  |
| --- |
| Test  package com.apex.mavenproject;  import java.io.IOException;  import org.apache.http.HttpResponse;  import org.apache.http.client.ClientProtocolException;  import org.apache.http.client.HttpClient;  import org.apache.http.client.methods.HttpDelete;  import org.apache.http.impl.client.HttpClientBuilder;  public class CustomerDELETEApiTest {  public static void main(String[] args) throws ClientProtocolException, IOException {  HttpClient client = HttpClientBuilder.create().build();  HttpDelete request = new HttpDelete("https://gorest.co.in/public/v2/users/2642");  request.setHeader("Authorization", "Bearer ac008d3b46143c71c281ca6d326232c1a863981fde6004cebf805d509decdb88");  HttpResponse response = client.execute(request);  System.out.println("Response =" + response.getStatusLine().getStatusCode());    }  } |