Mocktito dependency :

<dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>3.6.0</version>  
 <scope>test</scope>  
</dependency>

Test getClient :

|  |
| --- |
| import static org.mockito.Mockito.when;  ...  @Test public void testGetRequest() throws IOException {  // Create a mock HTTP client  CloseableHttpClient httpClient = mock(CloseableHttpClient.class);   // Set up the mock to return a response when the GET method is called  HttpResponse response = new BasicHttpResponse(HttpVersion.HTTP\_1\_1, HttpStatus.SC\_OK, "OK");  when(httpClient.execute(Mockito.any(HttpGet.class))).thenReturn(response);   // Test the HTTP client  MyHttpClient client = new MyHttpClient(httpClient);  HttpResponse result = client.get("http://example.com");  assertEquals(HttpStatus.SC\_OK, result.getStatusLine().getStatusCode()); } |

Post Testing

|  |
| --- |
| import static org.mockito.Mockito.verify;  ...  @Test public void testPostRequest() throws IOException {  // Create a mock HTTP client  CloseableHttpClient httpClient = mock(CloseableHttpClient.class);   // Set up the mock to return a response when the POST method is called  HttpResponse response = new BasicHttpResponse(HttpVersion.HTTP\_1\_1, HttpStatus.SC\_OK, "OK");  when(httpClient.execute(Mockito.any(HttpPost.class))).thenReturn(response);   // Test the HTTP client  MyHttpClient client = new MyHttpClient(httpClient);  HttpResponse result = client.post("http://example.com", "request body");  assertEquals(HttpStatus.SC\_OK, result.getStatusLine().getStatusCode());   // Verify that the POST method was called  verify(httpClient).execute(Mockito.any(HttpPost.class)); } |

Then Return :

|  |
| --- |
| import static org.mockito.Mockito.when;  ...  @Test public void testGetRequest() throws IOException {  // Create a mock HTTP client  CloseableHttpClient httpClient = mock(CloseableHttpClient.class);   // Set up the mock to return a different response depending on the request URI  when(httpClient.execute(Mockito.argThat(req -> req.getURI().toString().equals("http://example.com")))).thenReturn(new BasicHttpResponse(HttpVersion.HTTP\_1\_1, HttpStatus.SC\_OK, "OK"));  when(httpClient.execute(Mockito.argThat(req -> req.getURI().toString().equals("http://example.org")))).thenReturn(new BasicHttpResponse(HttpVersion.HTTP\_1\_1, HttpStatus.SC\_NOT\_FOUND, "Not Found"));   // Test the HTTP client  MyHttpClient client = new MyHttpClient(httpClient);  HttpResponse result = client.get("http://example.com");  assertEquals(HttpStatus.SC\_OK, result.getStatusLine().getStatusCode());  result = client.get("http://example.org");  assertEquals(HttpStatus.SC\_NOT\_FOUND, result.getStatusLine().getStatusCode()); } |