**2. Testing the Status Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | @Test  public void givenUserDoesNotExists\_whenUserInfoIsRetrieved\_then404IsReceived()    throws ClientProtocolException, IOException {       // Given     String name = RandomStringUtils.randomAlphabetic( 8 );     HttpUriRequest request = new HttpGet( "<https://api.github.com/users/>" + name );       // When     HttpResponse httpResponse = HttpClientBuilder.create().build().execute( request );       // Then     assertThat(       httpResponse.getStatusLine().getStatusCode(),equalTo(HttpStatus.SC\_NOT\_FOUND));  } |

**3. Testing the Media Type**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | @Test  public void  givenRequestWithNoAcceptHeader\_whenRequestIsExecuted\_thenDefaultResponseContentTypeIsJson()    throws ClientProtocolException, IOException {       // Given     String jsonMimeType = "application/json";     HttpUriRequest request = new HttpGet( "<https://api.github.com/users/eugenp>" );       // When     HttpResponse response = HttpClientBuilder.create().build().execute( request );       // Then     String mimeType = ContentType.getOrDefault(response.getEntity()).getMimeType();     assertEquals( jsonMimeType, mimeType );  } |

**4. Testing the JSON Payload**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | @Test  public void    givenUserExists\_whenUserInformationIsRetrieved\_thenRetrievedResourceIsCorrect()    throws ClientProtocolException, IOException {        // Given      HttpUriRequest request = new HttpGet( "<https://api.github.com/users/eugenp>" );        // When      HttpResponse response = HttpClientBuilder.create().build().execute( request );        // Then      GitHubUser resource = RetrieveUtil.retrieveResourceFromResponse(        response, GitHubUser.class);      assertThat( "eugenp", Matchers.is( resource.getLogin() ) );  } |