Rex Jones II

Selenium How To Find Broken Links

In this session, we are going to investigate how to find a broken link on a web page using Selenium WebDriver. There are different response codes to let us know the status of a link. For example, this HTTP response status codes page has 5 classes that group the responses. HTTP is an acronym for **H**yper **T**ext **T**ransfer **P**rotocol and the response codes indicate if a particular HTTP request is successful or not successful. With our links, we prefer response codes in the range of 200. If you happen to see a code missing from this group, then that code is a non-standard response.

1. Informational responses (`100` – `199`)
2. Successful responses (`200` – `299`)
3. Redirects (`300` – `399`)
4. Client errors (`400` – `499`)
5. Server errors (`500` – `599`)

The Application Under Test (AUT) will be LinkedIn. It has a lot of links and we are going to find the broken links. Let's inspect. To find a link, we search for the anchor tag by writing //a. We see 139 matches. It's kind of misleading because the match includes anchor tags and strings. Cycle through the matches and all we need is the a tag and href attribute. href stands for **H**ypertext **Ref**erence and it creates a link to a web page. It also creates a link to a location in the same page, a link to a file or a link to an email address. We see the url begins with https.

For our code, I already have Chrome setup and LinkedIn ready to load. Plus, teardown method to quit the driver.

```
public class BrokenLinks {

  WebDriver driver;

  @BeforeClass
  public void setUp() {
    WebDriverManager.chromedriver().setup();
    driver = new ChromeDriver();
    driver.get("https://www.linkedin.com");
  }

  @AfterClass
  public void tearDown() { driver.quit(); }
```

The Test Script will have 3 main steps: First, we are going to // Find & Get All Links then // Establish A Connection To The URL finally we // Get The Response Codes & Response Messages.

```
@Test
public void findAllLinks() {
  // Find & Get All Links
  |

  // Establish A Connection To The URL

  // Get The Response Codes & Response Messages
}
```

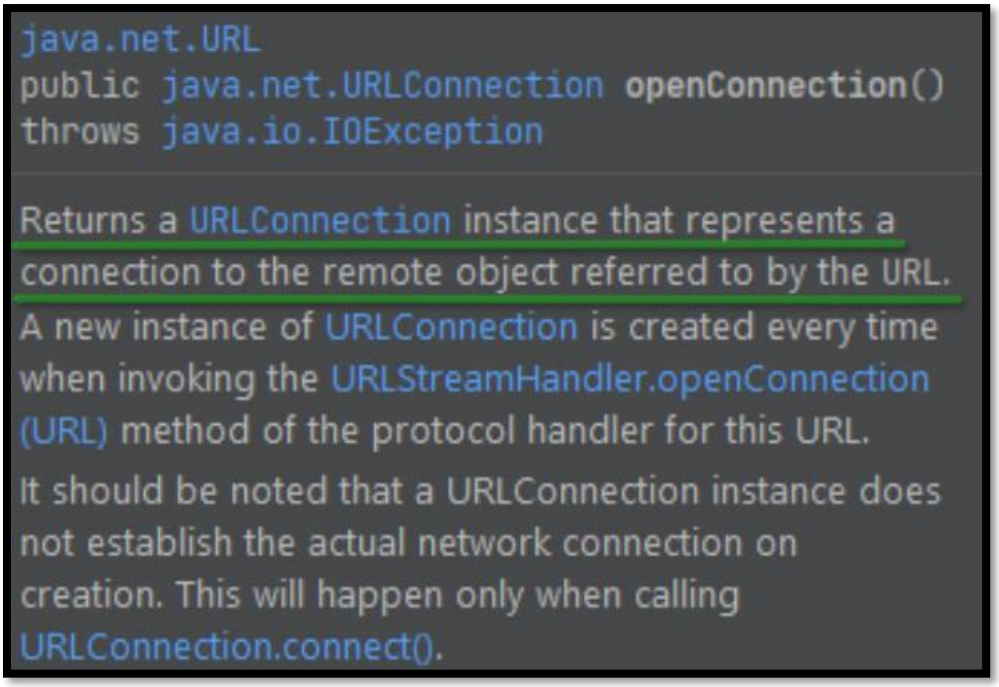
To get all links, we write driver.findElements(By.tagName(“a”)); findElements will locate all of the elements with an a tag. We assign it to a List<WebElement> with a name of allLinks. Let’s print the number of links sout(“# Links: ” + allLinks.size() + “\n”); int i = 1; This will be the counter for each url.

```
@Test
public void findAllLinks() {
  // Find & Get All Links
  List <WebElement> allLinks = driver.findElements(By.tagName("a"));
  System.out.println("# Links: " + allLinks.size());

  int i = 1;
```

At this point, we are going to iterate through every url using an enhanced for loop. It’s also called for each loop for (WebElement link : allLinks) { link.getAttribute(“href”); Recall href was the attribute so we are going to get the href then assign it to String url =. url will be the reference. if the (url != null && !url.contains(“javascript”)) { } does not equal null and does not contains javascript

```
for (WebElement link : allLinks) {
  String url = link.getAttribute( name: "href");

  if (url != null && !url.contains("javascript")) {
    |
  }
```

then next, we are going to get the establishment for the URL connection. That begins with HttpURLConnection, that's an abstract class that has different features for supporting http. connection = (HttpURLConnection) each HttpURLConnection is an instance used to represent 1 request. new URL(url).openConnection(); Add a throws declaration to the signature for url and openConnection. Open connection returns a URLConnection instance that represents a connection to the remote object referred to by the URL.



We have opened the connection but have not connected yet. We connect by writing connection.connect(); Now, our connection has been established. Next, is to get the response code and message.

```
if (url != null && !url.contains("javascript")) {
  // Establish A Connection To The URL
  HttpURLConnection connection =
          (HttpURLConnection) new URL(url).openConnection();
  connection.connect();
}
```

Next, is to get the response code and get the message. So, I'm going to take this comment and paste it in this if statement and write connection.getResponseCode(); connection.getResponseMessage();The

response code, has an int value so assign it to int responseCode =. Get response message returns a String so assign it to String responseMessage =.

Let's print both the code and message for each url. So the next line will be a print statement and it starts with sout(i + ". " + url +

"\n \t" + responseCode + "\n \t" + responseMessage); Cannot forget to Increment the counter i++; Last step is for our connection. to be disconnected.

```
// Get The Response Codes & Response Messages
int responseCode = connection.getResponseCode();
String responseMessage = connection.getResponseMessage();

System.out.println(i + ". " + url +
        "\n \t" + responseCode + "\n \t" + responseMessage);
i++;
connection.disconnect();
```

Now, let's run. There are 107 links. I see a lot of 200's response codes with an OK Message. The 200 Response Code returns the status code from each HTTP response and the OK message returns the HTTP message. I just saw a 404 Response Code and the message says Not Found for 26. I see Request denied for 35, code of 999. Let's go ahead and see what's next. Now, we are at 68. There's more 999 response codes with request denied message. That's a non-standard response.

```
26. https://www.linkedin.com/jobs/military-and-protective-services-jobs-forney-tx?trk=homepa
    404
    Not Found
27. https://www.linkedin.com/jobs/product-management-jobs-forney-tx?trk=homepage-basic_sugge
    200
    OK
28. https://www.linkedin.com/jobs/purchasing-jobs-forney-tx?trk=homepage-basic_suggested-sea
    200
    OK
```

```
35. https://www.linkedin.com/pub/dir/+/+?trk=homepage-basic_people-cta
    999
    Request denied
36. https://www.linkedin.com/learning/topics/training-and-education?trk=homepage-basic_learn
    200
    OK
```

Rex Jones II

Contact Info

- ✔ Email Rex.Jones@Test4Success.org
- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook https://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/