**Errors with solutions.**

Your code for checking broken URLs using Selenium and Java looks solid overall, but the java.net.SocketTimeoutException: Connect timed out error suggests that some URLs are either taking too long to respond or might not be reachable. Here are several steps and improvements you can make to handle this situation better:

**1. Improve Exception Handling**

Add more specific handling for SocketTimeoutException and other potential exceptions. This will help in distinguishing between different types of issues (e.g., timeouts vs. malformed URLs).

**2. Adjust Timeout Settings**

Ensure that your timeout settings are appropriate for your use case. You may want to increase the setConnectTimeout and add a setReadTimeout to handle different types of delays.

**3. Filter Out Null or Malformed URLs**

Some URLs might be null or malformed. You should add checks to handle these cases gracefully.

**4. Handle Redirections**

Some URLs might redirect to other URLs. Consider handling redirects if required.

**5. Logging and Debugging**

Enhance logging to better understand what URLs are causing issues and why.

Here’s an updated version of your verifyBrokenUrlSeqPar method with these considerations:

java

Copy code

import java.net.HttpURLConnection;

import java.net.URI;

import java.net.URL;

import java.net.SocketTimeoutException;

public class testMain {

// Existing methods ...

public static void verifyBrokenUrlSeqPar(String urlLink) {

if (urlLink == null || urlLink.isEmpty()) {

System.out.println("Skipping null or empty URL");

return;

}

try {

URI uri = new URI(urlLink);

URL url = uri.toURL();

HttpURLConnection huc = (HttpURLConnection) url.openConnection();

huc.setConnectTimeout(10000); // Increase timeout to 10 seconds

huc.setReadTimeout(10000); // Set read timeout

huc.setRequestMethod("HEAD"); // Use HEAD method to reduce data transfer

huc.connect();

int responseCode = huc.getResponseCode();

String responseMessage = huc.getResponseMessage();

if (responseCode < 400) {

System.out.println(urlLink + " --- " + responseMessage + " ----> (Good URL)");

} else {

System.out.println(urlLink + " --- " + responseMessage + " ----> (Broken URL)");

}

} catch (SocketTimeoutException e) {

System.out.println(urlLink + " --- Connection timed out ----> (Possibly Broken URL)");

} catch (MalformedURLException e) {

System.out.println(urlLink + " --- Malformed URL ----> (Skipping)");

} catch (IOException e) {

System.out.println(urlLink + " --- IO Exception: " + e.getMessage() + " ----> (Possible Network Issue)");

} catch (Exception e) {

System.out.println(urlLink + " --- Exception: " + e.getMessage() + " ----> (Error occurred)");

}

}

}

**Summary of Changes:**

1. **Timeouts:** Increased setConnectTimeout and added setReadTimeout to handle longer response times and reading delays.
2. **Error Handling:** Added specific handling for SocketTimeoutException, MalformedURLException, and IOException.
3. **Logging:** Improved logging to provide better insight into issues and potential reasons for failures.

With these changes, your script should handle different scenarios more gracefully and provide clearer insights into any problems encountered.