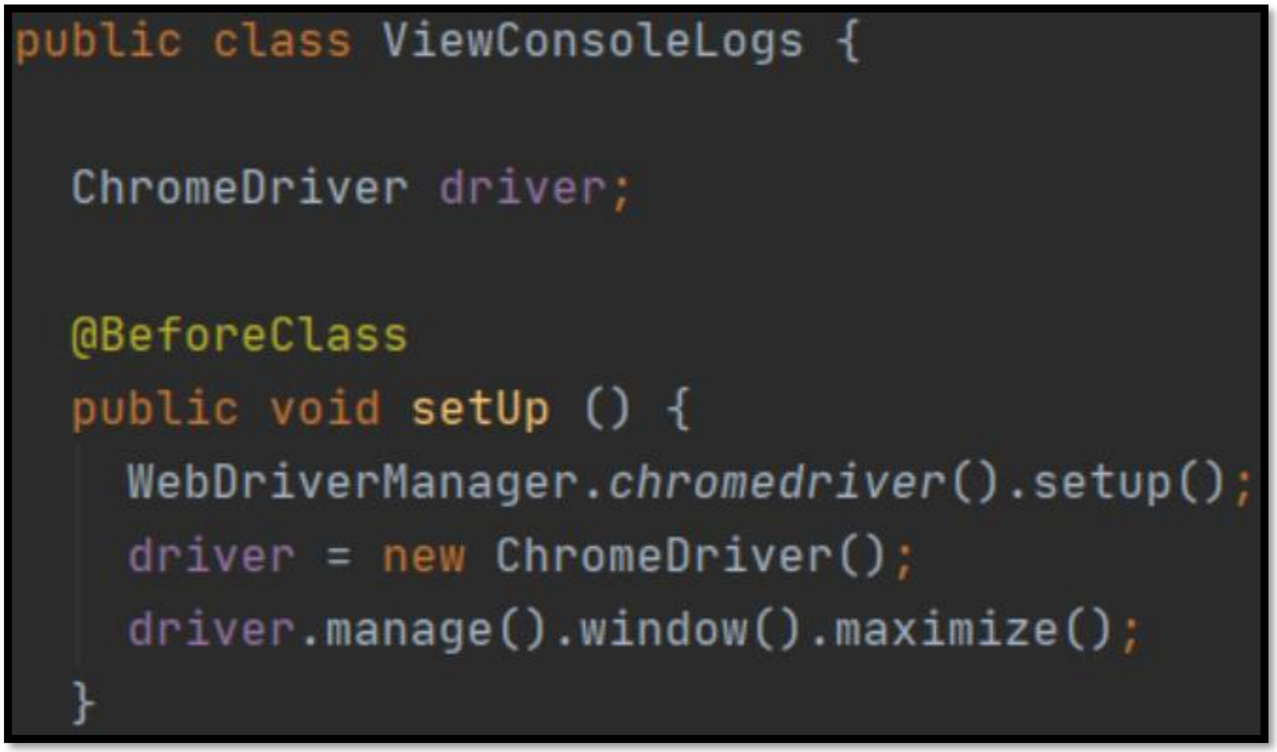
Rex
Jones II

Selenium 4: View Browser's Console Logs

There are 2 objectives for viewing a browser's Console panel. First, is to execute JavaScript. Second, is to view messages logged by the browser or logged by the developer. In this session, our focus will be to view messages logged by the browser. Most of the times, we view the Console Logs to get details about an error. With those details, we can identify the root cause of the problem.

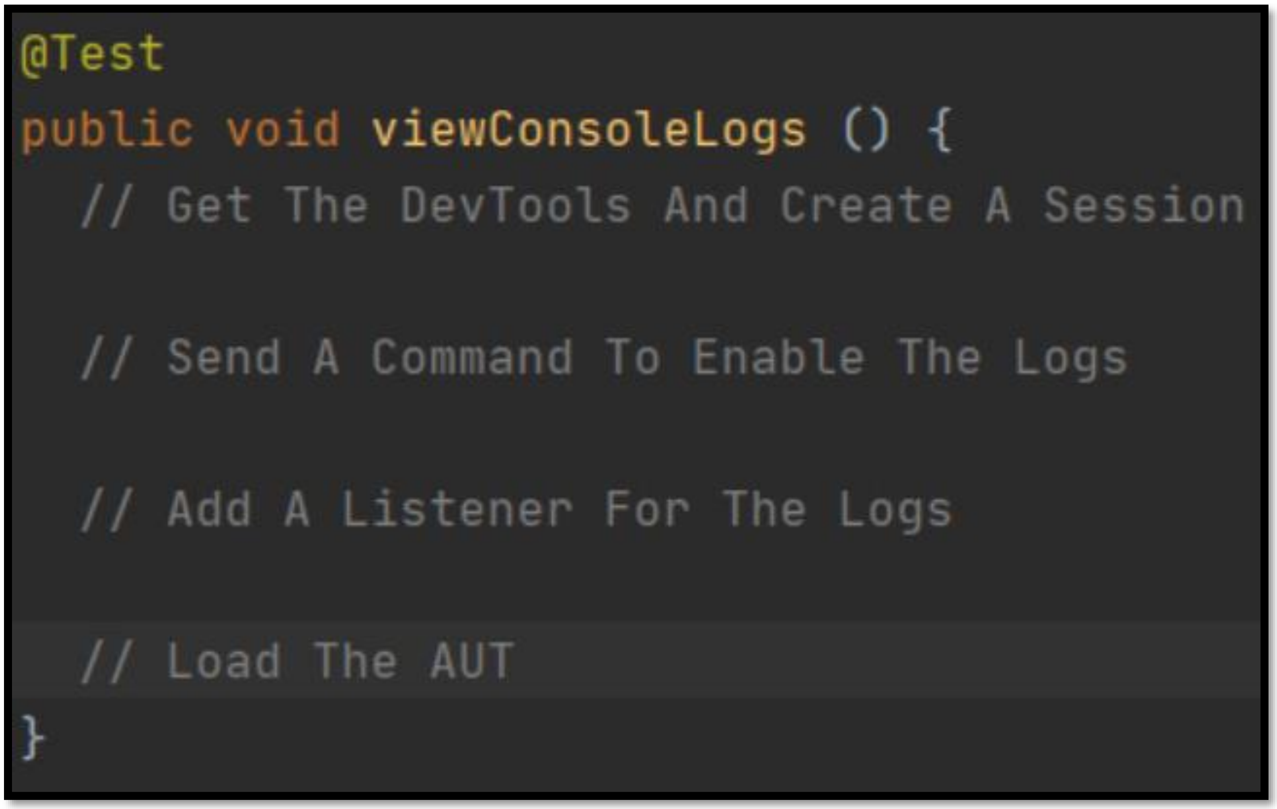
Our AUT is this lego page and CTRL + SHIFT + I is a shortcut to the Developer Tools panel. In the Console tab, we already have some logs because the page has already been loaded. We can clear the logs, reload the page, then watch the logs back show up. DevTool failed to load SourceMap and HTTP error Status Code 404. The DevTools class in Selenium 4 provides a way to listen to these logs which has 4 Log Levels: Verbose, Info, Warnings, and Errors. Verbose is not checked for default. Therefore, the messages will not be wordy. It won't have many words in the message. Info means the logs will have important information about an event. Warnings indicate a strange condition might cause a problem with operations. Errors let us know a condition is likely to cause a problem with operations.

Now for our Test Script, chromedriver is setup and the window is maximized.



```
public class ViewConsoleLogs {

  ChromeDriver driver;

  @BeforeClass
  public void setUp () {
    WebDriverManager.chromedriver().setup();
    driver = new ChromeDriver();
    driver.manage().window().maximize();
  }
```

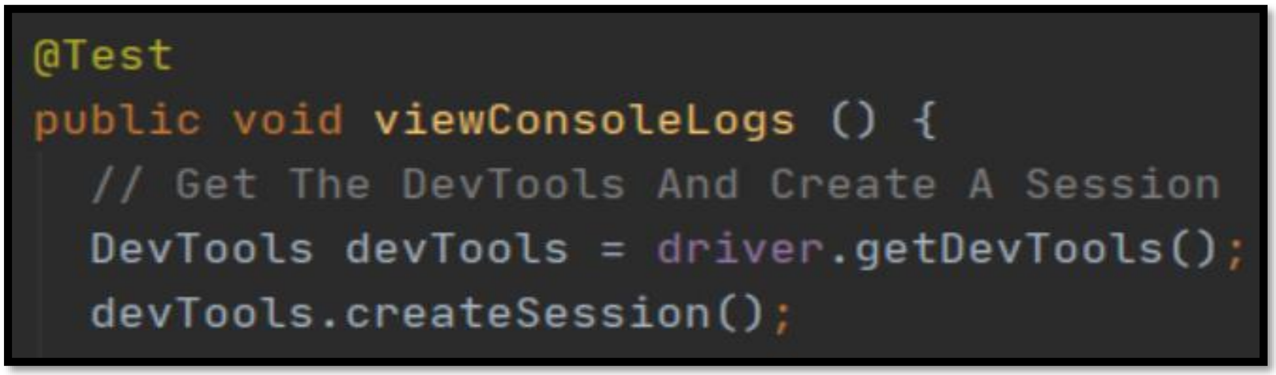
Start by writing @Test / public void viewConsoleLogs () { } Let's add some steps but write them as comments: First we // Get The DevTools and Create A Session. Then, // Send A Command To Enable The Logs. Next, we // Add A Listener For The Logs. Finally, we // Load The AUT



```
@Test
public void viewConsoleLogs () {
  // Get The DevTools And Create A Session

  // Send A Command To Enable The Logs

  // Add A Listener For The Logs

  // Load The AUT
}
```

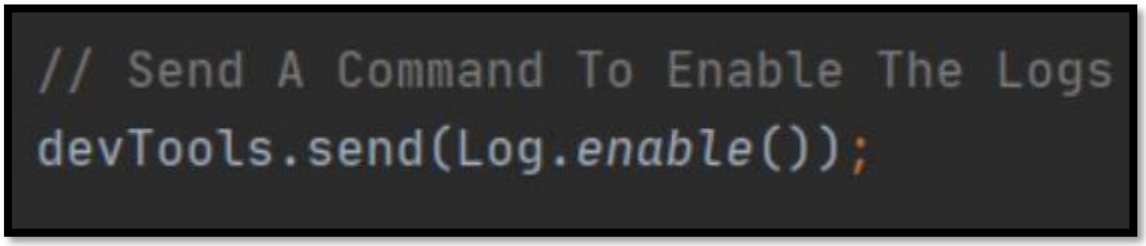
To get the dev tools, we must cast the driver if WebDriver is our type. We cannot write driver.getDevTools. Notice, getDevTools is not available. We have to cast the driver by writing ((ChromeDriver) driver).getDevTools(). If you want to directly write driver.getDevTools, we must change the type to ChromeDriver. Now, I'm going to start over and write driver.getDevTools().

Do you see how getDevTools return DevTools? That's an indicator, this statement must be assigned to DevTools with an object reference of anyname but I'm going to write devTools = . Now, we have access to all of the methods I mentioned in the [Introduction](). Before, we perform any more steps, we must take control of the Developer Tools panel in the browser by creating a session: devTools.createSession();. The purpose of this statement is to start a session in the Chrome browser.



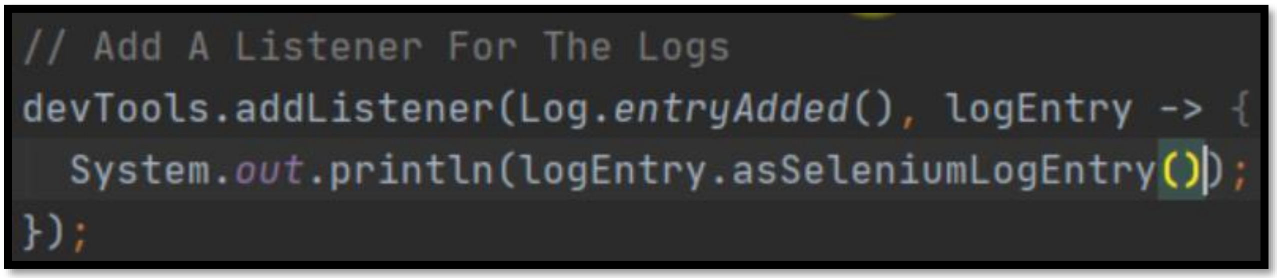
```java
@Test
public void viewConsoleLogs () {
  // Get The DevTools And Create A Session
  DevTools devTools = driver.getDevTools();
  devTools.createSession();
```

Next, is to enable logs in the Console. devTools.send(). Send is a method that has built in commands and accepts Command as a parameter. It allows us interact with the Developer Tools. Log.enable() allows us to listen to the logs but for here we see Log.enable()in the next part is one of those commands for the send() method. Log serves as a representation or model for a CDP domain.



```java
// Send A Command To Enable The Logs
devTools.send(Log.enable());
```

Now that we have the logs enabled, this is the part we go and listen for logs by writing devTools.addListener(). Once again, we write Log. but this time it's an event for entryAdded(), logEntry -> { }. To use this Lambda expression (->), you must have Java 1.8 or higher.

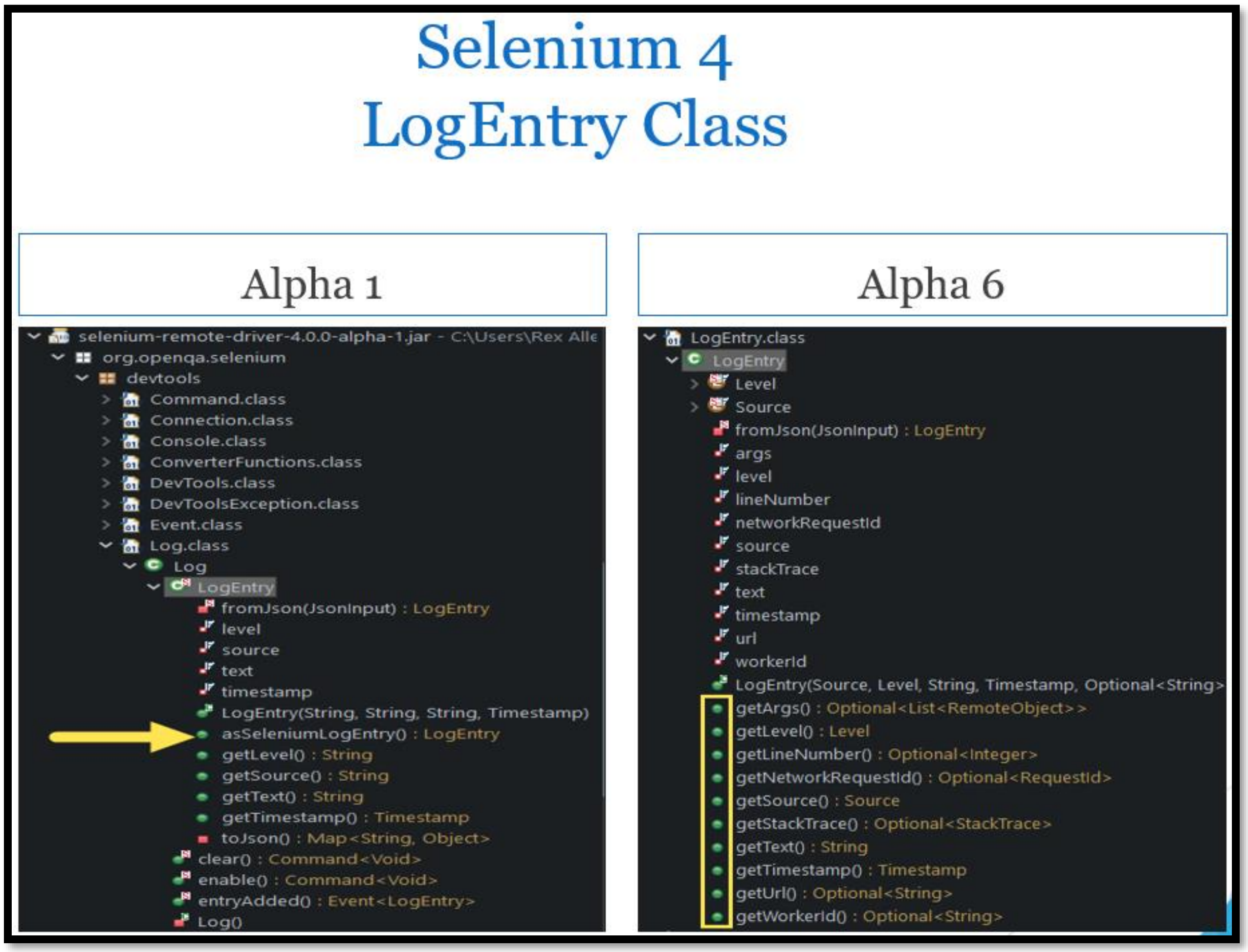


```java
// Add A Listener For The Logs
devTools.addListener(Log.entryAdded(), logEntry -> {
  System.out.println(logEntry.asSeleniumLogEntry());
});
```

Let's print the log entries sout(logEntry.asSeleniumLogEntry()); This is where I noticed a difference between Selenium 4 Alpha 1 and the current version. I changed my pom.xml file to use the version for Alpha . Alpha 1 has asSeleniumLogEntry.

In this screenshot, we see Alpha 6 does not have asSeleniumLogEntry for a method. Alpha 6 is not the most recent version. However, Alpha 6 and the most recent version added more methods although asSeleniumLogEntry is missing. Alpha 1 does not have methods getArgs, getLineNumber, getNetworkRequestId, getStackTrace, getUrl and getWorkerId.

Rex Jones II



Selenium 4 LogEntry Class

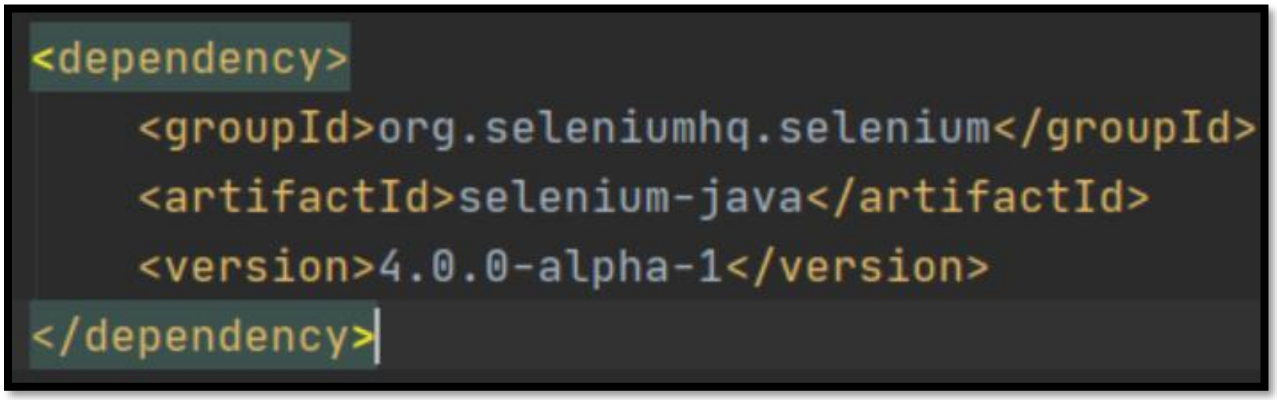
Alpha 1

```
selenium-remote-driver-4.0.0-alpha-1.jar - C:\Users\Rex Alle
  org.openqa.selenium
    devtools
      Command.class
      Connection.class
      Console.class
      ConverterFunctions.class
      DevTools.class
      DevToolsException.class
      Event.class
      Log.class
        Log
          LogEntry
            fromJson(JsonInput) : LogEntry
            level
            source
            text
            timestamp
            LogEntry(String, String, String, Timestamp)
            asSeleniumLogEntry() : LogEntry
            getLevel() : String
            getSource() : String
            getText() : String
            getTimestamp() : Timestamp
            toJson() : Map<String, Object>
          clear() : Command<Void>
          enable() : Command<Void>
          entryAdded() : Event<LogEntry>
          Log()
```

Alpha 6

```
LogEntry.class
  LogEntry
    Level
    Source
    fromJson(JsonInput) : LogEntry
    args
    level
    lineNumber
    networkRequestId
    source
    stackTrace
    text
    timestamp
    url
    workerId
    LogEntry(Source, Level, String, Timestamp, Optional<String>
    getArgs() : Optional<List<RemoteObject>>
    getLevel() : Level
    getLineNumber() : Optional<Integer>
    getNetworkRequestId() : Optional<RequestId>
    getSource() : Source
    getStackTrace() : Optional<StackTrace>
    getText() : String
    getTimestamp() : Timestamp
    getUrl() : Optional<String>
    getWorkerId() : Optional<String>
```

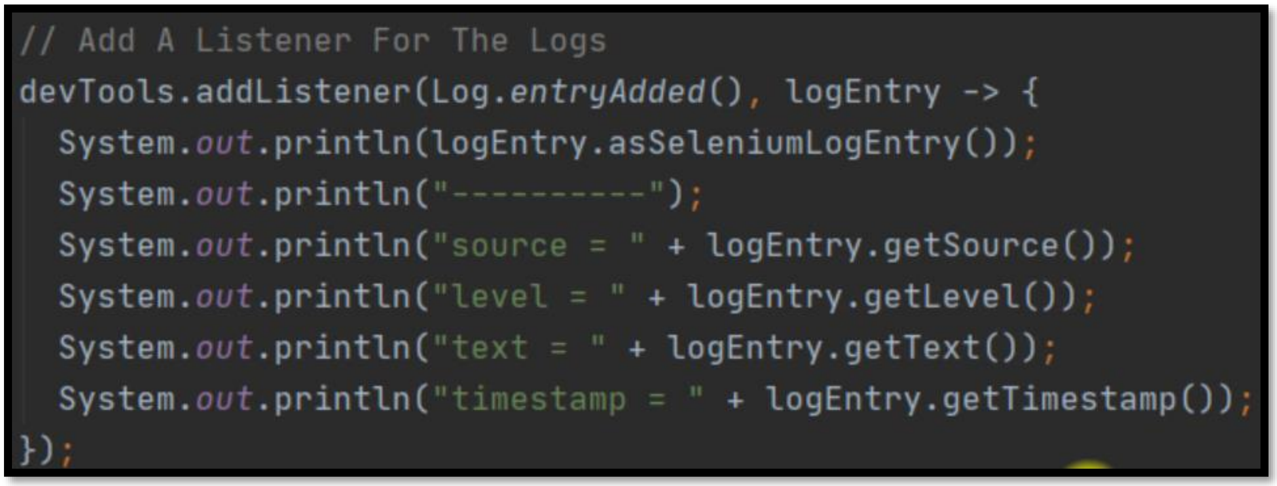
Here's my pom.xml file that shows Alpha 1.



```xml
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.0.0-alpha-1</version>
</dependency>
```

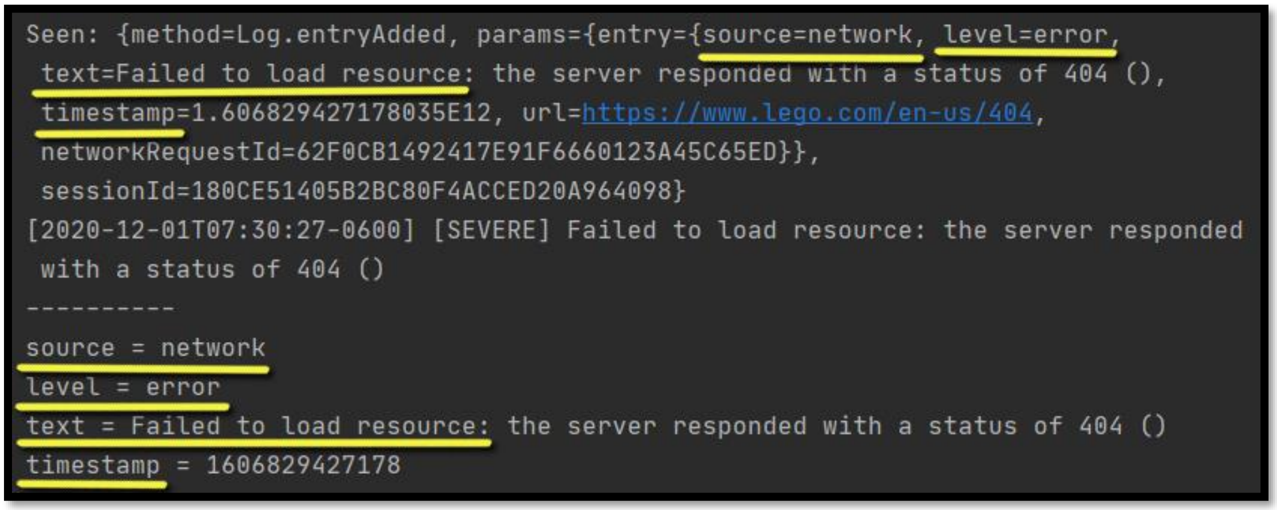
I'm going to add some hyphens then print the next Log Entries. sout("----------");
sout("source = " + logEntry.getSource());

Copy (“source = ” + logEntry.getSource()) then get the level getLevel() and change source to level Change level to text and getLevel to getText then level to timestamp and getLevel to getTimestamp.



```
// Add A Listener For The Logs
devTools.addListener(Log.entryAdded(), logEntry -> {
  System.out.println(logEntry.asSeleniumLogEntry());
  System.out.println("----------");
  System.out.println("source = " + logEntry.getSource());
  System.out.println("level = " + logEntry.getLevel());
  System.out.println("text = " + logEntry.getText());
  System.out.println("timestamp = " + logEntry.getTimestamp());
});
```

Last but not least we are going to load the AUT to view the Console logs is driver.get(); The URL is https://www.lego.com/404. Let’s Run. In the Console, we see some of the same information because that method asSeleniumLogEntry combines different log entries and we also printed individual log entries. For example, it has source as network, level = error, text = Failed to load resource and the timestamp.



```
Seen: {method=Log.entryAdded, params={entry={source=network, level=error,
 text=Failed to load resource: the server responded with a status of 404 (),
 timestamp=1.606829427178035E12, url=https://www.lego.com/en-us/404,
 networkRequestId=62F0CB1492417E91F6660123A45C65ED}},
 sessionId=180CE51405B2BC80F4ACCED20A964098}
[2020-12-01T07:30:27-0600] [SEVERE] Failed to load resource: the server responded
 with a status of 404 ()
----------
source = network
level = error
text = Failed to load resource: the server responded with a status of 404 ()
timestamp = 1606829427178
```

That’s it for capturing logs using Selenium 4 CDP and I’ll see you in the next session. If you are interested in more videos, feel free to subscribe to my YouTube channel and click the bell icon. Also, follow me on Twitter, connect with me on LinkedIn. The transcript and code will get placed on GitHub.

Rex Jones II

Contact

- ✓ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✓ Facebook http://facebook.com/JonesRexII
- ✓ Twitter https://twitter.com/RexJonesII
- ✓ GitHub https://github.com/RexJonesII/Free-Videos
- ✓ LinkedIn https://www.linkedin.com/in/rexjones34/