

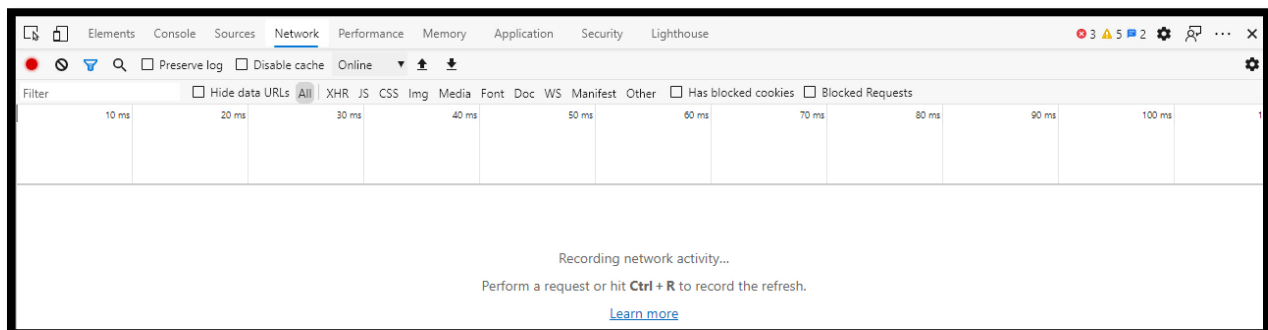
Selenium 4

How To Automate Slow Internet Connection

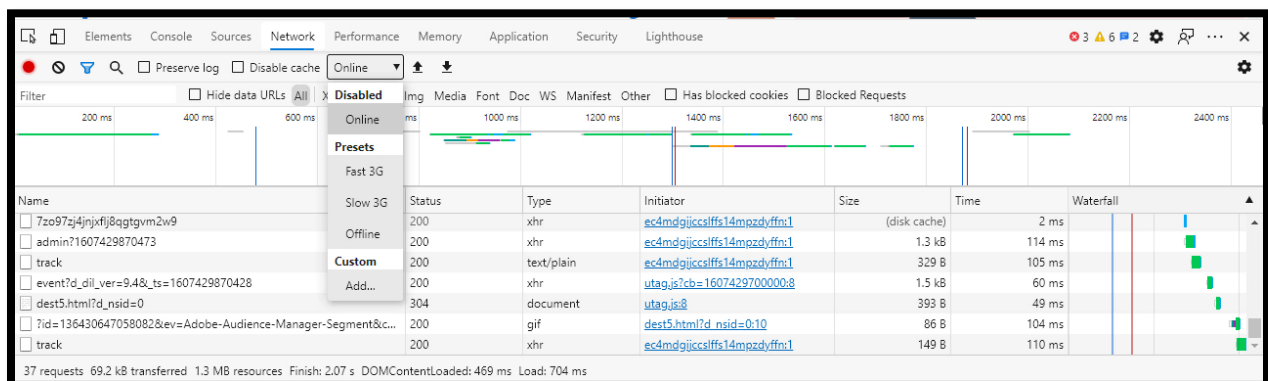
The Network panel is another way to debug a problem. It has logs to help us troubleshoot the problem. We mostly use the panel for 2 reasons. One reason is to inspect the network properties of a resource. Another reason is to make sure our resources are downloaded and uploaded as expected. When testing an application, it's easy to forget or not think about a user with a weak connection. Therefore, we are going to automate how to slow down our internet connection.

Slow Connection

In the Network panel, there's no available request. It's empty because we must open the panel before performing an action. That's how we preserve the network traffic data.



Refresh the page and all of the network activity shows up in the Network Log. Each row in the Network Log represents a resource. Do you see 69.2 kB transferred? That's the total download size. This Network Throttle dropdown is how we emulate different connection speeds. We see Fast 3G, Slow 3G, and Offline.



In this session, I will automate a 3G connection but next session I will take the network offline. Select any preset and the DevTools display a warning icon beside the Network tab.

For our Test Script, I have the setup method for EdgeDriver. Let me also add driver.getDevTools() assign it to devTools = . Also write DevTools devTools; up top.

```
public class EnableNetworks {  
  
    EdgeDriver driver;  
    DevTools devTools;  
  
    @BeforeMethod  
    public void setUp () {  
        WebDriverManager.edgedriver().setup();  
        driver = new EdgeDriver();  
        driver.manage().window().maximize();  
        devTools = driver.getDevTools();  
    }  
}
```

Enable the network to slow down by writing. `@Test public void enableSlowNetwork () { }` Start by creating a session `devTools.createSession();` then we send a command to enable the network by writing `devTools.send(Network.enable(Optional.empty(), let's write this statement 2 more times. This complete statement makes it possible to deliver network tracking and events to the client. All of the parameters are Optional.empty because they are not required. In fact, the first 2 parameters are experiments.`

Next, is to emulate the Network Conditions. We can go to [github](#) for the method and parameters. Search for emulate. At the bottom is `Network.emulateNetworkConditions`. It activates emulation of network condition.

Emulation.canEmulate

Tells whether emulation is supported.

Emulation.setEmulatedMedia

Emulates the given media type or media feature for CSS media queries.

Emulation.setEmulatedVisionDeficiency

Emulates the given vision deficiency.

Input.emulateTouchFromMouseEvent

Emulates touch event from the mouse event parameters.

Network.canEmulateNetworkConditions

Tells whether emulation of network conditions is supported.

Network.emulateNetworkConditions

Activates emulation of network conditions.

The parameters are offline, latency, downloadThroughput, uploadThroughput, and connectionType. Connection Type is the only optional parameter.

Network.emulateNetworkConditions

Activates emulation of network conditions.

PARAMETERS

offline	boolean	True to emulate internet disconnection.
latency	number	Minimum latency from request sent to response headers received (ms).
downloadThroughput	number	Maximal aggregated download throughput (bytes/sec). -1 disables download throttling.
uploadThroughput	number	Maximal aggregated upload throughput (bytes/sec). -1 disables upload throttling.
connectionType	ConnectionType	Connection type if known.
optional		

The method we wrote for enabling the network is right here and we see all 3 parameters are optional: maxTotalBufferSize, maxResourceBufferSize, and maxPostDataSize. If I hover Experimental, the tool tip says "This may be changed, moved, or removed".

Network.enable #

Enables network tracking, network events will now be delivered to the client.

PARAMETERS

maxTotalBufferSize optional	integer Buffer size in bytes to use when preserving network payloads (XHRs, etc). EXPERIMENTAL
maxResourceBufferSize optional	integer Per-resource buffer size in bytes to use when preserving network payloads (XHRs, etc). EXPERIMENTAL <small>This may be changed, moved or removed</small>
maxPostDataSize optional	integer Longest post body size (in bytes) that would be included in requestWillBeSent notification

For our Test Script, let's complete it by writing `devTools.send(Network.emulateNetworkConditions())`. We want the network to stay online so we set `offline` to `false`, `latency` is 150, `downloadThroughput` is 2500, `uploadThroughput` is 2000. Next, is the Connection Type. `Optional.of(ConnectionType.)` We see different choices. We have `BLUETOOTH`, `2G`, `3G`, `4G`, and `WIFI`. Let's go ahead and select `3G`.

Let's also compare the time between this slow network and the normal way by loading an application. Know what before we compare, we must load the application for LinkedIn:
`driver.get("https://www.linkedin.com")`. Let's also print the page title. `sout("Slow " + driver.getTitle())`.

```
@Test
public void enableSlowNetwork () {
    devTools.createSession();
    devTools.send(Network.enable(
        Optional.empty(),
        Optional.empty(),
        Optional.empty()));
    devTools.send(Network.emulateNetworkConditions(
        offline: false,
        latency: 150,
        downloadThroughput: 2500,
        uploadThroughput: 2000,
        Optional.of(ConnectionType.CELLULAR3G)));
    driver.get("https://www.linkedin.com");
    System.out.println("Slow " + driver.getTitle());
}
```

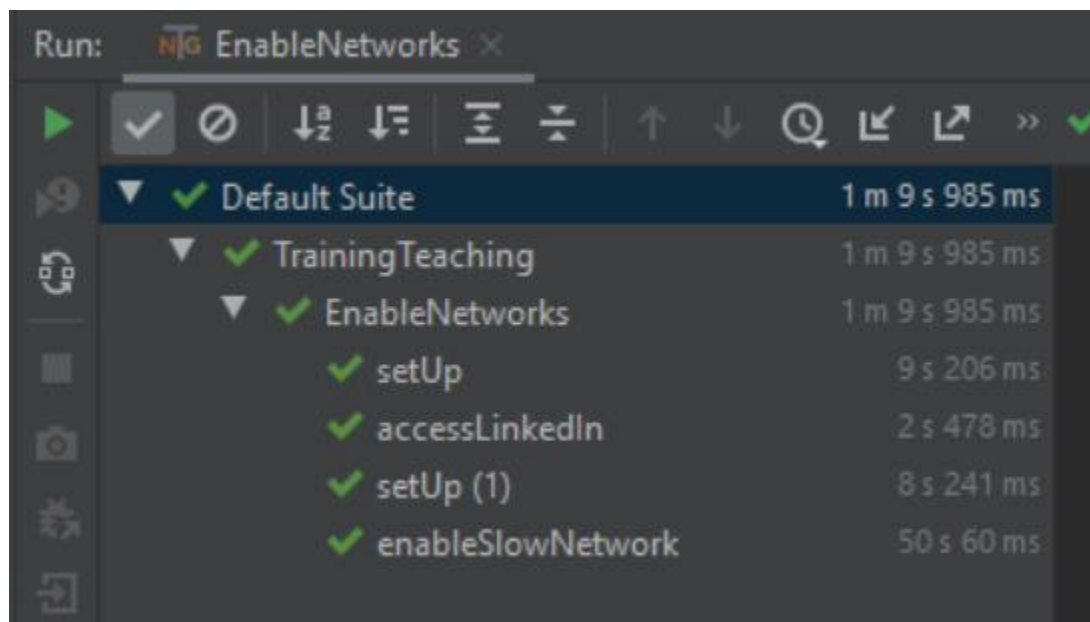
Now, let's go ahead and load LinkedIn the normal way.

```
@Test public void accessLinkedIn () { } driver.get("https://www.linkedin.com"); Also sout("Access " + driver.getTitle());
```

```
@Test
public void accessLinkedIn () {
    driver.get("https://www.linkedin.com");
    System.out.println("Access " + driver.getTitle());
}
```

Let's go ahead and run.

The Console shows accessLinkedIn loaded in 2 Seconds and 478 Milliseconds. enableSlowNetwork took longer to complete. It finished in 50 Seconds and 60 Milliseconds. Both Test Scripts show the Page Title. That's it for slowing down the Network to a slower connection.



The screenshot shows the Selenium IDE Run console with the following test results:

Test Case	Duration
Default Suite	1 m 9 s 985 ms
TrainingTeaching	1 m 9 s 985 ms
EnableNetworks	1 m 9 s 985 ms
setUp	9 s 206 ms
accessLinkedIn	2 s 478 ms
setUp (1)	8 s 241 ms
enableSlowNetwork	50 s 60 ms

Next, I'm going to show you how to take the Network offline. If you are interested in more videos, consider subscribing to my [YouTube](#) channel and clicking the bell icon. Also, follow me on [Twitter](#), connect with me on [LinkedIn](#) and [Facebook](#). The transcript and code will be placed on [GitHub](#). Thanks for watching and I'll see you in the next session.

Thank you

Follow: <https://www.linkedin.com/in/bhavin-thumar/>