Selenium ChromeDev Tools Integration –

What are ChromeDevTools:

Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser

With Chrome DevTools, developers have deeper access to the Applications which render on browser

What is Chrome DevTools Protocol (CDP)?

The Chrome DevTools Protocol provide tools to instrument, inspect, debug and profile chromium, chrome and other Blink-based browsers.

Selenium 4 introduces powerful commands which are wrapper around the CDP Domains to grants access to Chrome DevTools directly from your automated tests.

With this CDP Support Selenium opens up the possibilities of out of Box Testing with the complete access and control to the browser features within the Test.

Examples :

. Capture, Monitor and Stub the Network requests and response

. Inject session cookies and peform basic Auth

. Mock Device Coordinates for Mobile/Tabs view

. Check and Monitor the site’s performance

. Mock faster/slower networks speeds

. Block the Network requests

. Execute and debug javascript

. view console logs

Code Snippet to Initialize ChromeDevTools connection with selenium

Chrome Driver class has predefined methods to access the DevTools.

Chrome Driver and Edge Driver are inherited from Chromium Driver. So, we can access to Devtools with Chrome and Edge browsers

Step 1:

Initiate Chromium Driver.

Step 2:

Create the object for Chrome Dev tools with getDevTools() method

DevTools devTools = driver.getDevTools()

Return the new DevTools object which allows you to send() the build-in Selenium commands for CDP.

Step 3:

Initiate DevTools Session to send the commands from Selenium - devTools.createSession()

Step 4 :

devTools.send(COMMAND)

what is this command ? – These selenium inBuilt commands are wrapper methods that invoke CDP Domain functions.

Syntax of Command – {Domain}.{command}

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Selenium has built-in commands for most of the commonly used CDP methods.

For Unimplemented built-in commands, you can call CDP methods directly from selenium code with below step.

Driver.executeCdpCommand(Methodname,parameters)

two steps which are common here.

Initializing chrome driver is common and you should also initialize get dev tools

first step whenever you want to interact with chrome dev tools is to create dev tools object.

And then initiate session so that you can communicate with the dev tools. With create session stream.

218. What are Chrome Dev tools? Why do we need this for Selenium testing?

how to integrate Selenium with Chrome DevTools.

This is one of the feature introduced in the Selenium from the version 4.

Open chrome browser right click and inspect, this is nothing but Chrome developer tools.

Here u can see all the  HTML elements of your web application.

You can also see the console logs if there are any on your application

Network - If you hit any service API calls on your application, you can see what is the request,

what are responses, headers,everything you can track in the network tab.

and you can get the metrics from performance tab.

people use this a lot to debug their application and see how that is behaving.

So here you have all the Chrome DevTools information, but how to access them.

whereever have developed this have provided one APIs on the name of Chrome DevTools protocol.

Google – chorme devtools protocols - these guys exposed many APIs from which you will be able to access these level elements.

select network domain. so this network domain have many methods.

So basically this protocol exposes all the methods, events for you to perform and manipulate any actions on this Chrome DevTools, So this is nothing to do with Selenium.

whoever have invented these developer tools same team have also introduced an APIs

to access those dev tools programmatically with the help of methods.

So for a front end user you will just see the table data, but if you open the developer tools and go

to network tab, it'll clearly tell you what is the API call made when that guy clicked on that button.

CDP will help you to access the information from Chrome DevTools

CDP exposes all these methods, So what Selenium guys did, on this method, they wrote a wrapper. So when you have a command called clear browser cache and you are Selenium site as well and when you select that command,that command internally again calling this CDP method.

CDP method will go and call and access all the information in that developer tool,

So like that first you will initiate Selenium command and Selenium command calls these CDP methods and these CDP methods are acting upon Chrome DevTools

and giving the information what we need.

What is the benefit I get when I stop the network request from Selenium ?

 inject session cookies also. You need not log in with username and password. If you have an active session cookie, you can inject that cookie in your browser, and you can bypass log in and directly start into your portal

mock the device coordinates- make your browser look like a mobile device

for an Appium tool to test their browsers in mobile views.But using Selenium DevTools,now you can directly test using Selenium only. by just mocking your device coordinates

and invoking the browser in mobile views

By selecting this toggle device toolbar. Whatever future you have in this area, everything will get available for you through Selenium via Chrome DevTools protocol.

mock your network speeds. You develop your automation test, it passes. But when it runs through jenkin build, it fails because of sync issues, weight issues or application is not loading.

219. Understand Device metrics override function to simulate browser as mobile

So let's quickly create one new project in our eclipse and bring all that integration into live in form of coding.

Right click->new->java project(SeleniumFeaturesLatest) Next and click on finish.

convert this java project into Maven so that we can directly get the Selenium dependency into this java project.

Right click on the project -> configure->convert to maven project ->Finish

Group id : rahulshettyacademy artifactid: SeleniumFeaturesLates

So when you convert Java project directly to Maven, by default you won't see this compiler plugin. So you can just Google for Maven compiler plugin of, and you will get it from any website.

And here you have to give which version of Java you want to use in pom.xml.

Mvn repo – selenium java – copy & paste the depe in pom.xml

what are the browsers run on the chromium engine, do you know? -  Google Chrome and Microsoft Edge.

 in the latest version, the chromium based browsers like Chrome driver and edge driver are inherited not from remote. They are now pointing to Chromium driver.

So with these two browsers only you will be able to integrate with Chrome dev tools.

 web driver interface do not expose dev tools. So you have to explicitly use Chrome driver only.

Devicescalefactor -zoom in/zoom out.

So that's how you can emulate and make your device looks like iPhone by providing the screen width, height, everything as a parameters to your set device metrics override method.

selenium guys did their best in providing custom commands for all the methods, domains,

what you see here. But there are cases where you may not find custom selenium command for the respective CDP method.

what if this custom method is absent? In that case, you should write and create

your own custom command to directly call that CDP method what you see here,

import java.util.Optional;

import org.openqa.selenium.By;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.devtools.DevTools;

import org.openqa.selenium.devtools.v93.emulation.Emulation;

public class MobileEmulatorTest {

public static void main(String[] args) throws InterruptedException {

System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = new ChromeDriver();

DevTools devTools = driver.getDevTools();

devTools.createSession();

//send command to CDP Methods-> CDP Methods will invoke and get access to chrome dev tools

devTools.send(Emulation.setDeviceMetricsOverride(600, 1000, 50, true, Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty(), Optional.empty()));

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.cssSelector(".navbar-toggler")).click();

Thread.sleep(3000);

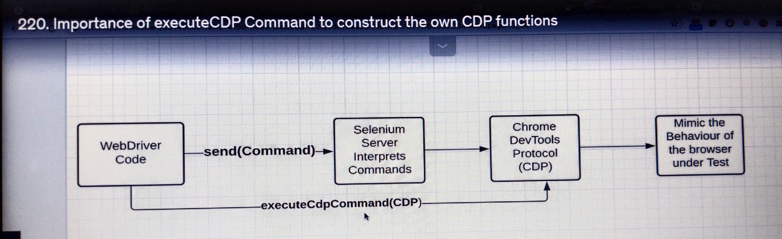
driver.findElement(By.linkText("Library")).click();

driver.close();

//Network.getRequestPostData

}

220. Importance of executeCDP Command to construct the own CDP



One way - And if you see this diagram, if you provide Selenium custom command, and if you run your test, Selenium server first interprets that commands and eventually that command will have a step to invoke CDP method, And CDP method finally will mimic the behavior of the browser which is under test.

what if that custom command is not present or not developed by selenium?

you can directly invoke CDP method from your Webdriver code. By calling this method, execute CDP command. And inside this you have to pass the CDP method.

so this way you can bypass calling the custom command of Selenium.

Earlier, we directly used set device matrix override method, right?

So how to achieve same thing without using a command from the selenium.

Created one new class (CdpCommandsTest)

first you have to again create session for dev tools ( DevTools devTools = driver.getDevTools() )

Once you have the object of dev tools, then you can go ahead and initiate the session.

So earlier you just called this method and you sent arguments of that method into that method as an parameters. But when you want to run directly the CDP command, it takes two arguments.

So this way you have directly called CDP method without bringing selenium custom command into picture.

import java.util.HashMap;

import java.util.Map;

import org.openqa.selenium.By;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.devtools.DevTools;

public class CdpCommandsTest {

public static void main(String[] args) throws InterruptedException {

System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = new ChromeDriver();

DevTools devTools = driver.getDevTools();

devTools.createSession();

Map<String, Object> deviceMetrics = new HashMap<String, Object>();

deviceMetrics.put("width",600);

deviceMetrics.put("height",1000);

deviceMetrics.put("deviceScaleFactor",50);

deviceMetrics.put("mobile",true);

driver.executeCdpCommand("Emulation.setDeviceMetricsOverride", deviceMetrics);

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.cssSelector(".navbar-toggler")).click();

Thread.sleep(3000);

driver.findElement(By.linkText("Library")).click();

}

}

221. Localization Testing with Selenium 4 using ChromeDevTools Protocols

Emulation – set geolocatin override

So when you browse in your browser on any webpage, so based upon your current location

your search results will be shown and based upon your current location your language also will be changed.

So if you browse it from United States or India, you will see it in English, but when you browse it from Europe countries, you will see in their respective languages like Spanish, French,

So if you want to test whether if you web application is changing the language

based upon the country where they're logging in, we generally call this concept as localization testing

 I want to test whether this is showing up properly in all the Spanish language, I have to log into the server from the Spanish country servers and validate the text changes. But now you can mimic the browser behavior of Chrome and you can actually test that from your local system only.  for automation testing there is no way that you can mimic the behavior earlier,

but with Selenium Chrome dev tools integration, yes, it is possible.

Setgeolocationoverride method will help you to change the location  provide latitude and longitude of that location where you are. And based upon that parameters,

it'll tell which country you are in and based upon that country experience you will see in the browser, Ex: If you're in Spain, you will see Spanish

there is no way you can do it in Selenium, testing it in different languages,

but let's see how to do that with Chrome dev tools integration ?

google-go to the Google Earth app and search any place and get that latitude and longitudinal location.

You can still test your functionality if it is behaving and switching to different languages

based upon your search

**import** java.util.HashMap;

**import** java.util.Map;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.devtools.DevTools;

**public** **class** SetGeoLocation {

**public** **static** **void** main(String[] args) {

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

DevTools devTools = driver.getDevTools();

devTools.createSession();

Map<String,Object>coordinates = **new** HashMap<String,Object>();

//40 3

coordinates.put("latitude", 40);

coordinates.put("longitude", 3);

coordinates.put("accuracy", 1);

driver.executeCdpCommand("Emulation.setGeolocationOverride", coordinates);

driver.get("http://google.com");

driver.findElement(By.*name*("q")).sendKeys("netflix",Keys.***ENTER***);

driver.findElements(By.*cssSelector*(".LC20lb")).get(0).click();

String title =driver.findElement(By.*cssSelector*(".our-story-card-title")).getText();

System.***out***.println(title);

}

222. How to extract Network Responses and status codes with Selenium CDP listeners

how to log network activity on your web application ?

so you should be able to log all API requests, responses, what all activities happened on your web page.

let's say tomorrow your test is failed, you can simply open the log file and see which API request is failed. Based upon that, you can understand why the UI rendering is failing.

So if you see something wrong in UI, most of the cases it would be based upon some failing responseon API side.

If it is not giving the data properly or if something goes wrong from back end, front end will mess up.

That is, most of the times that was the reason for the functional errors.

So if you can log that response code response body in your file from selenium, then instead of replicating the issue or instead of saying simply it has a flaky, you can open the log file and see what exactly happened and why the test is failed by reading your network logs.

how you can log all that information from Selenium, Chrome Dev Tools integration?

public class NetworkLogActivity {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = new ChromeDriver();

//log file ->

DevTools devTools = driver.getDevTools();

devTools.createSession();

devTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));

devTools.addListener(Network.requestWillBeSent(), request -> // lambdaa expression (->)

{

Request req = request.getRequest();

System.out.println(req.getUrl());

//req.getHeaders()

});

//Event will get fired-

devTools.addListener(Network.responseReceived(), response ->

{

Response res = response.getResponse();

System.out.println(res.getUrl());

System.out.println(res.getStatus());

if(res.getStatus().toString().startsWith("4"))

{

System.out.println(res.getUrl()+"is failing with status code"+res.getStatus());

}

});

driver.get("https://rahulshettyacademy.com/angularAppdemo");

driver.findElement(By.cssSelector("button[routerlink\*='library']")).click();

223. Intercept Network /API Responses with Selenium Chrome dev tools

 how to mock network requests with selenium on Chrome Dev Tools integration ?

 when you click on virtual library, you are getting a lot of books, So we are using this account to validate many cases on this page.

here you should also test oops, only one book is available. Text that comes only when there is one row. So what you do you would get again another account and do another test.

It's not required in the same test. So there are multiple records displaying here.

You can still test that message by mocking your response or by mocking your request to make sure that you get only one row here at run time. You can do that and make this page look like having only one browser. So that's the beauty of mocking your network request.

two steps which are common here.

Initializing chrome driver is common and you should also initialize get dev tools

first step whenever you want to interact with chrome dev tools is to create dev tools object.

And then initiate session so that you can communicate with the dev tools. With create session stream.

network domain - it will allow you to track network activities. You can get the real request, real response, real headers, anything.

But if you want to mock or if you want to intercept your existing calls, then there is another domain which is fetch.

when you want to do network log activity, first thing, what you did is enable the network so that from Selenium you can listen to the network activity.

If you want to execute a method you need to use send ( devtools.send())

**import** java.util.Optional;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.devtools.DevTools;

**import** org.openqa.selenium.devtools.v120.fetch.Fetch;

**public** **class** NetworkMocking {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

DevTools devTools = driver.getDevTools();

devTools.createSession();

devTools.send(Fetch.*enable*(Optional.*empty*(), Optional.*empty*()));

devTools.addListener(Fetch.*requestPaused*(), request->

{

**if**(request.getRequest().getUrl().contains("shetty"))

{

String mockedUrl =request.getRequest().getUrl().replace("=shetty", "=BadGuy");

System.***out***.println(mockedUrl);

devTools.send(Fetch.*continueRequest*(request.getRequestId(), Optional.*of*(mockedUrl), Optional.*of*(request.getRequest().getMethod()),Optional.*empty*(), Optional.*empty*(),Optional.*empty*()));

}

**else** {

devTools.send(Fetch.*continueRequest*(request.getRequestId(),Optional.*of*(request.getRequest().getUrl()), Optional.*of*(request.getRequest().getMethod()),Optional.*empty*(), Optional.*empty*(),Optional.*empty*()));

}

});

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.*cssSelector*("button[routerlink\*='library']")).click();

Thread.*sleep*(3000);

System.***out***.println(driver.findElement(By.*cssSelector*("p")).getText());

}

}

224. How to test failed Network request calls with Selenium CDP Commands

sometimes when the request calls are failed, you should display some

custom message on your screen.

somethingtext like that or oops, our servers are having real problem.

To get the data, please try again.So there are some custom messages you want to give in your web page.

For that verification, we do network fail.

So request passed is the listener comes into picture whenever there is.

If there is any request that's going to server instead of going to server, we are stopping that first and making some modifications and then sending back to server either with continue request. So that means just continue sending back to the request or fail that request.

"Fetch.enable." this method takes two parameters. One is "patterns" and one is "handleAuthRequests,"

So what is patterns? So instead of monitoring each and every request, you can tell that which kind of request you should monitor, So when you are automating an application, there could be 10s or 20s of REST API calls made to the server, Instead of monitoring each and every call,

you can tell a kind of pattern what you are looking for, In our Case in the application what we are testing, we have different API calls, HTTP calls, basically. So I want to pause and fail only this

"AuthorName =" to whatever we get. So, you can give the pattern saying "GetBook." So that will only stop this request. So whatever actions you want to perform, it will perform those actions

only on that particular HTTP call which match with our pattern.

GetBook regular expression, because \*\* represents it could be anything before and after.

If it matches with that pattern, then we are only tracking that kind of HTTP calls

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.Optional;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.devtools.DevTools;

**import** org.openqa.selenium.devtools.v120.fetch.Fetch;

**import** org.openqa.selenium.devtools.v120.fetch.model.RequestPattern;

**import** org.openqa.selenium.devtools.v120.network.model.ErrorReason;

**public** **class** NetworkFailedRequest {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

//log file ->

DevTools devTools = driver.getDevTools();

devTools.createSession();

//java.util.Optional<java.lang.String> urlPattern

Optional<List<RequestPattern>> patterns = Optional.*of*(Arrays.*asList*(**new** RequestPattern(Optional.*of*("\*GetBook\*"),Optional.*empty*(),Optional.*empty*())));

devTools.send(Fetch.*enable*(patterns, Optional.*empty*()));

devTools.addListener(Fetch.*requestPaused*(), request ->

{

devTools.send(Fetch.*failRequest*(request.getRequestId(), ErrorReason.***FAILED***));

});

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.*cssSelector*("button[routerlink\*='library']")).click();

}

}

225. Blocking unwanted Network request calls to speed up the execution with selenium

when do we use network block?

 now you want to test an application, a functional logic around it. Due to some images loading it taking more time to load

So you are doing some functional validation and those images are loading is really blocking you and making your execution slower.

So it is more applicable when you have e-commerce websites like Amazon.

If you see like that, there are a lot of images which is really not required for your automation.

So you can go ahead and block all of them so that you can smoothly continue with your execution.

Now for example, I want to click on Browse products and I want to click on Selenium and then add to Cart. And I have to choose the cart. So this is my functionality.

So if you want to do it faster by not loading images of this page.

Because let's say image is loading in the page, it is taking little more time so you can actually block them.

So once you create the session. So you should start enabling your network before you perform any operation.

**import** java.util.Optional;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.devtools.DevTools;

//import org.openqa.selenium.devtools.

**import** org.openqa.selenium.devtools.v120.network.Network;

**import** com.google.common.collect.ImmutableList;

**public** **class** BlockNetworkRequest {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

// **TODO** Auto-generated method stub

//css , images

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

//log file ->

DevTools devTools = driver.getDevTools();

devTools.createSession();

devTools.send(Network.*enable*(Optional.*empty*(), Optional.*empty*(), Optional.*empty*()));

devTools.send(Network.*setBlockedURLs*(ImmutableList.*of*("\*.jpg","\*.css")));

**long** startTime = System.*currentTimeMillis*();

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.*linkText*("Browse Products")).click();

driver.findElement(By.*linkText*("Selenium")).click();

driver.findElement(By.*cssSelector*(".add-to-cart")).click();

System.***out***.println(driver.findElement(By.*cssSelector*("p")).getText());

**long** endTime = System.*currentTimeMillis*();

System.***out***.println(endTime - startTime);

//1793 2033

}

}

226. How to emulate network speed with Selenium Chromedevtools Integration

while you are developing the automation test in the morning of your time, you don't have much traffic to your application servers, so you don't see any, uh, wait issues.

when they are running in the nights when its

application have too much traffic, then you know you will come across this slowness and your test will start failing. That is the main reason why you see, tests are flaky in front end automation

**import** java.util.Optional;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.devtools.DevTools;

**import** org.openqa.selenium.devtools.v120.network.Network;

**import** org.openqa.selenium.devtools.v120.network.model.ConnectionType;

**public** **class** NetworkSpeed {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//waits - 2-3 , 2

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

//log file ->

DevTools devTools = driver.getDevTools();

devTools.createSession();

//Network.emulateNetworkConditions

devTools.send(Network.*enable*(Optional.*empty*(), Optional.*empty*(), Optional.*empty*()));

devTools.send(Network.*emulateNetworkConditions*(**false**, 3000, 20000, 100000, Optional.*of*(ConnectionType.***ETHERNET***)));

// 3000 - 3 sec dely to response back from the server

// 2000 - 2 sec how many bytes you have to download per second

// ETHERNET - Cable connection

devTools.addListener(Network.*loadingFailed*(), loadingFailed->

{

System.***out***.println(loadingFailed.getErrorText());

System.***out***.println(loadingFailed.getTimestamp());

});

**long** startTime = System.*currentTimeMillis*();

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.*cssSelector*("button[routerlink\*='library']")).click();

**long** endTime = System.*currentTimeMillis*();

System.***out***.println(endTime - startTime);

/\*

\* driver.get("http://google.com");

\* driver.findElement(By.name("q")).sendKeys("netflix",Keys.ENTER);

\* driver.findElements(By.cssSelector(".LC20lb")).get(0).click(); String title

\* =driver.findElement(By.cssSelector(".our-story-card-title")).getText();

\* System.out.println(title);

\* long endTime = System.currentTimeMillis();

\* System.out.println(endTime - startTime);

\* driver.close();

\* //14960 2054

\*/

}

}

227. Working with Basic Authentication using Selenium uriPredicate function

sometimes you might see a popup like this to enter username and password. so when you hit that website URL this is how it shows up.

when you provide username and password.So this is nothing but basic authentication request,

So this is an HTTP get request. We are hitting on the browser, but with this URL we have enabled basic authentication. So when I say basic authentication this URL also expects username and password to get the response, So if you don't provide these details

then you won't get the response back because this API is backed up with basic authentication.

So when you click on cancel You will not get the data and when you refresh it and then again it will pop up.

whenever you see such kind of webpages in your company or anywhere that means those are tagged with basic authentication and with Selenium, how will you handle it?

This is not a web popup for you to get credentials locators and to write it in your script.

So this is entirely a window popup where Selenium cannot act upon it.

So from Java 8, there is a concept called Predicate.

So what is Predicate?

Predicate will help you to create one filter condition for your data

Consumer consumes that event emitted object.

So what does consumer do?

Consumer consumes the data and do nothing. So that means they don't return anything.

So that's why if you carefully observe, this is a consumed object and here we are printing the details like request dot, get me that, get me this. But you are not returning anything.

Here you are just executing the commands,but you don't see any return statement here.

So consumer duty is just to consume and manipulate the data but it don't return anything.

if it is more than one line you can actually get rid of this curly braces and write that in single line

with the help of lambda expression.

so there is a class called URI in Java. So this class will help you to take the URL of the what you have hidden in the browser and it will help you to pass it.

what Selenium is asking, if you encounter any basic authentication pop-up stuff then go and register it

**import** org.openqa.selenium.HasAuthentication;

**import** org.openqa.selenium.UsernameAndPassword;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** java.net.URI;

**import** java.util.function.Predicate;

**public** **class** BasicAuthentication {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

//predicate, consumer

Predicate<URI> uriPredicate =uri -> uri.getHost().contains("httpbin.org");

((HasAuthentication)driver).register(uriPredicate,UsernameAndPassword.*of*("foo", "bar"));

driver.get("http://httpbin.org/basic-auth/foo/bar");

}

}

228. How to log javascript errors from Selenium Script to console for debugging

how to log Java script errors on your application.

Now for example, let's come to the homepage of this app, and from here I want to browse the products, select the console tab here to see if there are any JavaScript errors. Because if there is any failure on JavaScript site you should see in the console logs here from Chrome dev tools.

in real time, I recommend you to write that in that TestNG on test failure listener

because you don't know where it will fail. You just can't write after each

and every action on Selenium. So if you write in this block, whenever it failed

at that particular moment, it will capture, just like screenshot.

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.logging.LogEntries;

**import** org.openqa.selenium.logging.LogEntry;

**import** org.openqa.selenium.logging.LogType;

**public** **class** ConsoleLogsCapture {

**public** **static** **void** main(String[] args) {

//System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

ChromeDriver driver = **new** ChromeDriver();

//listeners - OnTestFailure

driver.get("https://rahulshettyacademy.com/angularAppdemo/");

driver.findElement(By.*linkText*("Browse Products")).click();

driver.findElement(By.*partialLinkText*("Selenium")).click();

driver.findElement(By.*cssSelector*(".add-to-cart")).click();

driver.findElement(By.*linkText*("Cart")).click();

driver.findElement(By.*id*("exampleInputEmail1")).clear();

driver.findElement(By.*id*("exampleInputEmail1")).sendKeys("2");

LogEntries entry= driver.manage().logs().get(LogType.***BROWSER***); //Get LogEntries object

List<LogEntry>logs = entry.getAll(); //LogEntryobject- getAll method return all logs in list

**for**(LogEntry e : logs)//iterating through list and printing each log message

{

System.***out***.println(e.getMessage()); //Log4j

}

}

}