

How to Freeze DOM

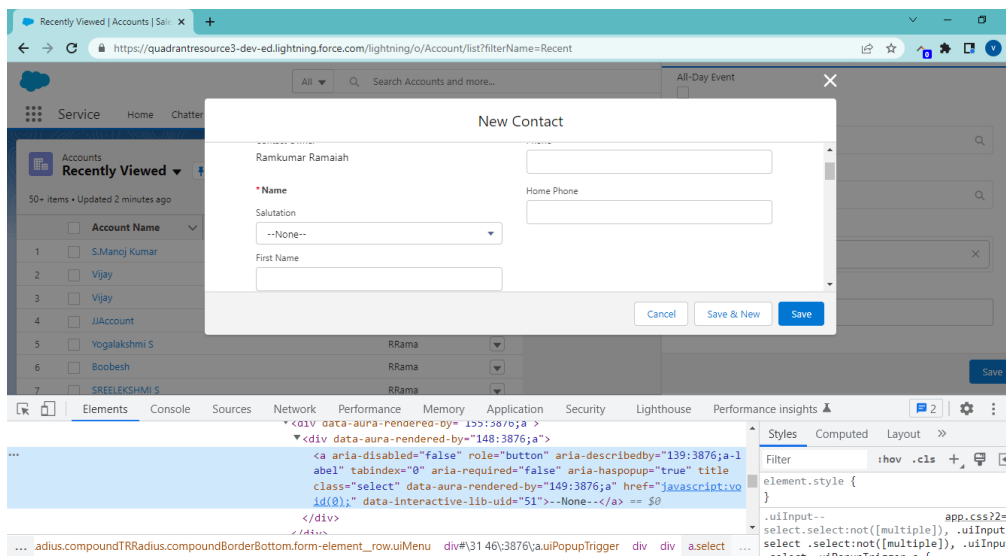
In few scenarios, we are unable to locate/inspect element in Chrome Dev tool by freezing the DOM.

Either F8 or Pause option doesn't work (with elements in dropdown or calendar). On right click and trying to inspect, element disappears in UI and unable to inspect in DOM, which has been supported in Previous versions of the browser, predominantly with Salesforce

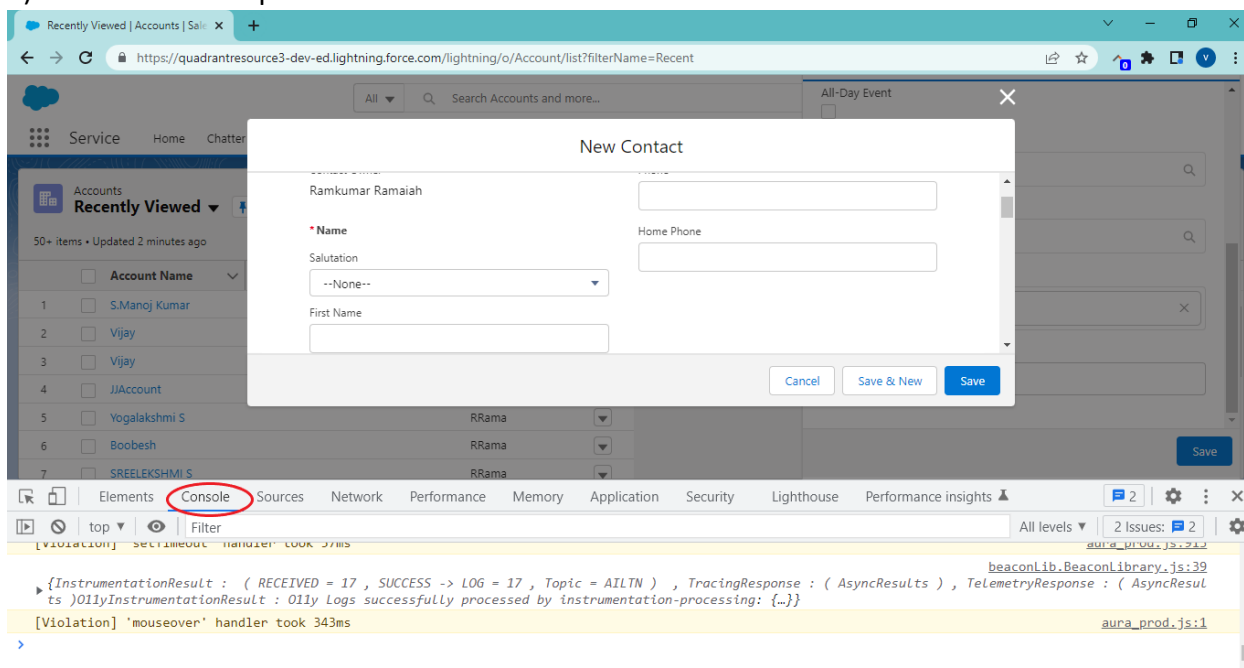
Solution and Steps to be Followed in Chrome Dev Tools:

1) Navigate to the WebPage

2) Open the Chrome Dev Tools with either options F12 or Right Click + Inspect or ctrl+shift+i



3) Click on Console panel in Chrome Dev Tools

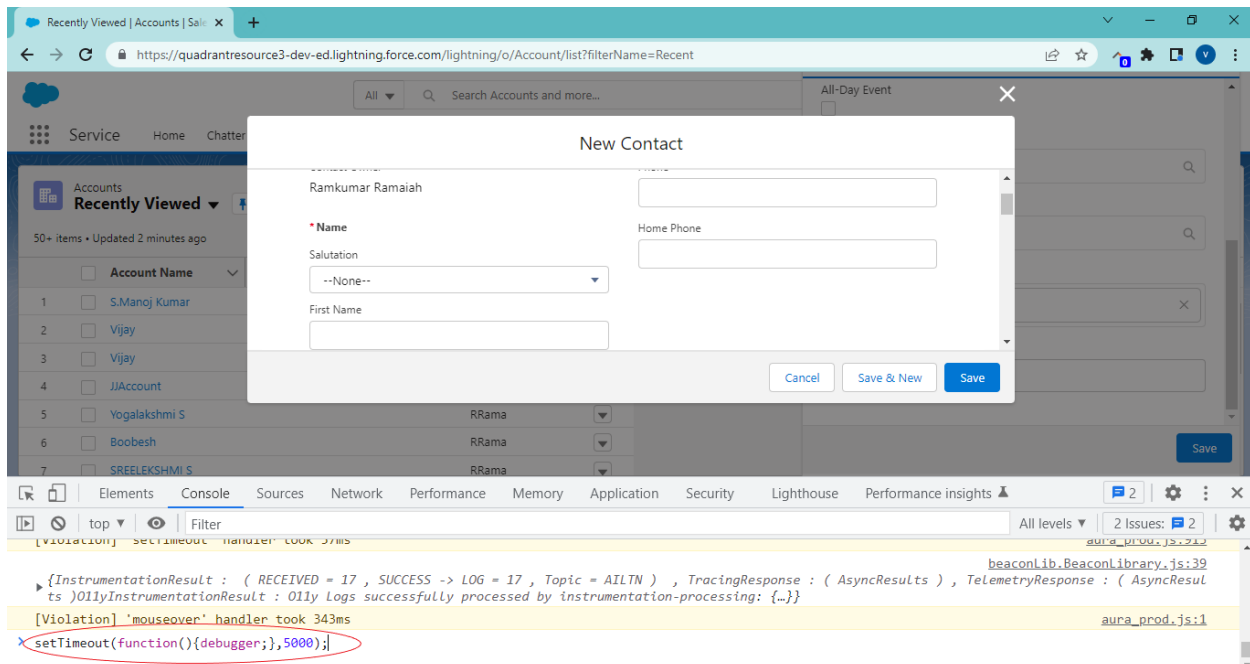


4) Type or Paste the Below lines of code and press the Enter Key

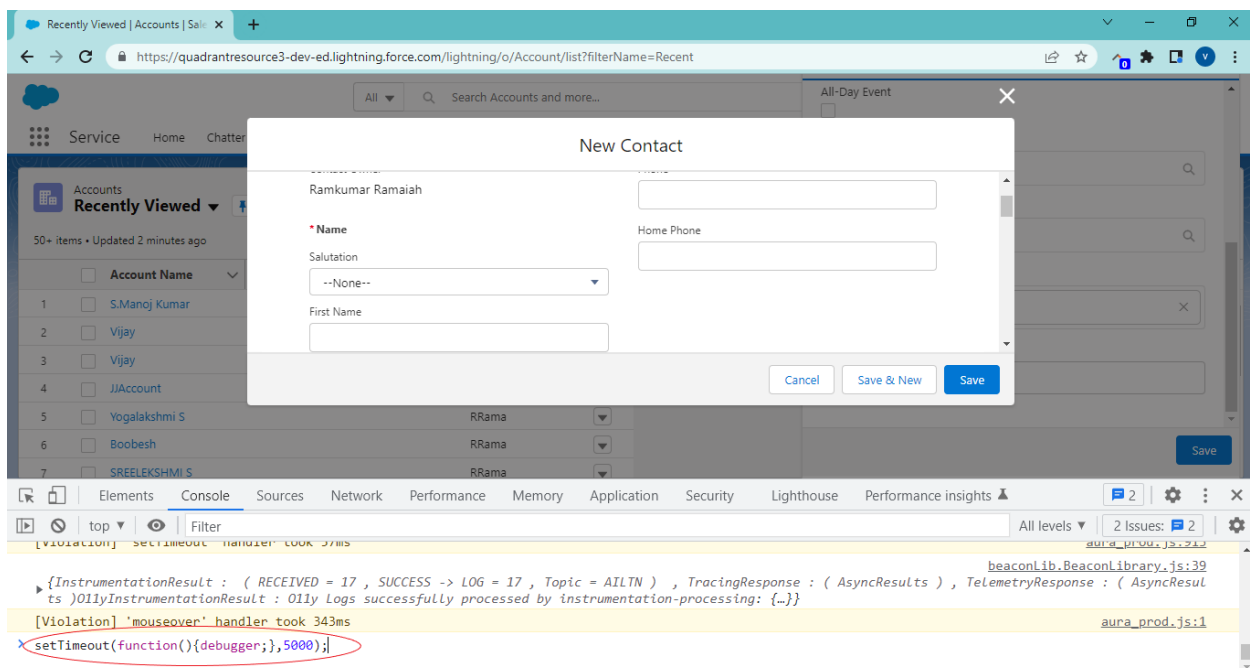
```
setTimeout(function(){debugger;},5000);
```

--Time unit here is Milliseconds 5000 is 5 Seconds

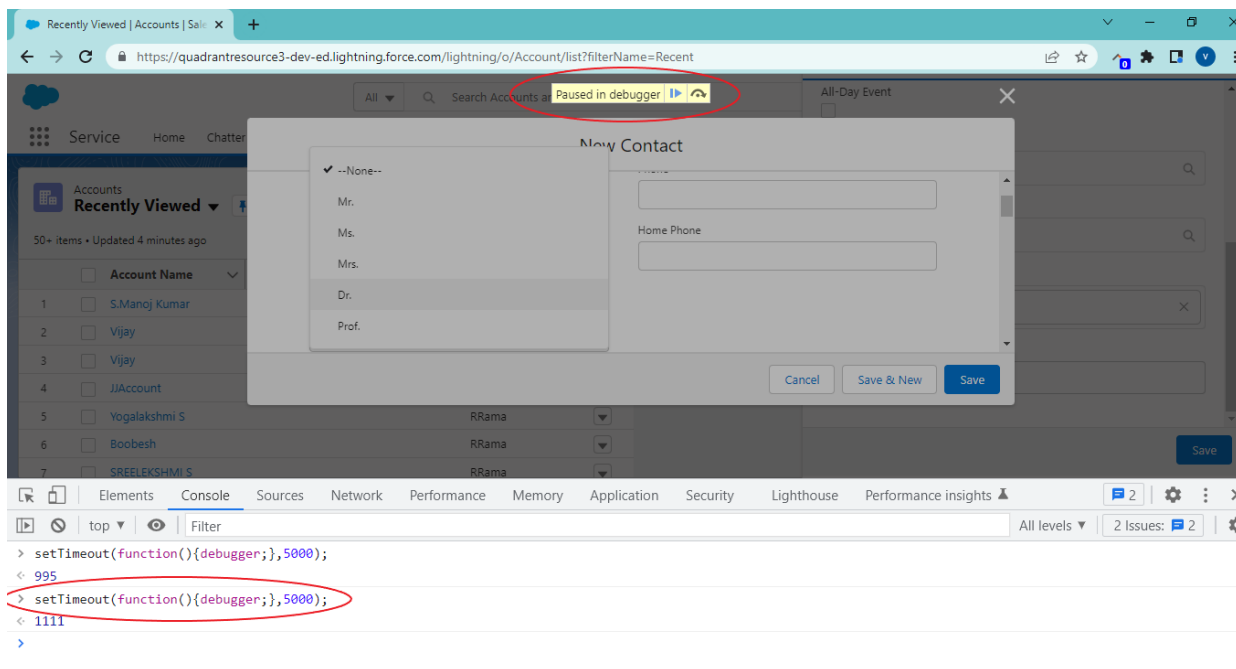
Adding wait to freeze the DOM in Debugger mode which has been done programatically in JavaScript



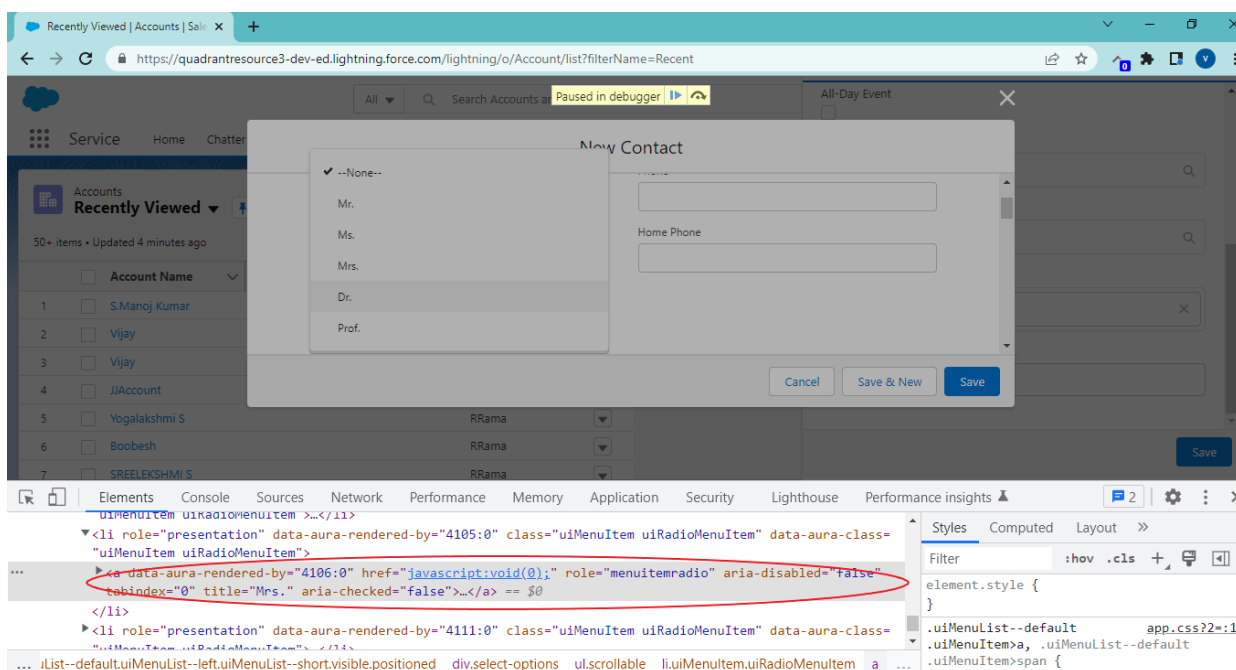
5) Do the action for the element to be inspected ,Say Click for dropdown (Refer video)



6) Wait for few seconds, Debugger Mode (respect to action on F8) will be enabled



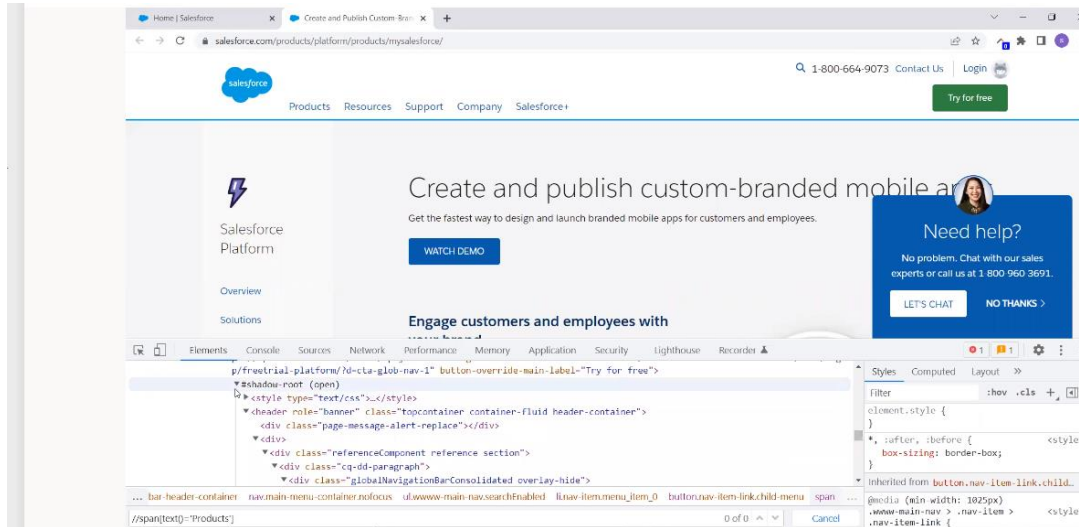
7) Indicate the Element to be located and same appears in Element Tab



8) Now follow the locator strategy with defined hierarchy (Say ID/name/Class if unique or going forward with XPath) to locate element.

How to handle Shadow Root DOM:

Shadow root is a special type of DOM which encapsulates the webElements in a web page. If any webpage is designed by developer with Shadow DOM, then the elements cannot be located by findElement method.



In order to automate the web elements which are kept inside the shadow dom, there are 2 steps to be followed

Step:1-> the project should have the **sukgu** dependency added to the pom.xml

Dependency to be added in pom.xml

```
<!-- https://mvnrepository.com/artifact/io.github.sukgu/automation -->
```

```
<dependency>
```

```
    <groupId>io.github.sukgu</groupId>
```


```
    <artifactId>automation</artifactId>
```

```
    <version>0.1.4.1</version>
```

```
</dependency>
```

Step:2 -> Instantiate the Shadow class

```
Shadow dom=new Shadow(driver);
```



```
import io.github.sukgu.Shadow;
```

Using shadowdom object the elements are located in the webpage

Sample Code:

```
EY-SDET - Selenium/src/main/java/learnSelenium/LearnShadowdom.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

LearnShadowdom.java
1 package learnSelenium;
2
3 import java.time.Duration;
12
13 public class LearnShadowdom {
14
15     public static void main(String[] args) throws InterruptedException {
16
17         WebDriverManager.chromedriver().setup();
18         ChromeDriver driver = new ChromeDriver();
19
20         driver.manage().window().maximize();
21         driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));
22         driver.get("https://www.salesforce.com/products/platform/products/m
23         Shadow dom = new Shadow(driver);
24         WebElement pdts = dom.findElementByXPath("//span[text()='Products']
25         pdts.click();
26         String title = driver.getTitle();
27         System.out.println(title);
28     }
29 }
30
31 }
32
```

```
<terminated> LearnShadowdom [Java Application] C:\Users\j...eclipse\plugins\org.eclipse.justj...oper
SLF4J: Failed to load class "org.slf4j.impl.Static
SLF4J: Defaulting to no-operation (NOP) logger imp
SLF4J: See http://www.slf4j.org/codes.html#StaticL
ERROR StatusLogger Log4j2 could not find a logging
Starting ChromeDriver 103.0.5060.134 (8ec6fce403b3
Only local connections are allowed.
Please see https://chromedriver.chromium.org/secur
ChromeDriver was started successfully.
Jul 29, 2022 1:48:44 PM org.openqa.selenium.remote
INFO: Detected upstream dialect: W3C
Jul 29, 2022 1:48:45 PM org.openqa.selenium.devtoo
WARNING: Unable to find an exact match for CDP ver
Jul 29, 2022 1:48:45 PM org.openqa.selenium.devtoo
INFO: Found CDP implementation for version 103 of
Create and Publish Custom-Branded Mobile Apps - Sa
```