

## Deep Dive into the Issue: ElementClickInterceptedException

The error message:

```
org.openqa.selenium.ElementClickInterceptedException: element click intercepted:
```

```
Other element would receive the click: <div aria-hidden="true" class="MuiBackdrop-root css-41vu0l">...</div>
```

indicates that **Selenium is unable to click the intended element (#select-category) because another element (MuiBackdrop-root) is blocking it.**

### 1 Understanding Why This Happens

#### What is ElementClickInterceptedException?

This exception occurs when:

- The intended element **is not in the foreground**, meaning another element (often an overlay, modal, or backdrop) is covering it.
- **Clicking it would instead interact with the overlaying element**, not the dropdown.

#### Why is MuiBackdrop-root Blocking the Click?

Your UI framework (likely **Material-UI**) uses a **backdrop div** (<div class="MuiBackdrop-root">) to **create a fading effect or block interactions** when a dropdown or modal is opening/closing.

Here's what's likely happening:

1. You try to click the dropdown (#select-category).
2. A **backdrop div** (MuiBackdrop-root) is rendered **above** the dropdown.

3. Selenium detects that clicking would actually affect the backdrop, so it throws `ElementClickInterceptedException`.

## 2 How to Confirm the Issue

Before fixing the issue, **verify that the backdrop is covering the dropdown**:

### 1. Inspect the Element (F12 in Chrome DevTools)

- a. Right-click `#select-category` and check if a `<div class="MuiBackdrop-root">` is present.
- b. If it **disappears when you manually open the dropdown**, it means it's an overlay.

### 2. Use JavaScript in the Console to Check Element Position

Run this in the browser console:

```
document.querySelector("#select-category").getBoundingClientRect();
```

If the top and height values indicate it's **behind another element**, that confirms the issue.

### 3. Check for CSS Styles Like `z-index`

- a. The backdrop might have a **higher `z-index`**, which means it's visually on top of the dropdown.
- b. Look for:

```
.MuiBackdrop-root {  
  z-index: 9999; /* High value means it's above everything */  
}
```

## Solutions Based on Root Cause

### Case 1: Backdrop is Temporary (Wait for it to Disappear)

If the backdrop disappears **after a short delay**, you should **wait for it to go away** before clicking the dropdown.

```
WebDriverWait wait = new WebDriverWait(driver,
Duration.ofSeconds(10));
wait.until(ExpectedConditions.invisibilityOfElementLocated(By.className
e("MuiBackdrop-root")));
```

```
WebElement staticDropdown =
driver.findElement(By.cssSelector("#select-category"));
staticDropdown.click();
```

✅ **Best for:** UI where the backdrop fades out after a delay.

### Case 2: Backdrop Stays (Use JavaScript to Click)

If the backdrop **stays and does not disappear**, force Selenium to bypass it and click directly using JavaScript:

```
WebElement staticDropdown =
driver.findElement(By.cssSelector("#select-category"));
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("arguments[0].click();", staticDropdown);
```

✅ **Best for:** UI where backdrop remains **but dropdown is interactable**.

### Case 3: Click the Parent Element Instead

If the dropdown is inside a bigger container that **is not blocked**, clicking its parent can sometimes work:

```
WebElement dropdownParent =  
driver.findElement(By.xpath("//div[@id='select-  
category']/parent::*"));  
dropdownParent.click();
```

✅ **Best for:** Dropdowns wrapped inside a larger div.

### Case 4: Simulate Keyboard Input Instead of Clicking

If the dropdown allows keyboard navigation, you can **send keyboard events** instead:

```
WebElement staticDropdown =  
driver.findElement(By.cssSelector("#select-category"));  
staticDropdown.sendKeys(Keys.ARROW_DOWN);  
staticDropdown.sendKeys(Keys.ENTER);
```

✅ **Best for:** Forms where dropdowns can be controlled via keyboard.

## Final Steps: How to Choose the Best Fix

1. Check if the backdrop disappears naturally
  - a. ✅ If yes → Use **Solution 1 (wait for invisibility)**.
  - b. ❌ If not, go to Step 2.
2. Check if clicking with JavaScript works
  - a. ✅ If yes → Use **Solution 2 (JavaScript Click)**.
  - b. ❌ If not, go to Step 3.
3. Try clicking a parent element
  - a. ✅ If it works → Use **Solution 3 (Click Parent)**.

b. ❌ If not, go to Step 4.

#### 4. Use keyboard navigation

a. ✅ If the dropdown supports it → Use **Solution 4 (Send Keys)**.

## 5 Summary

Solution	Works When	Code
Wait for backdrop to disappear	Backdrop fades out after time	<code>wait.until(ExpectedConditions.invisibilityOfElementLocated(...));</code>
Click using JavaScript	Backdrop remains, but dropdown is active	<code>js.executeScript("arguments[0].click();", element);</code>
Click parent element	Parent is clickable	<code>driver.findElement(By.xpath("//div[@id='select-category']/parent::*")).click();</code>
Use keyboard navigation	Dropdown supports keys	<code>element.sendKeys(Keys.ARROW_DOWN);</code>

## Next Steps

- Try **waiting first** (Solution 1) and **use JavaScript if that fails** (Solution 2).
- If those don't work, **test the parent click or keyboard navigation**.

Let me know which solution works for your case! 