To take screenshots using AShot, you'll need to add the AShot dependency to your project and then use the AShot library to capture screenshots of the entire page or specific elements.

1. Add the AShot Dependency:

maven.

Code

<dependency>  
 <groupId>ru.yandex.qatools.ashot</groupId>  
 <artifactId>ashot</artifactId>  
 <version>1.1</version>  
 </dependency>

Gradle.

Code

implementation 'ru.yandex.qatools.ashot:ashot:1.1'

* **Other Dependency Managers:** Refer to the [AShot GitHub repository](https://github.com/pazone/ashot) for instructions on adding the dependency using other dependency managers.

2. Code to Take a Full Page Screenshot:

Java

import ru.yandex.qatools.ashot.AShot;  
 import ru.yandex.qatools.ashot.Screenshot;  
 import ru.yandex.qatools.ashot.shooting.ShootingStrategies;  
 import org.openqa.selenium.WebDriver;  
 import java.io.File;  
 import java.io.IOException;  
 import javax.imageio.ImageIO;  
  
 public class FullPageScreenshot {  
  
 public static void main(String[] args) throws IOException {  
 *// Replace with your WebDriver instance*  
 WebDriver driver = new ChromeDriver(); *// Example using ChromeDriver*  
 driver.get("https://www.example.com"); *// Replace with the URL*  
  
 *// Take the screenshot*  
 Screenshot screenshot = new AShot()  
 .shootingStrategy(ShootingStrategies.viewportPasting(1000)) *// Scrolls the page to capture the full content*  
 .takeScreenshot(driver);  
  
 *// Save the screenshot*  
 File file = new File("full\_page\_screenshot.png");  
 ImageIO.write(screenshot.getImage(), "PNG", file);  
  
 System.out.println("Screenshot saved to: " + file.getAbsolutePath());  
  
 driver.quit();  
 }  
 }

3. Code to Take a Screenshot of a Specific Element:

Java

import ru.yandex.qatools.ashot.AShot;  
 import ru.yandex.qatools.ashot.Screenshot;  
 import org.openqa.selenium.WebDriver;  
 import org.openqa.selenium.WebElement;  
 import org.openqa.selenium.By;  
 import java.io.File;  
 import java.io.IOException;  
 import javax.imageio.ImageIO;  
  
 public class ElementScreenshot {  
  
 public static void main(String[] args) throws IOException {  
 *// Replace with your WebDriver instance*  
 WebDriver driver = new ChromeDriver(); *// Example using ChromeDriver*  
 driver.get("https://www.example.com"); *// Replace with the URL*  
  
 *// Find the element*  
 WebElement element = driver.findElement(By.id("myElement")); *// Replace with the element's locator*  
  
 *// Take the screenshot*  
 Screenshot screenshot = new AShot().takeScreenshot(driver, element);  
  
 *// Save the screenshot*  
 File file = new File("element\_screenshot.png");  
 ImageIO.write(screenshot.getImage(), "PNG", file);  
  
 System.out.println("Screenshot saved to: " + file.getAbsolutePath());  
  
 driver.quit();  
 }  
 }

Explanation:

* AShot: The main class for taking screenshots.
* ShootingStrategies.viewportPasting(1000): Used for taking full page screenshots by scrolling the page and stitching the screenshots together. The 1000 value is the scrolling speed in milliseconds.
* takeScreenshot(driver): Captures the screenshot of the entire page or a specific element.
* takeScreenshot(driver, element): Captures the screenshot of a specific element.
* screenshot.getImage(): Returns the image data of the screenshot.
* ImageIO.write(): Saves the image to a file.
* File: Represents the file where the screenshot will be saved.
* driver.quit(): Closes the browser.

Important Notes:

* **WebDriver:** Make sure you have a WebDriver instance (driver) initialized and pointing to the browser you want to use.

You use the AShot library in Selenium WebDriver when you need to capture screenshots of the entire web page, specific web elements, or when you want to perform screenshot comparisons, as Selenium WebDriver itself doesn't inherently support full-page screenshots or element-specific screenshots.

Here's a breakdown of when and why you'd use AShot:

When to use AShot:

* **Capturing Full Page Screenshots:**

Selenium WebDriver's default screenshot functionality captures only the visible viewport. AShot allows you to take screenshots of the entire web page content, even if it requires scrolling.

* **Capturing Specific Web Elements:**

You can use AShot to capture screenshots of individual web elements, which can be useful for visual verification or debugging.

* **Screenshot Comparison:**

AShot provides features for comparing screenshots, which is helpful for visual regression testing to ensure that UI elements haven't changed unexpectedly.

* **Working with Different Browsers:**

While Firefox supports full-page screenshots natively, Chrome, Safari, and Edge do not. AShot provides a cross-browser solution for capturing full-page screenshots.

* **Beautifying Screenshots:**

AShot provides features to beautify screenshots, like adding borders, shadows, or other visual enhancements.

* **Fine-grained control over screenshot capture:**

AShot provides options to control the screenshot capture process, such as specifying the shooting strategy, scaling, and pasting options.

Why use AShot:

* **Extended Functionality:**

AShot extends Selenium WebDriver's screenshot capabilities, providing features not available out-of-the-box.

* **Cross-Browser Compatibility:**

AShot helps ensure consistent screenshot capture across different browsers, even those that don't natively support full-page screenshots.

* **Visual Verification:**

AShot streamlines visual verification and regression testing by allowing you to capture and compare screenshots easily.

* **Debugging:**

Capturing screenshots of specific elements or the entire page can be invaluable for debugging UI issues.