**What are the differences between Selenium 3 and Selenium 4?**

* In Selenium 3 we have to specify Web Driver executable path explicitly by writing System.*setProperty*("webdriver.chrome.driver","chrome driver path");But Where as in selenium 4 , We no need to specify the web driver executable path explicitly
* New type of Relative locators added in Selenium 4 , It uses strategy as locate neighbouring elements .The Relative locators are : To Left Of , To Right Of , below , Above
* In Selenium 4 , Actions class methods syntax was updated for

Click method , context click, click and hold And for double click method,

Actions actions = **new** Actions(*driver*);

actions.doubleClick(WebElement Object Reference).perform();

actions.contextClick(WebElement Object Reference).perform();

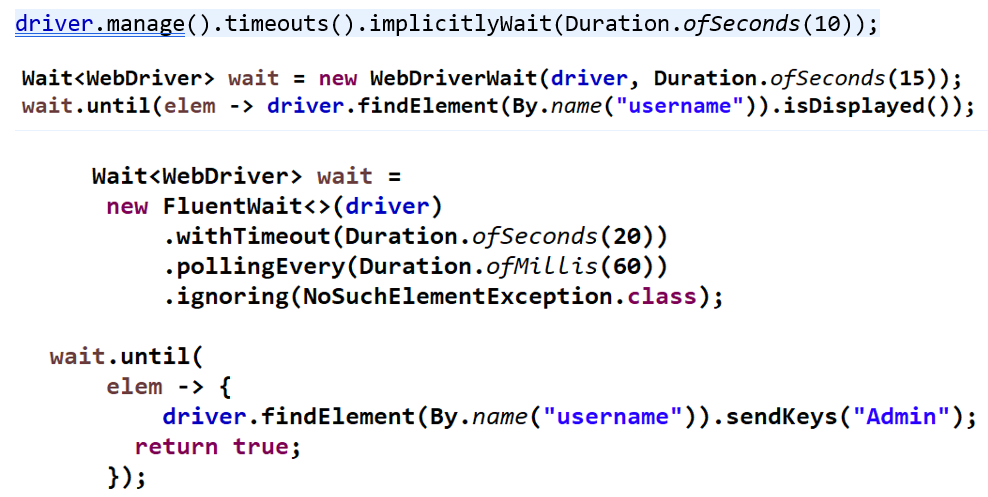
actions.dragAndDrop(Source\_Drag\_Button, Target\_Drop\_Button).perform();

actions.moveToElement(Hello\_Signin).

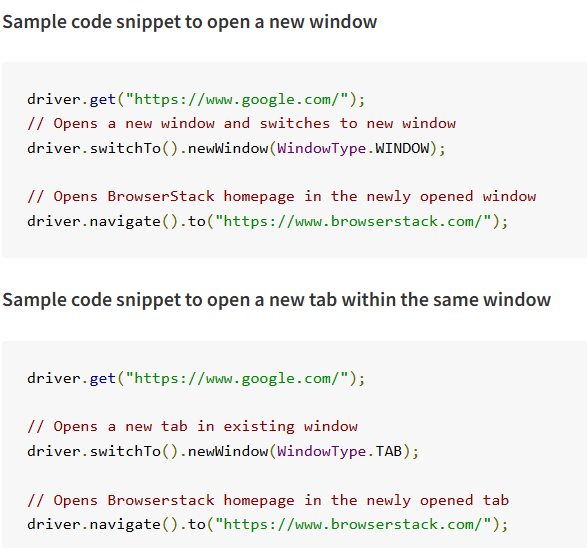
moveToElement(Your\_Account).

click().build().perform();

* In Selenium 4 , Waits syntax was updated



* In Selenium 4, there is a new method called getScreenshotAs(). It allows us to take a screenshot of a particular Web Element as well.
* With Selenium 4, we can take full page screenshots using the getFullPageScreenshotAs() method. It is only available for the Firefox browser
* in Selenium 4 as it comes with a new API – **new Window** that allows users to create and switch to a new window/tab without creating a new WebDriver object.



* In Selenium 4, Desired capabilities objects are replaced with Chrome Options. This means , now need to create an Options object, set test requirements, and pass the object to the Driver constructor.
* grid implementation feature was updated in Selenium 4 . The earlier version of [Selenium Grid](https://www.browserstack.com/guide/selenium-grid-tutorial) was complex to set up and rigid in terms of scaling. The new Selenium Grid comes with Docker support. This will enable developers or testers to spin up the containers rather than setting up heavy virtual machines

**In Selenium 3 , we have to specify web driver executable path explicitly like as**



But in selenium 4 , We no need to specify the **web driver executable path explicitly**

Simply we will write WebDriver driver=new ChromeDriver() ;

so latest chrome driver downloaded automatically according to chrome browser version

And in POM.xml , we can specify the dependencies as

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.11.0</version>

</dependency>

<dependency>

<groupId>io.github.bonigarcia</groupId>

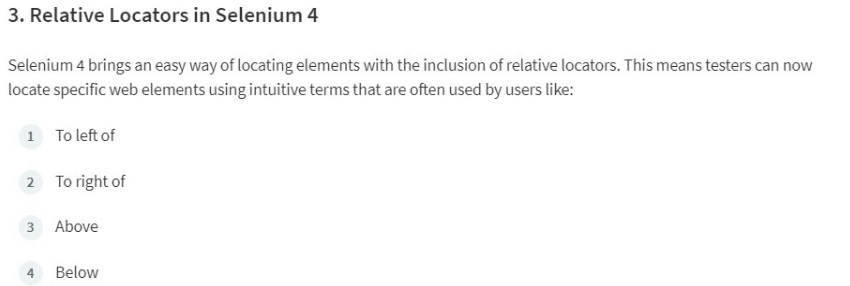
<artifactId>webdrivermanager</artifactId>

<version>5.5.3</version>

</dependency>

New type of Relative locators added in Selenium 4

**It uses strategy as locate neighbouring elements**

****

**In Selenium 4 , Actions class methods syntax was updated for**

**Click method ,context click method ,click and hold method And for double click method,**

Actions actions = **new** Actions(*driver*);

actions.doubleClick(WebElement Object Reference).perform();

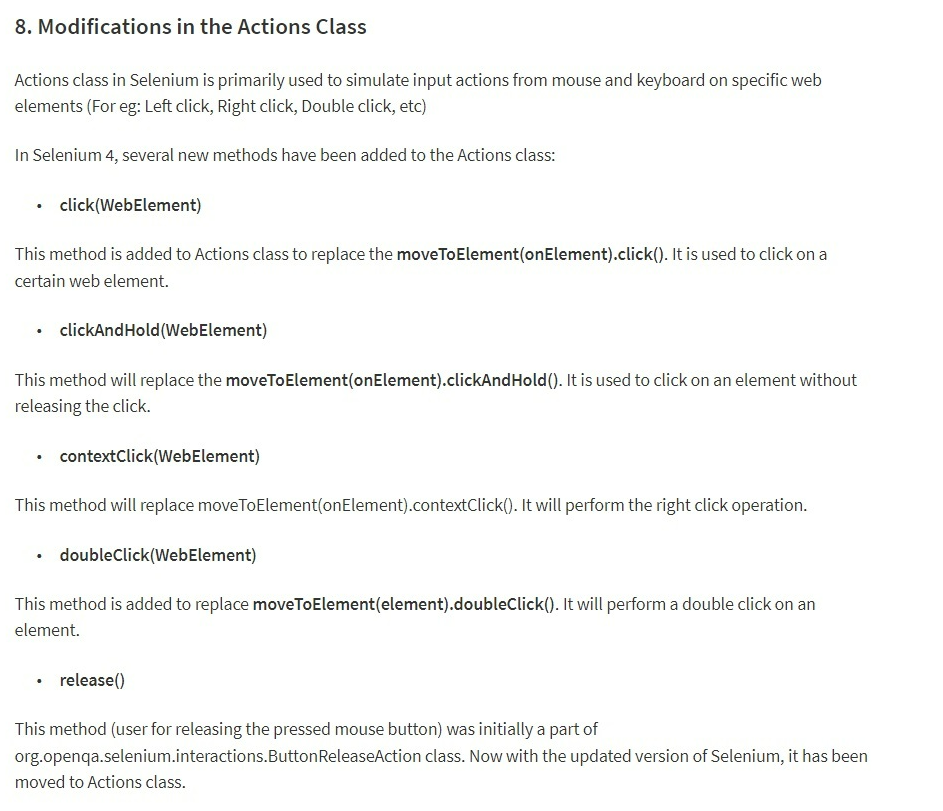
actions.contextClick(WebElement Object Reference).perform();

actions.dragAndDrop(Source\_Drag\_Button, Target\_Drop\_Button).perform();

actions.moveToElement(Hello\_Signin).

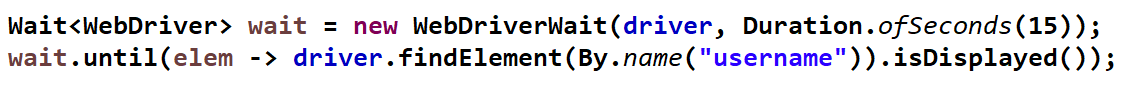
moveToElement(Your\_Account).

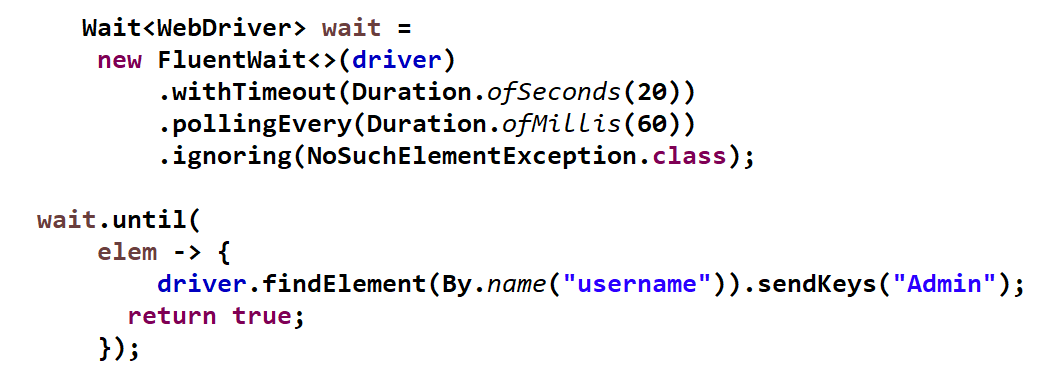
click().build().perform();

****

**In Selenium 4 , Waits syntax was updated**

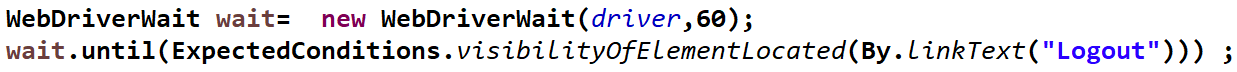
driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

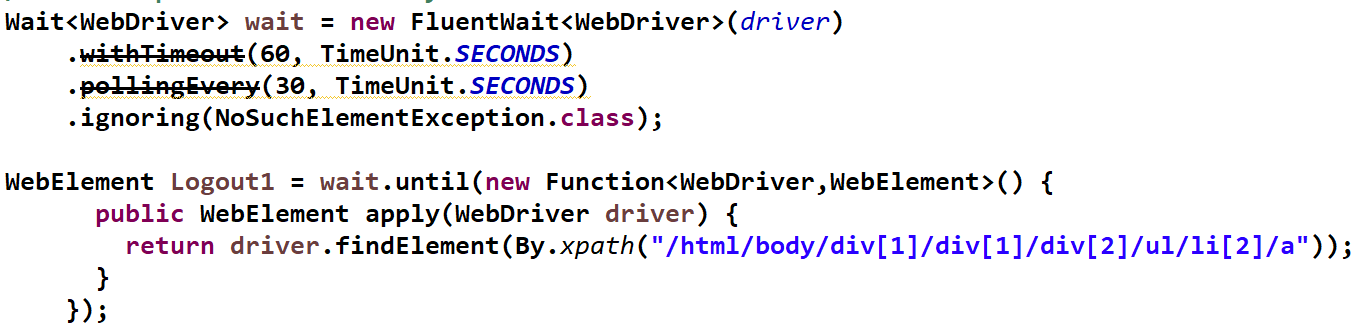


****

**In Selenium 3 Waits syntax was like**

****

****

****

**we can able to take snapshot of a particular web element as well . this new was added in selenium 4**

In Selenium 4, there is a new method called **getScreenshotAs().**It allows us to take a screenshot of a particular WebElement with this method.

WebElement elementLogo = driver.findElement(By.className("has-logo-image"));

File src = elementLogo.getScreenshotAs(OutputType.FILE);

File dest = new File(System.getProperty("user.dir") + "/screenshots/elementLogo.png");

FileHandler.copy(src, dest);

**or**

WebElement element = driver.findElement(By.id("imagetrgt"));

File source = ((TakesScreenshot)element).getScreenshotAs(OutputType.FILE);

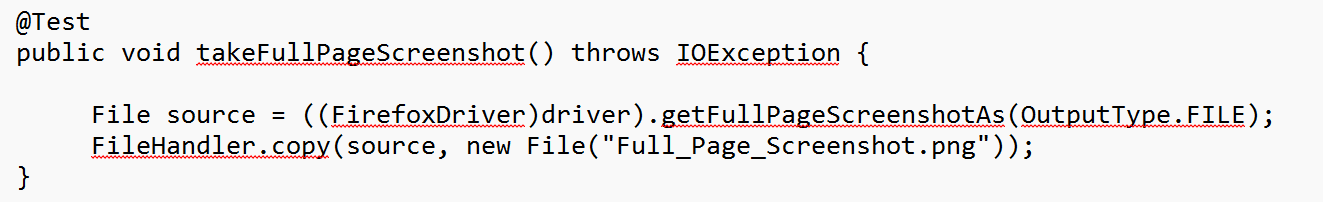
FileHandler.copy(source, new File("./screenshots/element.png"));

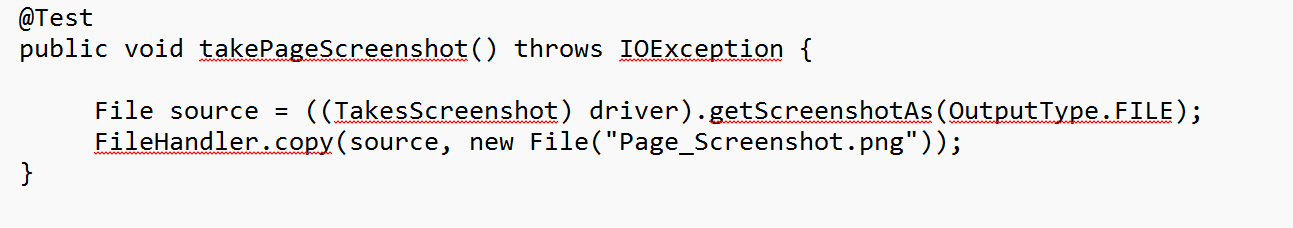
import org.openqa.selenium.OutputType;

import org.openqa.selenium.io.FileHandler;

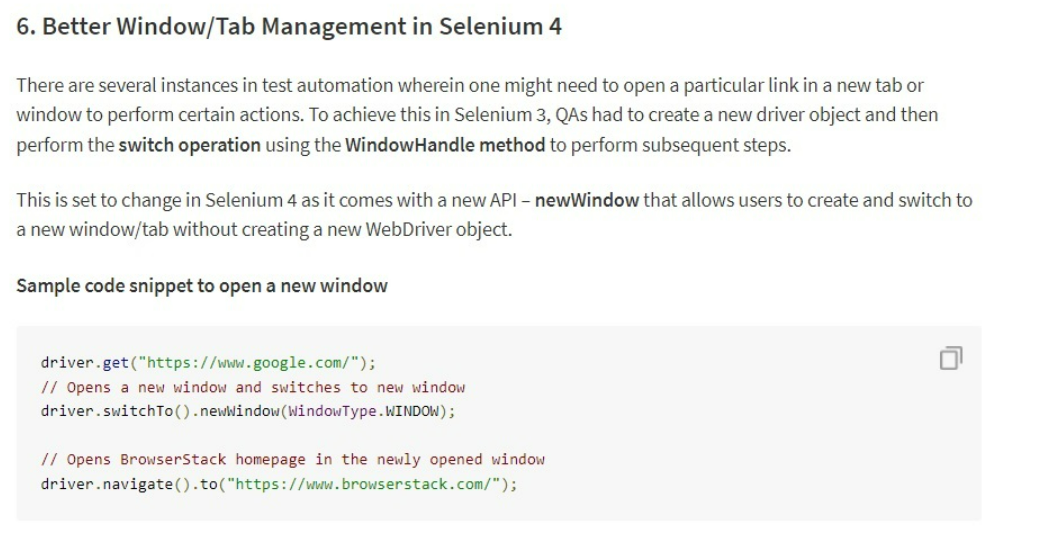
import java.io.File;

With Selenium 4, we can take full page screenshots using the getFullPageScreenshotAs() method. It is only available for the Firefox browser.

****

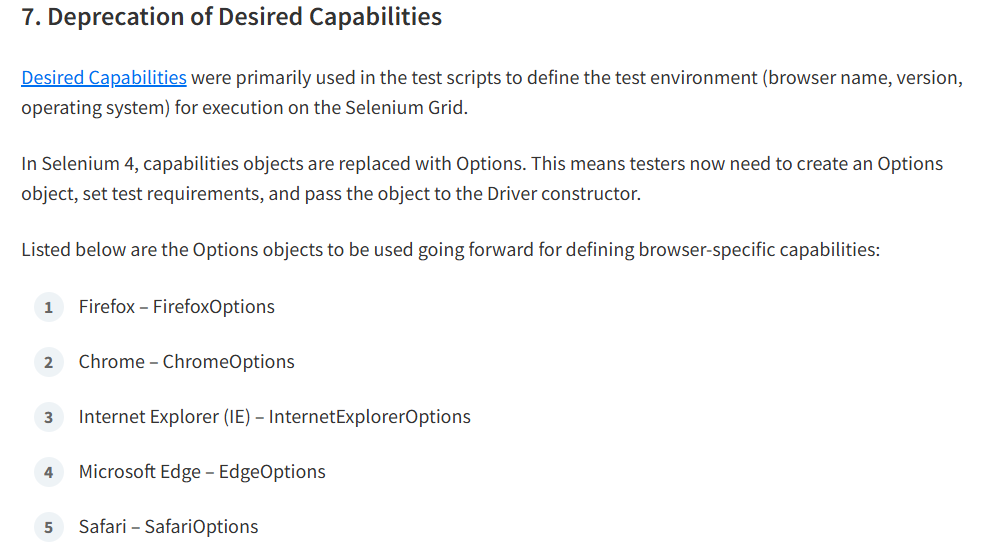
****

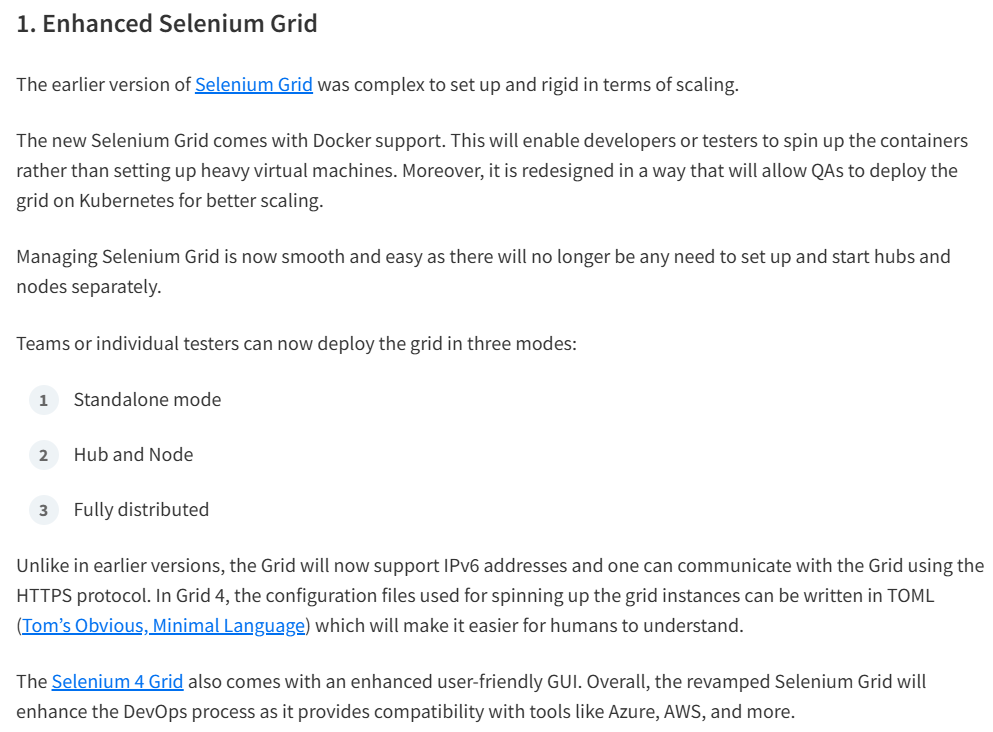
****

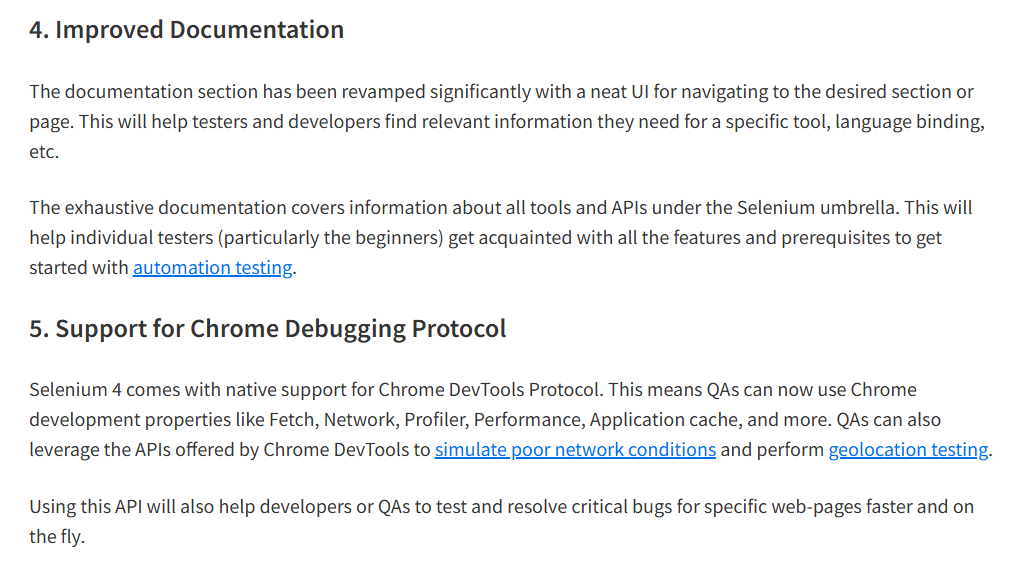
****

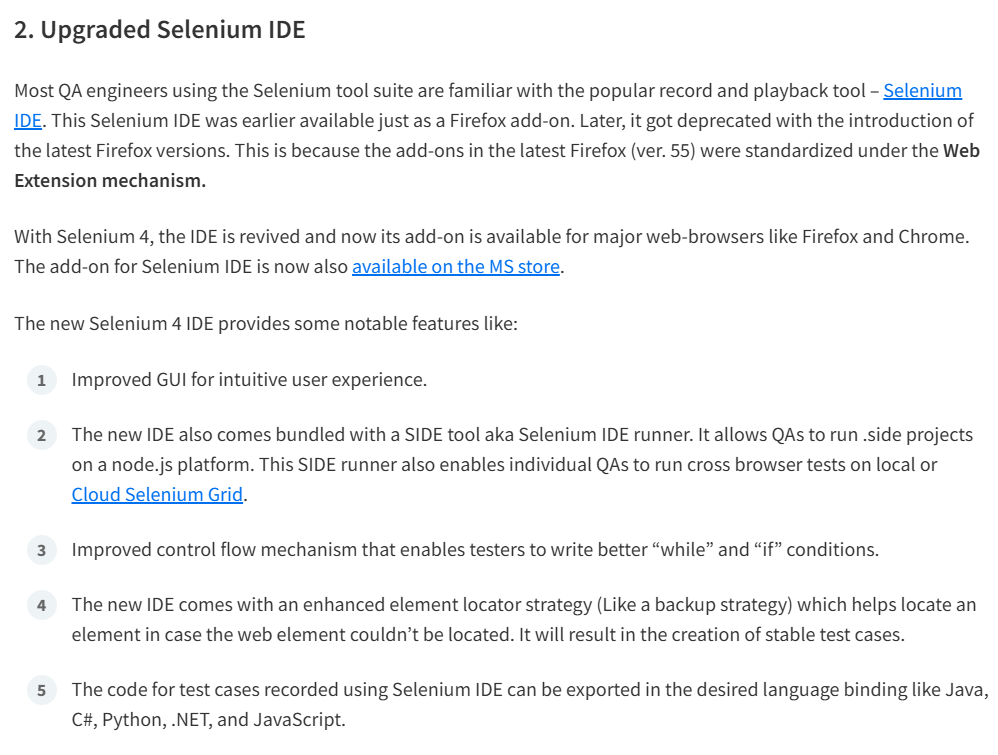
****

**In Selenium 4 Desired Capabilities was depreciated , instead of that chrome options class was replaced .**

****

****

****

****