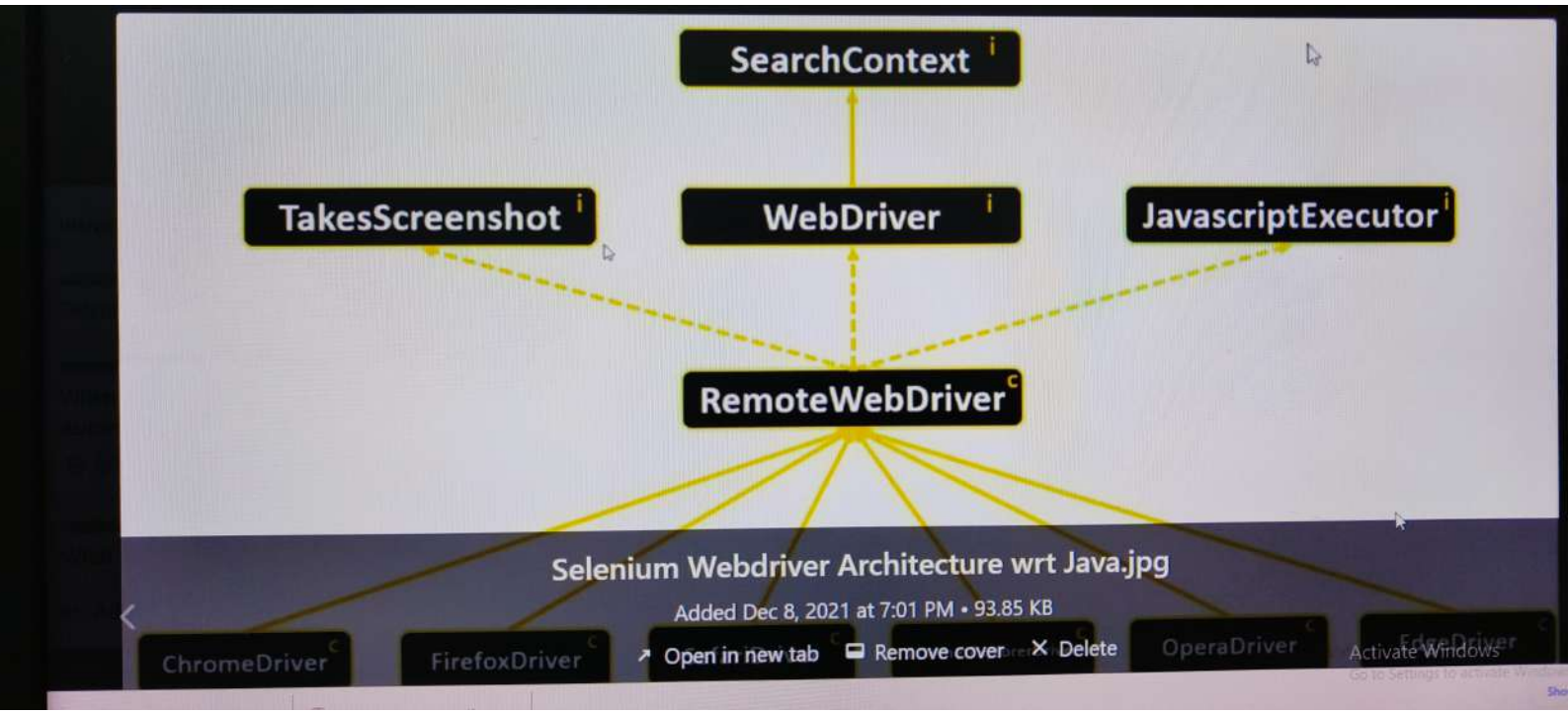
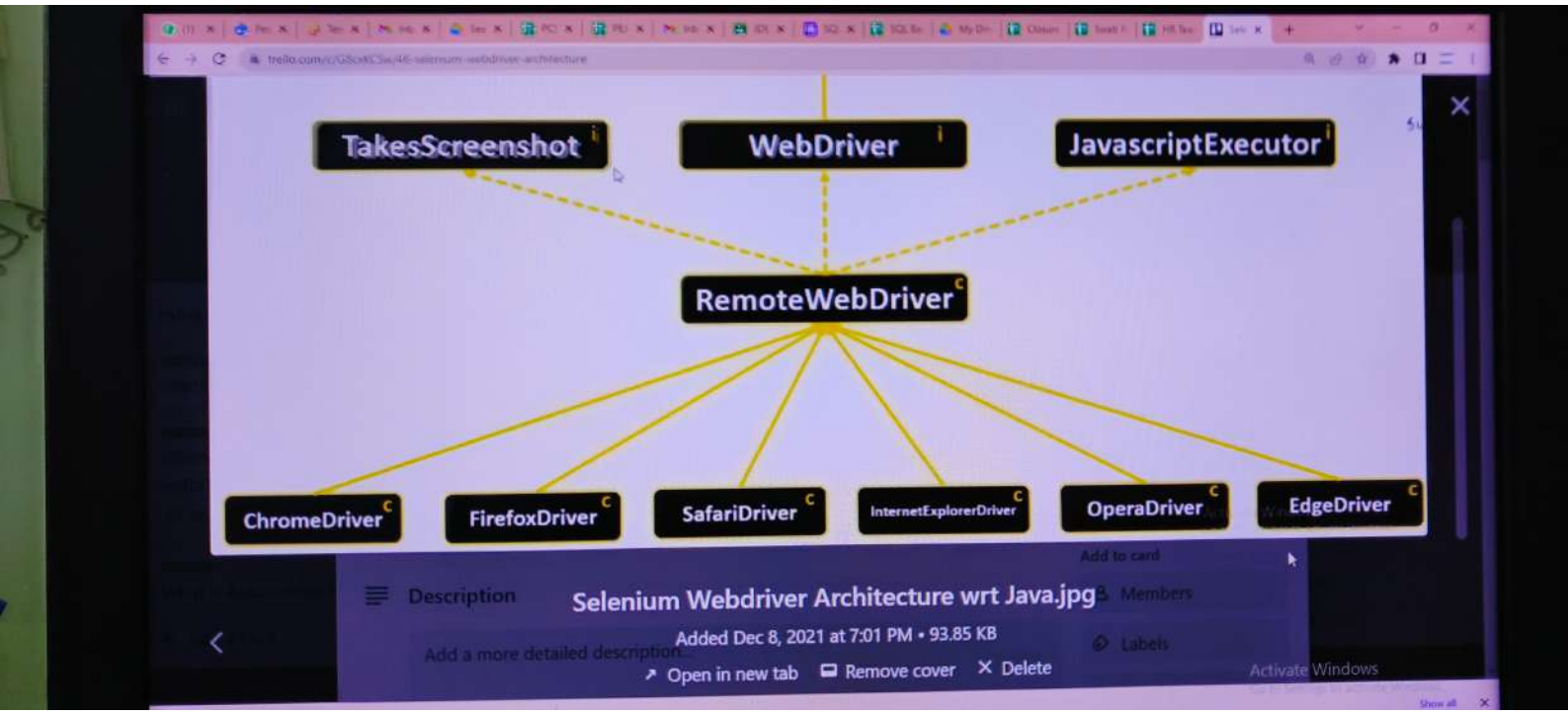
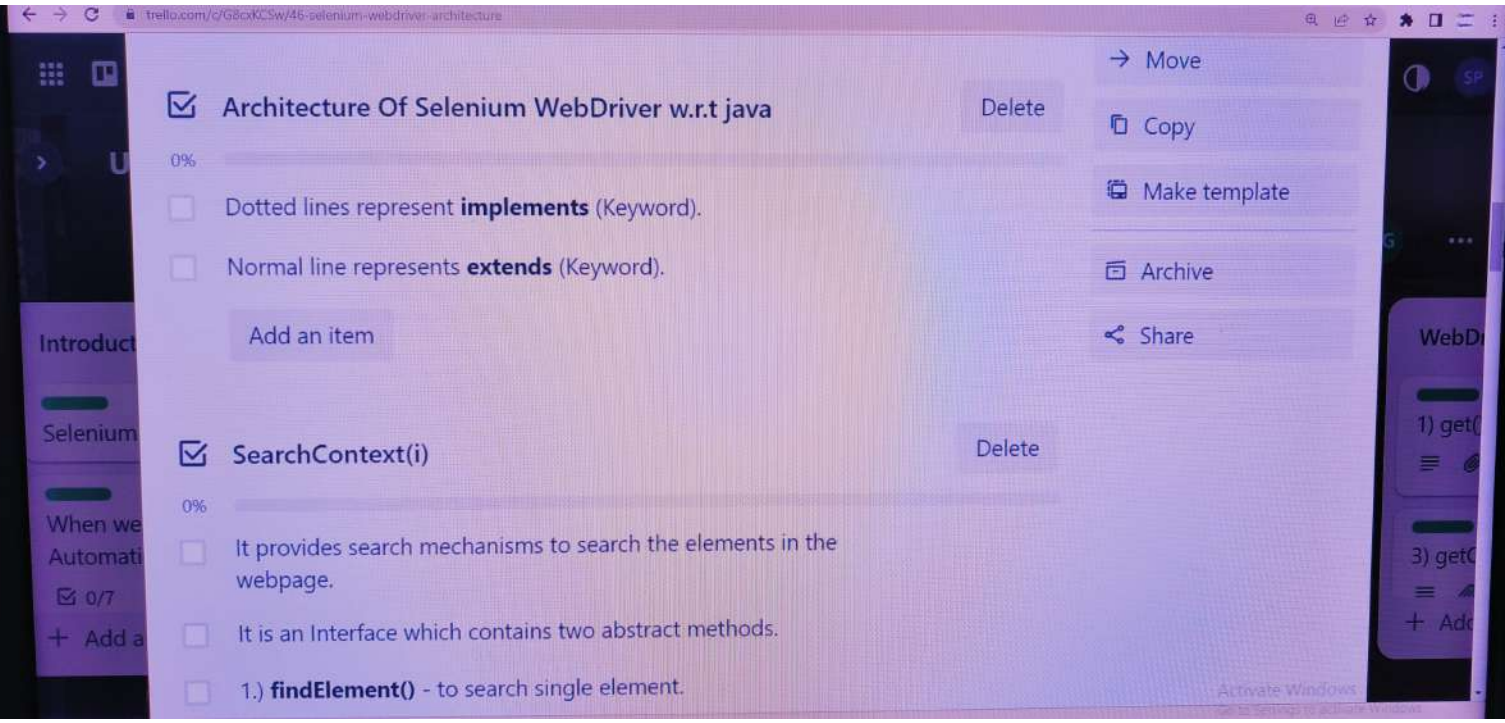

SearchContext
TakesScreenshot
WebDriver
JavascriptExecutor
RemoteWebDriver
ChromeDriver
FirefoxDriver
OperaDriver
Selenium Webdriver Architecture wrt Java.jpg
Added Dec 8, 2021 at 7:01 PM • 93.85 KB
Open in new tab
Remove cover
Delete
Activate Windows


TakesScreenshot
WebDriver
JavascriptExecutor
RemoteWebDriver
ChromeDriver
FirefoxDriver
SafariDriver
InternetExplorerDriver
OperaDriver
EdgeDriver
Description
Selenium Webdriver Architecture wrt Java.jpg
Added Dec 8, 2021 at 7:01 PM • 93.85 KB
Open in new tab
Remove cover
Delete
Add to card
Members
Labels
Add a more detailed description
Activate Windows


trello.com/c/G8cxKCSw/46-selenium-webdriver-architecture
Architecture Of Selenium WebDriver w.r.t java
Delete
0%
Dotted lines represent **implements** (Keyword).
Normal line represents **extends** (Keyword).
Add an item
Move
Copy
Make template
Archive
Share
SearchContext(i)
Delete
0%
It provides search mechanisms to search the elements in the webpage.
It is an Interface which contains two abstract methods.
1.) **findElement()** - to search single element.
Activate Windows

1.) **findElement()** - to search single element.

2.) **findElements()** - to search multiple elements.

Add an item

TakesScreenshot (Interface)

Delete

0%

- It Provides a way to take the page/element screenshot of the application under testing.
- It is an interface which contains only 01 abstract method i.e. **getScreenshotAs()**.

Add an item

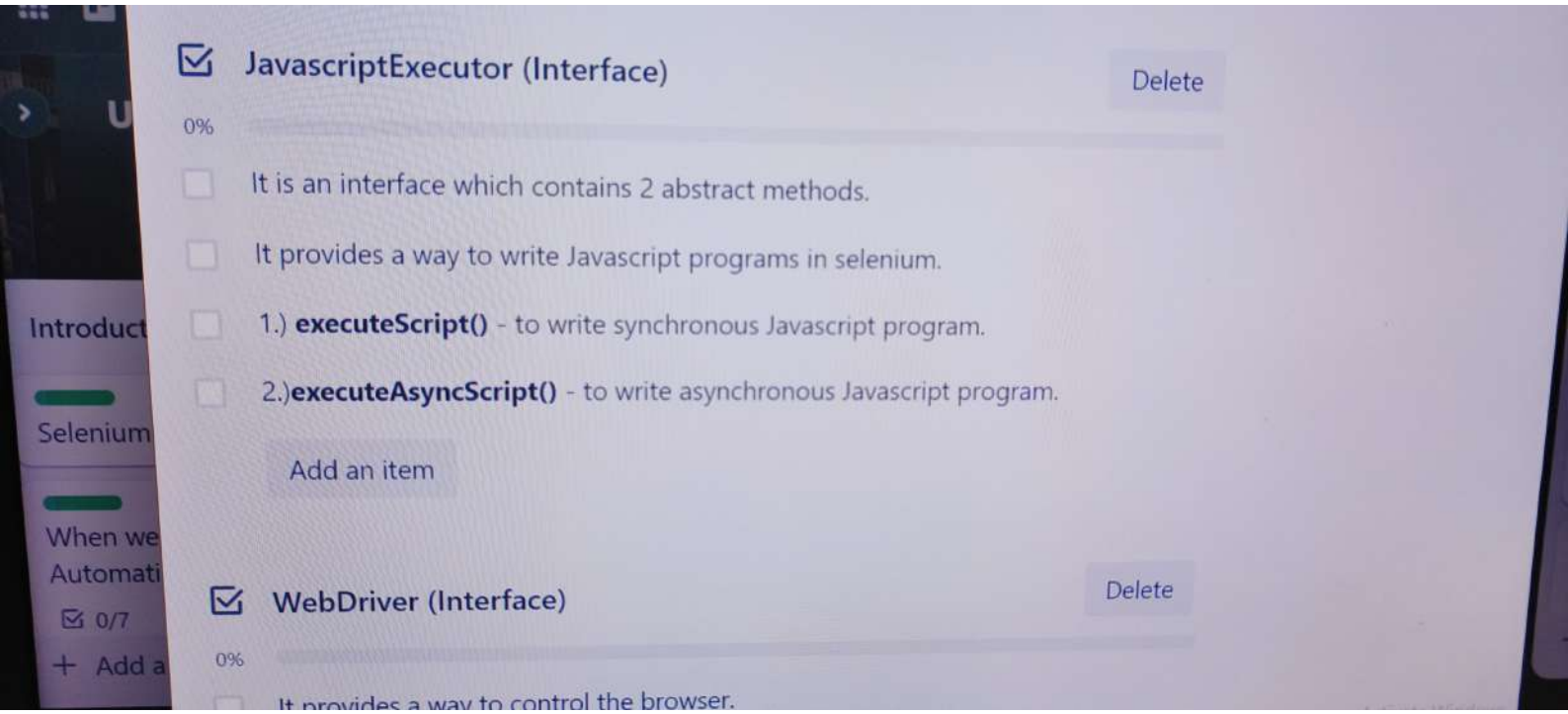
Introduct

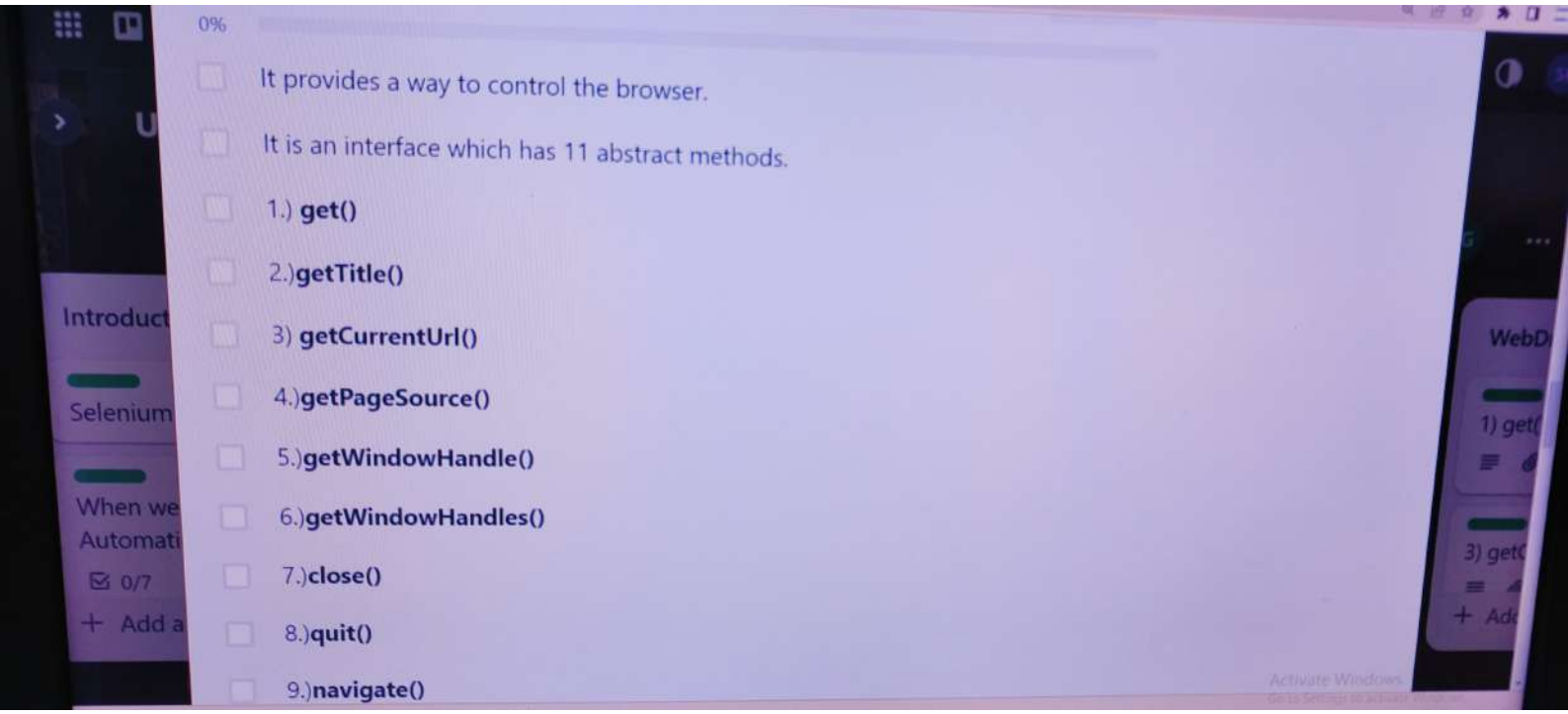
Selenium

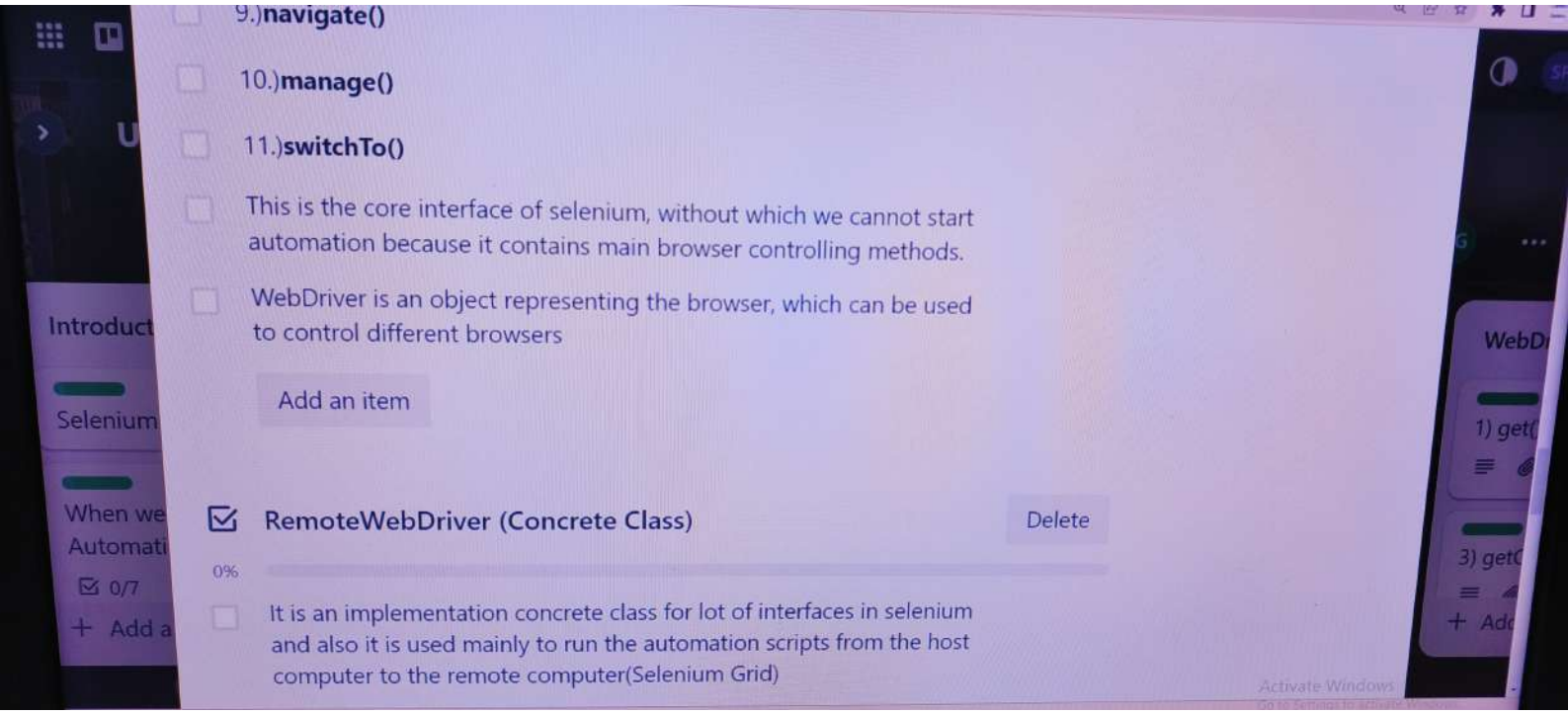
When we
Automati

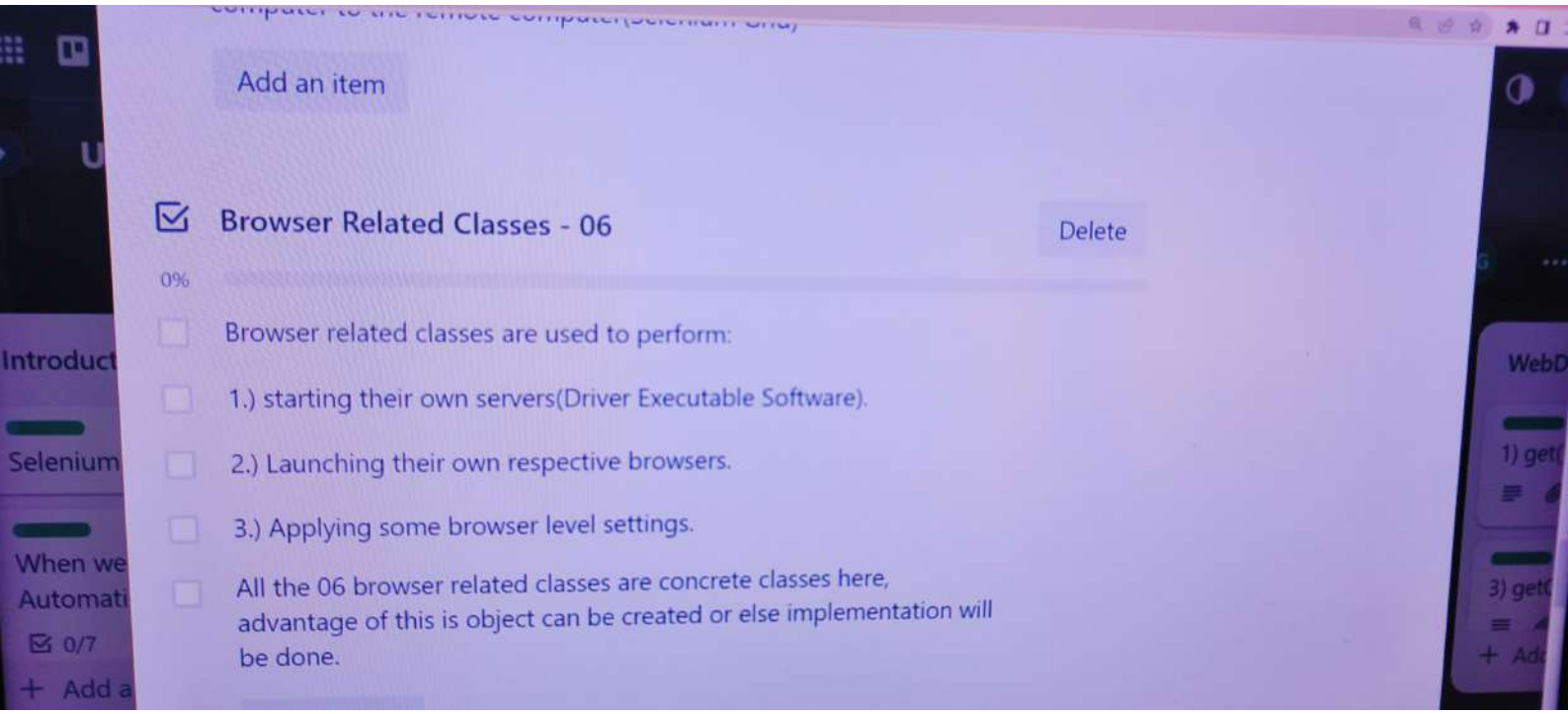
0/7

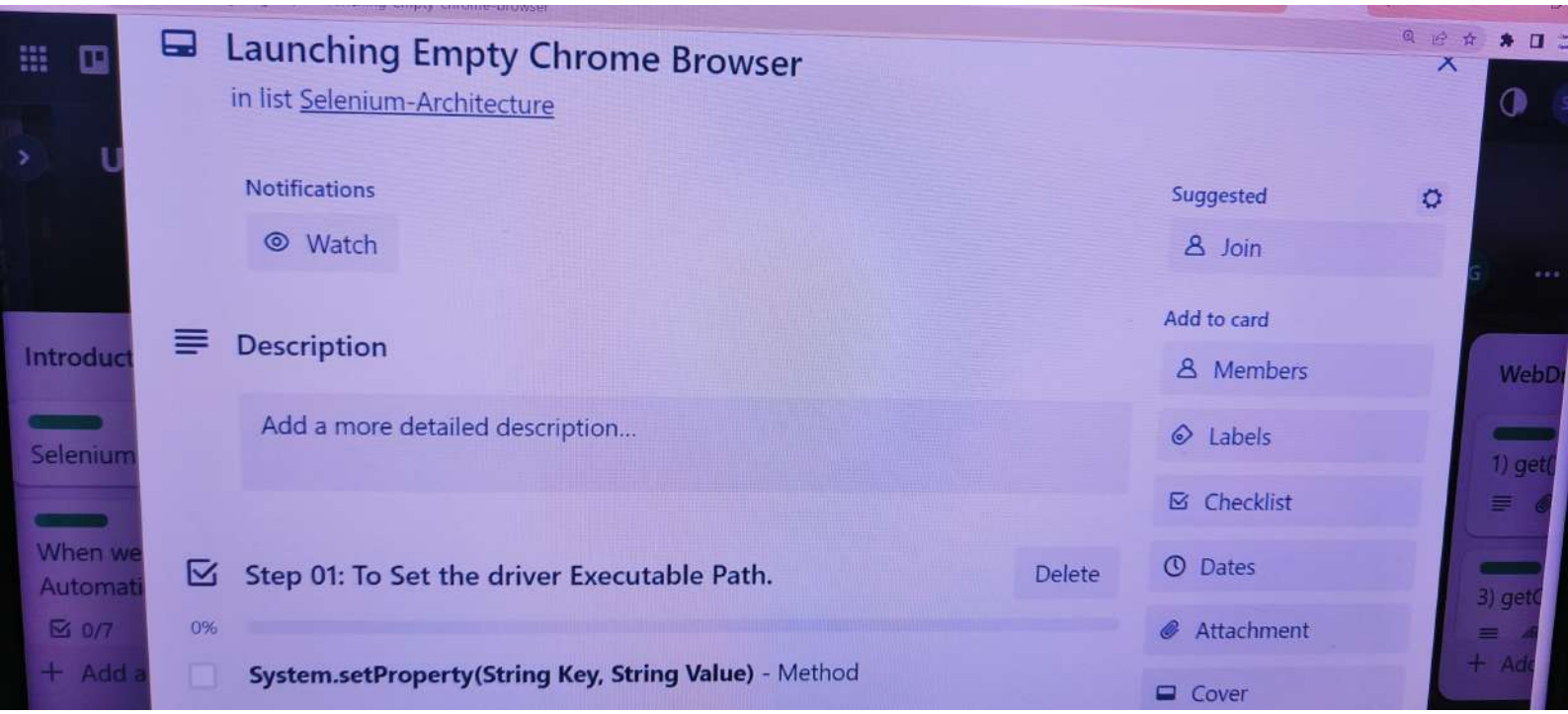
Add a

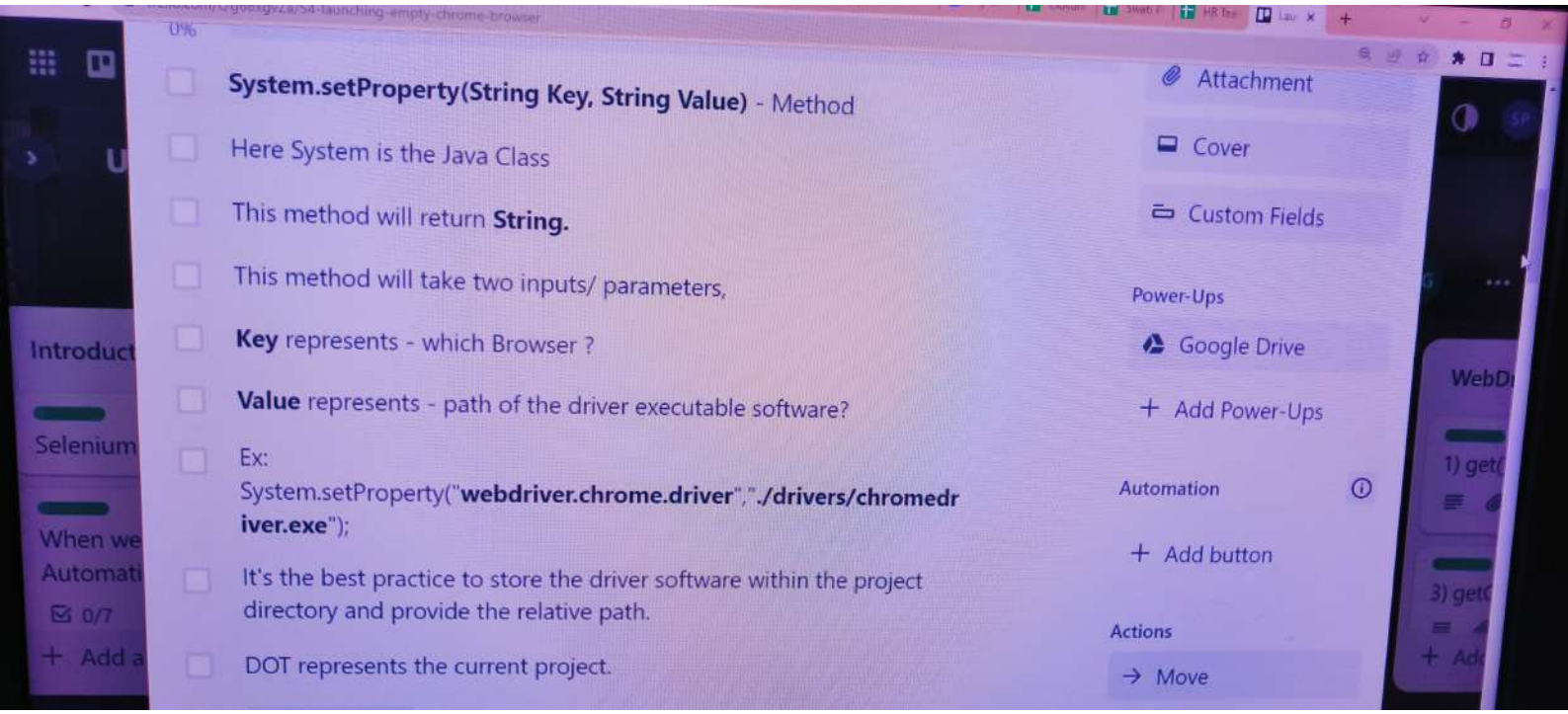

JavascriptExecutor (Interface)
Delete
0%
It is an interface which contains 2 abstract methods.
It provides a way to write Javascript programs in selenium.
1.) executeScript() - to write synchronous Javascript program.
2.)executeAsyncScript() - to write asynchronous Javascript program.
Add an item
WebDriver (Interface)
Delete
0%
Introduct
Selenium
When we
Automati
0/7
Add a

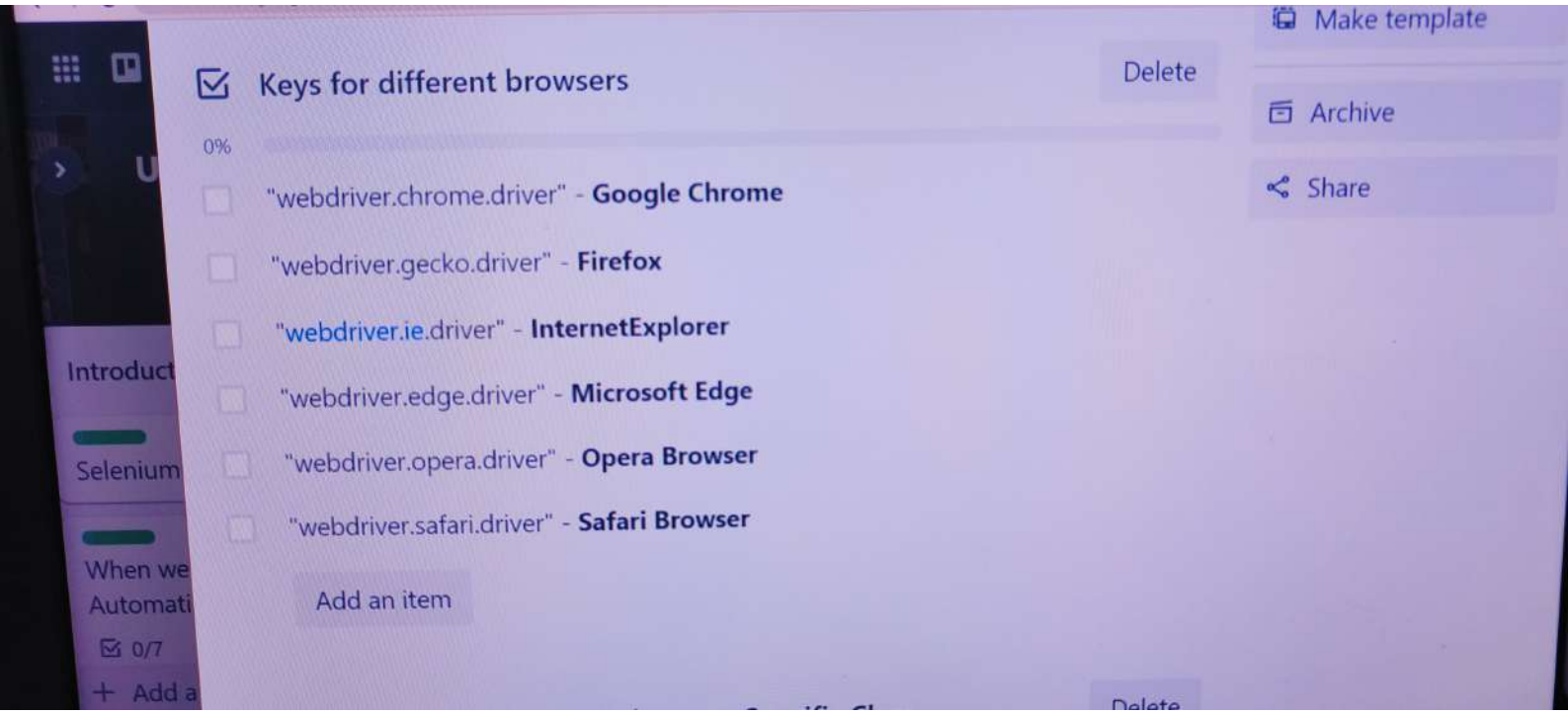

0%
It provides a way to control the browser.
It is an interface which has 11 abstract methods.
1.) get()
2.)getTitle()
3) getCurrentUrl()
4.)getPageSource()
5.)getWindowHandle()
6.)getWindowHandles()
7.)close()
8.)quit()
9.)navigate()

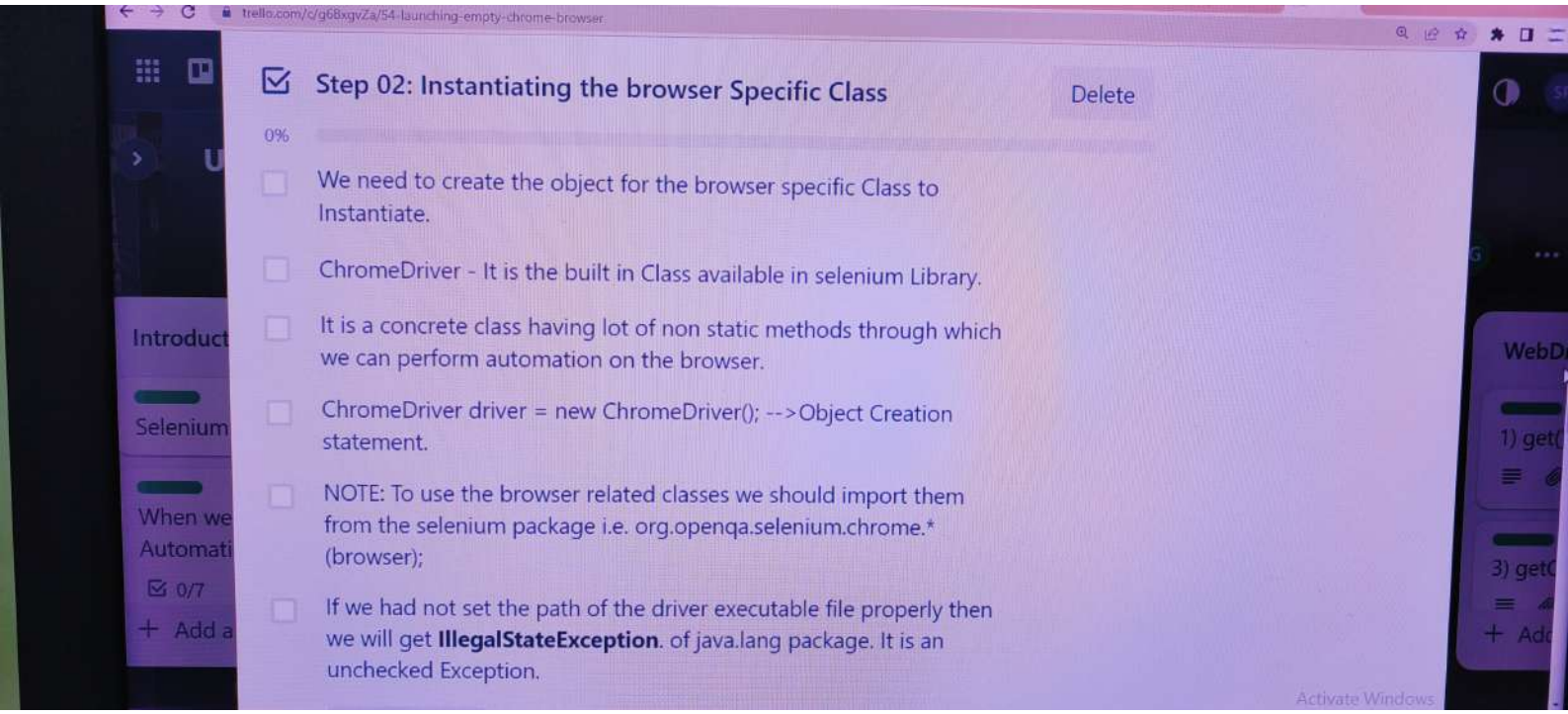

9.)navigate()
10.)manage()
11.)switchTo()
This is the core interface of selenium, without which we cannot start automation because it contains main browser controlling methods.
WebDriver is an object representing the browser, which can be used to control different browsers
Add an item
RemoteWebDriver (Concrete Class)
Delete
0%
It is an implementation concrete class for lot of interfaces in selenium and also it is used mainly to run the automation scripts from the host computer to the remote computer(Selenium Grid)
Introduct
Selenium
When we
Automati
0/7
Add a
WebD
1) get(
3) get(
Add
Activate Windows


Add an item
Browser Related Classes - 06
Delete
0%
Browser related classes are used to perform:
1.) starting their own servers(Driver Executable Software).
2.) Launching their own respective browsers.
3.) Applying some browser level settings.
All the 06 browser related classes are concrete classes here, advantage of this is object can be created or else implementation will be done.


Launching Empty Chrome Browser
in list Selenium-Architecture
Notifications
Watch
Suggested
Join
Description
Add a more detailed description...
Add to card
Members
Labels
Checklist
Dates
Attachment
Cover
Step 01: To Set the driver Executable Path.
Delete
0%
System.setProperty(String Key, String Value) - Method


0%
System.setProperty(String Key, String Value) - Method
Here System is the Java Class
This method will return String.
This method will take two inputs/ parameters,
Key represents - which Browser ?
Value represents - path of the driver executable software?
Ex:
System.setProperty("webdriver.chrome.driver","./drivers/chromedriver.exe");
It's the best practice to store the driver software within the project directory and provide the relative path.
DOT represents the current project.
Attachment
Cover
Custom Fields
Power-Ups
Google Drive
Add Power-Ups
Automation
Add button
Actions
Move


Make template
Keys for different browsers
Delete
Archive
0%
Share
"webdriver.chrome.driver" - Google Chrome
"webdriver.gecko.driver" - Firefox
"webdriver.ie.driver" - InternetExplorer
"webdriver.edge.driver" - Microsoft Edge
"webdriver.opera.driver" - Opera Browser
"webdriver.safari.driver" - Safari Browser
Add an item
Introduct
Selenium
When we
Automati
0/7
Add a
Delete


trello.com/c/g68xgvZa/54-launching-empty-chrome-browser
Step 02: Instantiating the browser Specific Class
Delete
0%
We need to create the object for the browser specific Class to Instantiate.
ChromeDriver - It is the built in Class available in selenium Library.
It is a concrete class having lot of non static methods through which we can perform automation on the browser.
ChromeDriver driver = new ChromeDriver(); -->Object Creation statement.
NOTE: To use the browser related classes we should import them from the selenium package i.e. org.openqa.selenium.chrome.* (browser);
If we had not set the path of the driver executable file properly then we will get IllegalStateException. of java.lang package. It is an unchecked Exception.
Activate Windows