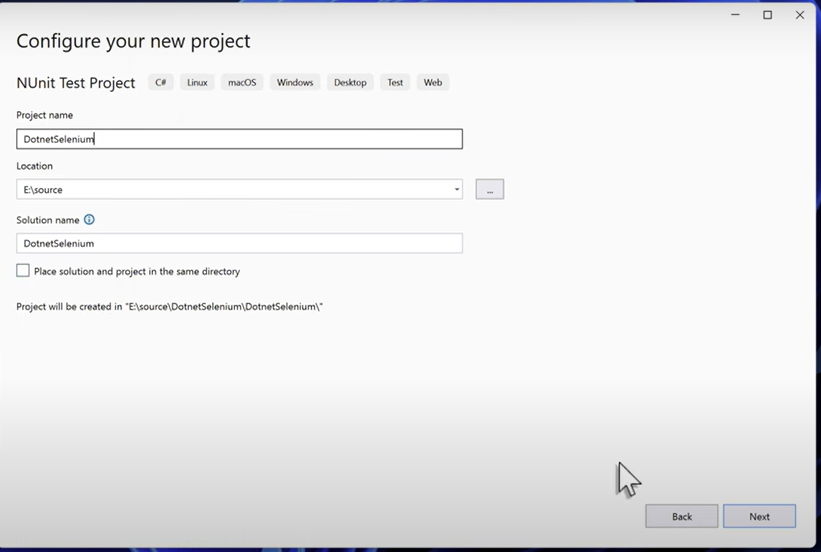
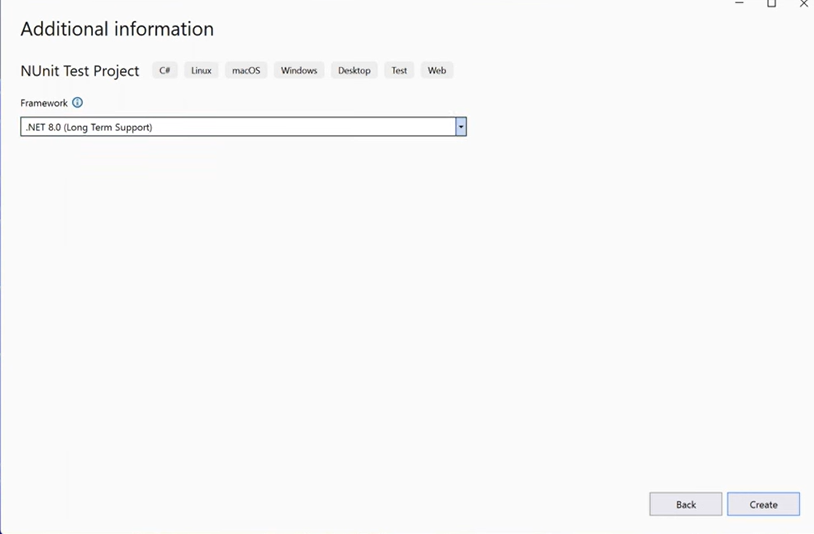


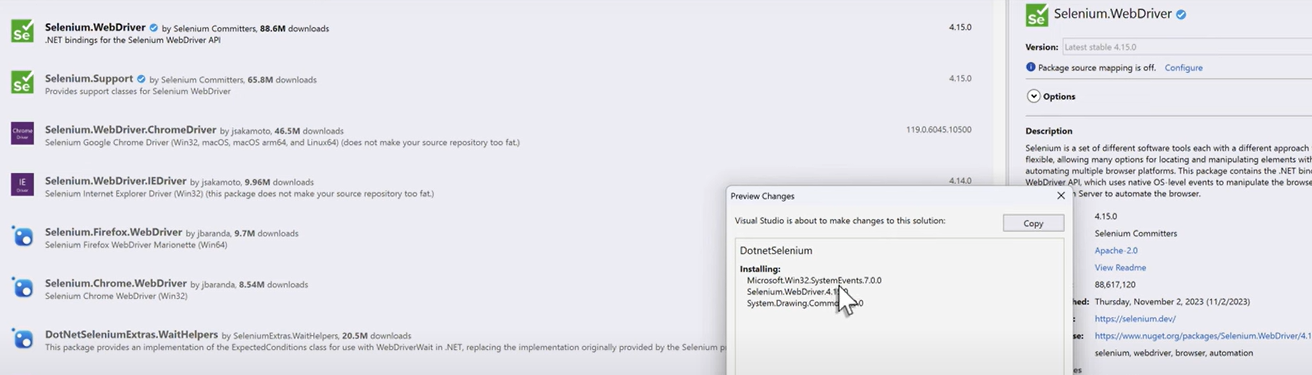
Vs 2022 🡺 Create new NunitTest Project

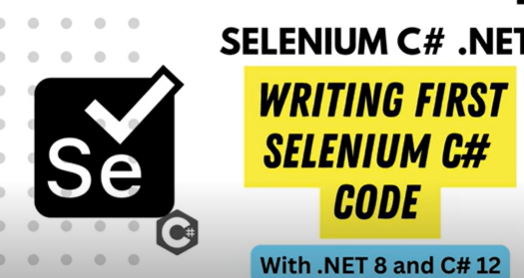




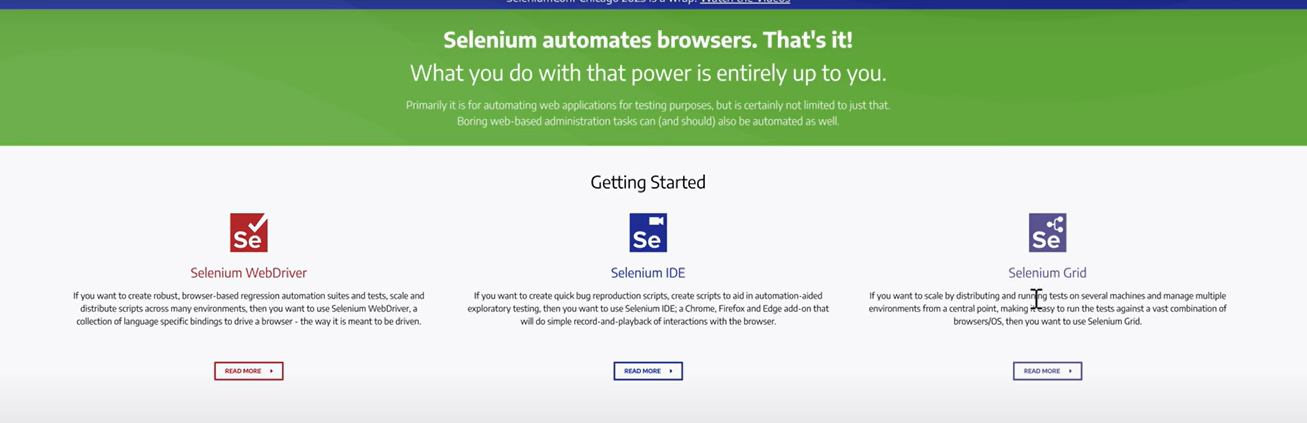


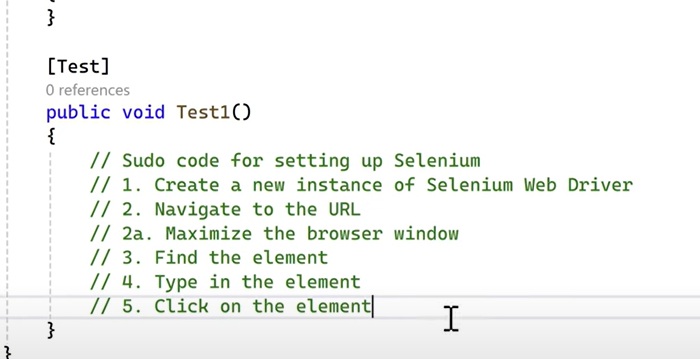
Manage nuget package 🡺Selenium.WebDriver





Selenium.dev





using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

namespace DotnetSeleniumTest

{

public class Tests

{

[SetUp]

public void Setup()

{

}

[Test]

public void Test1()

{

// 1. create a new instance of Selenium Web Driver

IWebDriver driver = new ChromeDriver();

// 2.Navigate to URL

driver.Navigate().GoToUrl("https://www.google.com/");

// 3. Maximize the browser window

driver.Manage().Window.Maximize();

// 4. Find the Element

IWebElement webElement = driver.FindElement(By.Name("q"));

// 5. Type in element

webElement.SendKeys("Selenium");

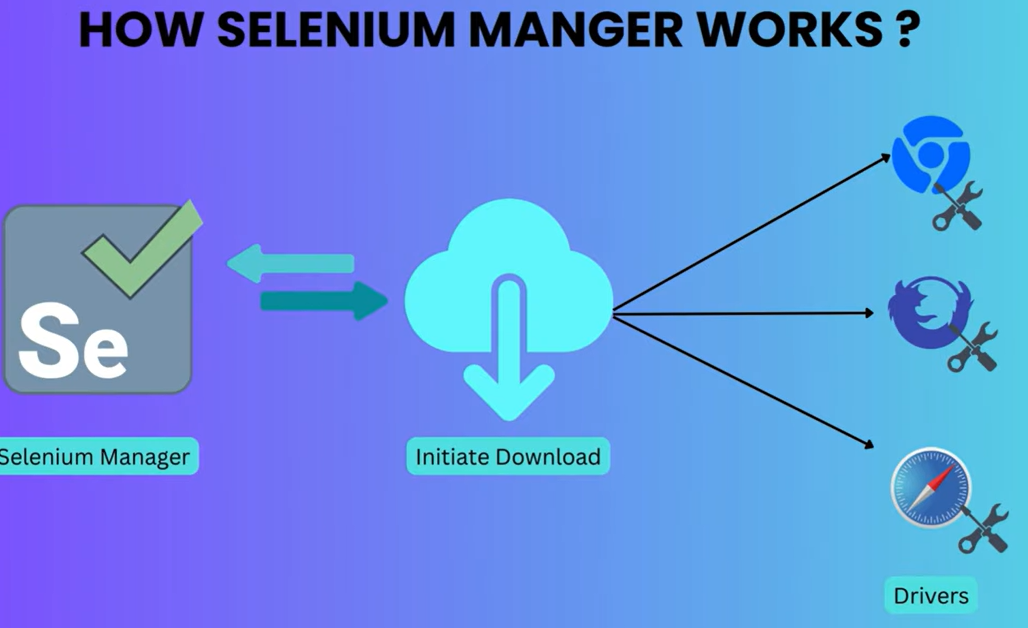
// 6.click on the element

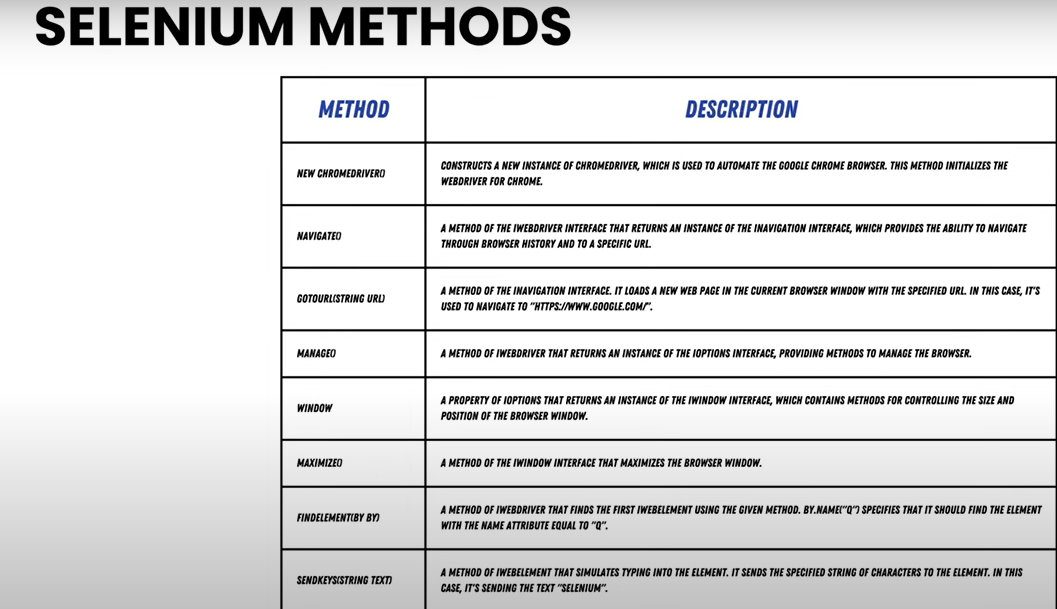
webElement.SendKeys(Keys.Return);

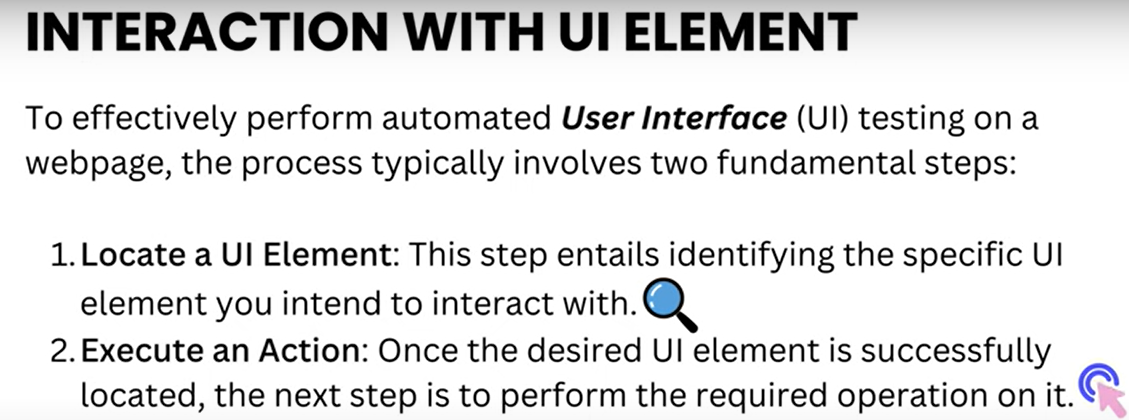
}

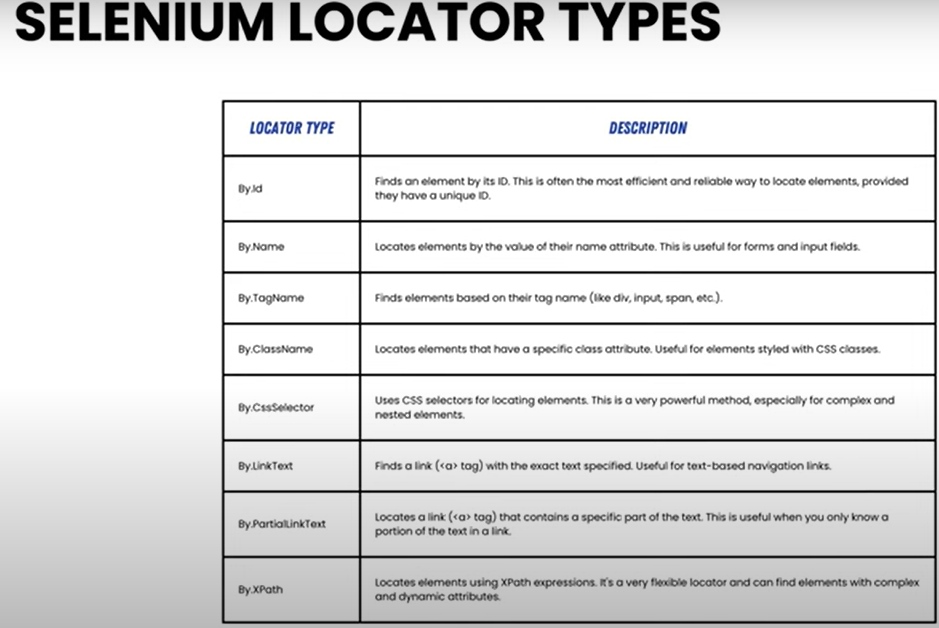
}

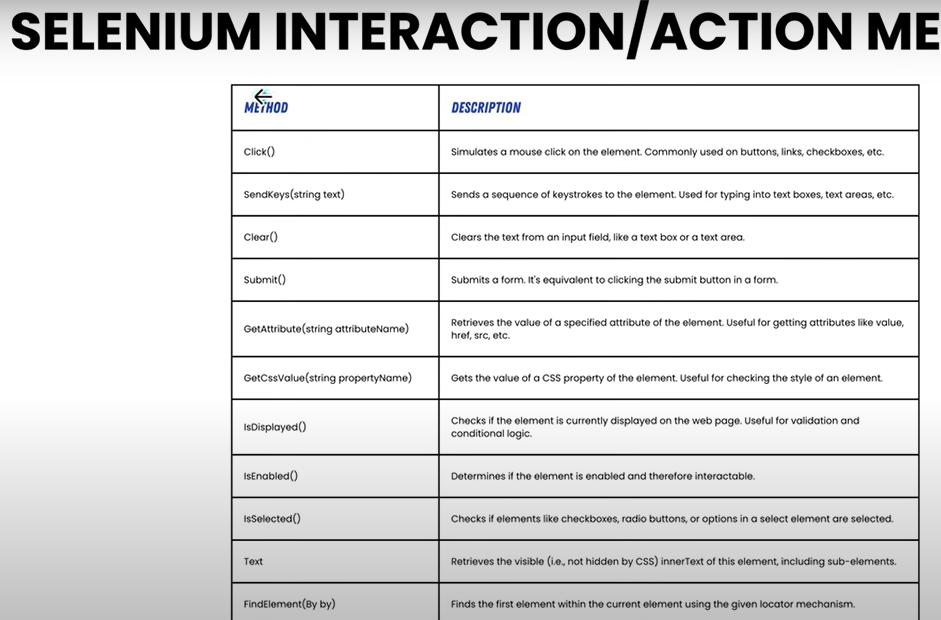
}



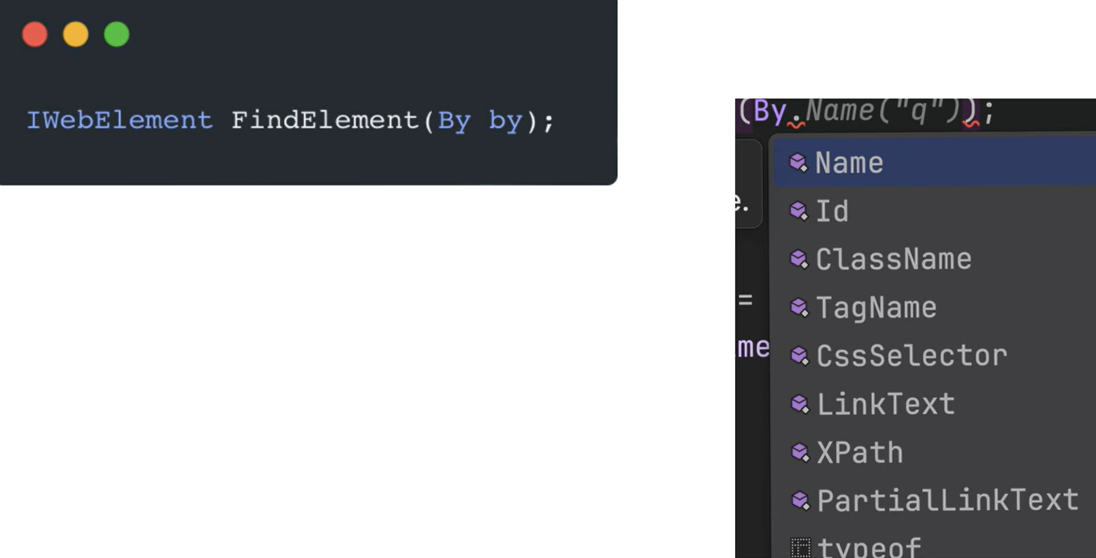








**Working with Locators in Selenium with C# .NET**



<http://eaapp.somee.com/>

[Test]

public void EAWebsiteTest()

{

IWebDriver driver=new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

//Find the Login Link

//IWebElement loginLink = driver.FindElement(By.LinkText("Login"));

IWebElement loginLink = driver.FindElement(By.Id("loginLink"));

loginLink.Click();

// find the userName textbox

IWebElement txtUserName=driver.FindElement(By.Id("UserName"));

// typing on the txtUserName

txtUserName.SendKeys("admin");

IWebElement txtPassword = driver.FindElement(By.Id("Password"));

// typing on the txtPassword

txtPassword.SendKeys("password");

IWebElement btnLogin = driver.FindElement(By.Id("loginIn"));

// click on Login button

btnLogin.Click();

}

[Test]

public void EAWebsiteTest()

{

IWebDriver driver=new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

//Find the Login Link

//IWebElement loginLink = driver.FindElement(By.LinkText("Login"));

IWebElement loginLink = driver.FindElement(By.Id("loginLink"));

loginLink.Click();

// find the userName textbox

IWebElement txtUserName=driver.FindElement(By.Id("UserName"));

// typing on the txtUserName

txtUserName.SendKeys("admin");

IWebElement txtPassword = driver.FindElement(By.Id("Password"));

// typing on the txtUserName

txtPassword.SendKeys("password");

//IWebElement btnLogin = driver.FindElement(By.Id("loginIn"));

//// typing on the txtUserName

//// btnLogin.Click();

IWebElement btnLogin1 = driver.FindElement(By.ClassName(".btn"));

//IWebElement btnLogin1 = driver.FindElement(By.ClassName("btn"));

//btnLogin1.Submit();

IWebElement btnLogin2 = driver.FindElement(By.CssSelector(".btn"));

btnLogin2.Submit();

}

Now let us achieve the same using less lines of code

[Test]

public void EAwebsiteTestReduceSizeCode()

{

IWebDriver driver=new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

driver.FindElement(By.Id("loginLink")).Click();

driver.FindElement(By.Id("UserName")).SendKeys("admin");

driver.FindElement(By.Id("Password")).SendKeys("password");

driver.FindElement(By.CssSelector(".btn")).Submit();

}

**Interacting with DropDown and MultiSelect Option using Selenium**

Create testpage.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Employee Registration Form</title>

    <style>

      body {

        font-family: Arial, sans-serif;

        background-color: #f4f4f4;

        margin: 0;

        padding: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

      }

      .registration-form {

        background-color: #fff;

        padding: 20px;

        border-radius: 8px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        width: 300px;

      }

      .registration-form h2 {

        text-align: center;

        margin-bottom: 20px;

      }

      .registration-form label {

        display: block;

        margin-bottom: 5px;

        font-weight: bold;

      }

      .registration-form input[type="text"],

      .registration-form input[type="email"],

      .registration-form input[type="date"],

      .registration-form select {

        width: 100%;

        padding: 8px;

        margin-bottom: 15px;

        border: 1px solid #ccc;

        border-radius: 4px;

      }

      .registration-form fieldset {

        border: 1px solid #ccc;

        border-radius: 4px;

        padding: 10px;

        margin-bottom: 15px;

      }

      .registration-form fieldset legend {

        font-weight: bold;

        padding: 0 5px;

      }

      .registration-form input[type="radio"] {

        margin-right: 5px;

      }

      .registration-form input[type="checkbox"] {

        margin-right: 5px;

      }

      .registration-form button {

        width: 100%;

        padding: 10px;

        background-color: #28a745;

        color: #fff;

        border: none;

        border-radius: 4px;

        cursor: pointer;

        font-size: 16px;

      }

      .registration-form button:hover {

        background-color: #218838;

      }

    </style>

  </head>

  <body>

    <div class="registration-form">

      <h2>Employee Registration</h2>

      <form>

        <label for="name">Name:</label>

        <input type="text" id="name" name="name" required />

        <label for="email">Email:</label>

        <input type="email" id="email" name="email" required />

        <label for="doj">Date of Joining:</label>

        <input type="date" id="doj" name="doj" required />

        <fieldset>

          <legend>Gender:</legend>

          <label

            ><input type="radio" name="gender" value="male" required />

            Male</label

          >

          <label

            ><input type="radio" name="gender" value="female" /> Female</label

          >

          <label

            ><input type="radio" name="gender" value="other" /> Other</label

          >

        </fieldset>

        <label for="city">City:</label>

        <select id="city" name="city" required>

          <option value="">Select City</option>

          <option value="hyderabad">Hyderabad</option>

          <option value="coimbatore">Coimbatore</option>

          <option value="chennai">Chennai</option>

          <option value="bangalore">Bangalore</option>

        </select>

        <label for="designation">Designation:</label>

        <input type="text" id="designation" name="designation" required />

        <label for="skills">Skills:</label>

        <select id="skills" name="skills" multiple required>

          <option value="testing">Testing</option>

          <option value="java">Java</option>

          <option value=".net">.NET</option>

          <option value="cloud">Cloud</option>

        </select>

        <!-- <fieldset>

          <legend>Skills:</legend>

          <label

            ><input type="checkbox" name="skills" value="testing" />

            Testing</label

          >

          <label

            ><input type="checkbox" name="skills" value="java" /> Java</label

          >

          <label

            ><input type="checkbox" name="skills" value=".net" /> .NET</label

          >

          <label

            ><input type="checkbox" name="skills" value="cloud" /> Cloud</label

          >

        </fieldset> -->

        <button type="submit">Register</button>

      </form>

    </div>

  </body>

</html>

[Test]

public void WorkingWithAdvancedControls()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("file:///C:/Personal/cmr\_capgemini/Testing/Selenium/Testpage.html");

SelectElement selectElement = new SelectElement(driver.FindElement(By.Id("city")));

selectElement.SelectByText("Coimbatore");

SelectElement multiSelect = new SelectElement(driver.FindElement(By.Id("skills")));

multiSelect.SelectByText("Testing");

multiSelect.SelectByText("Cloud");

IList<IWebElement> selectedoptions = multiSelect.AllSelectedOptions;

foreach (IWebElement option in selectedoptions)

{

Console.WriteLine(option.Text);

}

}

**10 .Writing Custom Methods for Selenium UI Actions in Selenium with C# .NET**

[Test]

public void WorkingWithAdvancedControls()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("file:///C:/Personal/cmr\_capgemini/Testing/Selenium/Testpage.html");

SelectElement selectElement = new SelectElement(driver.FindElement(By.Id("city")));

selectElement.SelectByText("Coimbatore");

SelectElement multiSelect = new SelectElement(driver.FindElement(By.Id("skills")));

multiSelect.SelectByText("Testing");

multiSelect.SelectByText("Cloud");

IList<IWebElement> selectedoptions = multiSelect.AllSelectedOptions;

foreach (IWebElement option in selectedoptions)

{

Console.WriteLine(option.Text);

}

IWebElement loginLink = driver.FindElement(By.Id("loginLink"));

loginLink.Click();

}

Add New Class 🡺 SeleniumCustomMethods

Add below code

Public class SeleniumCustomMethods

{

public static void Click(IWebDriver driver,By locator)

{

driver.FindElement(locator).Click();

}

}

Go back to UnitTest.cs

IWebElement loginLink = driver.FindElement(By.Id("loginLink"));

loginLink.Click();

instead of writing 2 lines of code you can do the same in single line

SeleniumCustomMethods.Click(driver, By.Id("loginLink"));

Now you might have the question instead of writing in same file why should we use another class and method. Whenever we need perform multiple operation for an example I need to clear(reset myform fields) all textboxes you should repeat the code

public static void EnterText(IWebDriver driver,By locator,string text) {

driver.FindElement(locator).Clear(); // clear the text from textbox

driver.FindElement(locator).SendKeys(text);

}

[Test]

public void EAwebsiteTestReduceSizeCode()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

SeleniumCustomMethods.Click(driver, By.Id("loginLink"));

// driver.FindElement(By.Id("UserName")).SendKeys("admin");

SeleniumCustomMethods.EnterText(driver, By.Id("UserName"), "admin");

// driver.FindElement(By.Id("Password")).SendKeys("password");

SeleniumCustomMethods.EnterText(driver, By.Id("Password"), "password");

driver.FindElement(By.CssSelector(".btn")).Submit();

}

Let us add custom method for DropdownSelect

public static void SelectDropDownByText (IWebDriver driver,By locator,string text)

{

SelectElement selectElement=new SelectElement(driver.FindElement(locator));

selectElement.SelectByText(text);

}

public static void SelectDropDownByValue(IWebDriver driver, By locator, string value)

{

SelectElement selectElement = new SelectElement(driver.FindElement(locator));

selectElement.SelectByValue(value);

}

UnitTest1.cs

[Test]

public void WorkingWithAdvancedControls()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("file:///C:/Personal/cmr\_capgemini/Testing/Selenium/Testpage.html");

//SelectElement selectElement = new SelectElement(driver.FindElement(By.Id("city")));

//selectElement.SelectByText("Coimbatore");

SeleniumCustomMethods.SelectDropDownByText(driver,By.Id("city"), "Coimbatore");

SelectElement multiSelect = new SelectElement(driver.FindElement(By.Id("skills")));

multiSelect.SelectByText("Testing");

multiSelect.SelectByText("Cloud");

IList<IWebElement> selectedoptions = multiSelect.AllSelectedOptions;

foreach (IWebElement option in selectedoptions)

{

Console.WriteLine(option.Text);

}

SeleniumCustomMethods.Click(driver, By.Id("loginLink"));

}

**Writing Custom Methods for Selenium custom controls in C#**

public static void MultiSelectElement(IWebDriver driver, By locator, string[] values)

{

SelectElement multiSelect = new SelectElement(driver.FindElement(locator));

foreach (var value in values)

{

multiSelect.SelectByValue(value);

}

}

UnitTest1.cs

[Test]

public void WorkingWithAdvancedControls()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("file:///C:/Personal/cmr\_capgemini/Testing/Selenium/Testpage.html");

//SelectElement selectElement = new SelectElement(driver.FindElement(By.Id("city")));

//selectElement.SelectByText("Coimbatore");

SeleniumCustomMethods.SelectDropDownByText(driver,By.Id("city"), "Coimbatore");

//SelectElement multiSelect = new SelectElement(driver.FindElement(By.Id("skills")));

//multiSelect.SelectByText("Testing");

//multiSelect.SelectByText("Cloud");

//IList<IWebElement> selectedoptions = multiSelect.AllSelectedOptions;

//foreach (IWebElement option in selectedoptions)

//{

// Console.WriteLine(option.Text);

//}

SeleniumCustomMethods.MultiSelectElements(driver, By.Id("skills"), ["testing", "cloud"]);

var getSelectedOptions = SeleniumCustomMethods.GetAllSelectedLists(driver, By.Id("skills"));

//foreach (var option in getSelectedOptions)

//{

// Console.WriteLine(option);

//}

getSelectedOptions.ForEach(Console.WriteLine);

}

Next will check the whether it is loading the next page or not and verify anyone of the content in next page shows as we expects (like image)

[Test]

public void EAWebsiteTest()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

IWebElement loginLink = driver.FindElement(By.Id("loginLink"));

loginLink.Click();

IWebElement txtUserName = driver.FindElement(By.Id("UserName"));

txtUserName.SendKeys("admin");

IWebElement txtPassword = driver.FindElement(By.Id("Password"));

txtPassword.SendKeys("password");

IWebElement btnLogin2 = driver.FindElement(By.CssSelector(".btn"));

btnLogin2.Submit();

// Wait for the page to load

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementExists(By.TagName("img")));

// Verify the image is present

try

{

IWebElement imageElement = driver.FindElement(By.CssSelector("img[src='/Image/EA\_banner\_white\_v1.jpg']"));

Assert.IsTrue(imageElement.Displayed, "Image is displayed on the page.");

Console.WriteLine("Image is present on the page.");

}

catch (NoSuchElementException)

{

Assert.Fail("Image not found on the page.");

}

driver.Quit();

}

Extend the Form handling test

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Employee Registration Form</title>

    <style>

      body {

        font-family: Arial, sans-serif;

        background-color: #f4f4f4;

        margin: 0;

        padding: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

      }

      .registration-form {

        background-color: #fff;

        padding: 20px;

        border-radius: 8px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        width: 300px;

      }

      .registration-form h2 {

        text-align: center;

        margin-bottom: 20px;

      }

      .registration-form label {

        display: block;

        margin-bottom: 5px;

        font-weight: bold;

      }

      .registration-form input[type="text"],

      .registration-form input[type="email"],

      .registration-form input[type="date"],

      .registration-form select {

        width: 100%;

        padding: 8px;

        margin-bottom: 15px;

        border: 1px solid #ccc;

        border-radius: 4px;

      }

      .registration-form fieldset {

        border: 1px solid #ccc;

        border-radius: 4px;

        padding: 10px;

        margin-bottom: 15px;

      }

      .registration-form fieldset legend {

        font-weight: bold;

        padding: 0 5px;

      }

      .registration-form input[type="radio"] {

        margin-right: 5px;

      }

      .registration-form input[type="checkbox"] {

        margin-right: 5px;

      }

      .registration-form button {

        width: 100%;

        padding: 10px;

        background-color: #28a745;

        color: #fff;

        border: none;

        border-radius: 4px;

        cursor: pointer;

        font-size: 16px;

      }

      .registration-form button:hover {

        background-color: #218838;

      }

    </style>

    <script>

      function submitForm(event) {

        event.preventDefault(); // Prevent actual form submission

        // Hide the form and show a success message

        document.getElementById("registration-form").style.display = "none";

        document.getElementById("success-message").style.display = "block";

      }

    </script>

  </head>

  <body>

    <div class="registration-form">

      <h2>Employee Registration</h2>

      <form id="registration-form" onsubmit="submitForm(event)">

        <label for="name">Name:</label>

        <input type="text" id="name" name="name" required />

        <label for="email">Email:</label>

        <input type="email" id="email" name="email" required />

        <label for="doj">Date of Joining:</label>

        <input type="date" id="doj" name="doj" required />

        <fieldset>

          <legend>Gender:</legend>

          <label

            ><input type="radio" name="gender" value="male" required />

            Male</label

          >

          <label

            ><input type="radio" name="gender" value="female" /> Female</label

          >

          <label

            ><input type="radio" name="gender" value="other" /> Other</label

          >

        </fieldset>

        <label for="city">City:</label>

        <select id="city" name="city" required>

          <option value="">Select City</option>

          <option value="hyderabad">Hyderabad</option>

          <option value="coimbatore">Coimbatore</option>

          <option value="chennai">Chennai</option>

          <option value="bangalore">Bangalore</option>

        </select>

        <label for="designation">Designation:</label>

        <input type="text" id="designation" name="designation" required />

        <label for="skills">Skills:</label>

        <select id="skills" name="skills" multiple required>

          <option value="testing">Testing</option>

          <option value="java">Java</option>

          <option value=".net">.NET</option>

          <option value="cloud">Cloud</option>

        </select>

        <fieldset>

          <legend>Hobbies:</legend>

          <label

            ><input

              type="checkbox"

              name="hobbies"

              value="Reading Books"

            />Reading Books</label

          >

          <label

            ><input

              type="checkbox"

              name="hobbies"

              value="Playing Chess"

            />Playing Chess</label

          >

          <label

            ><input

              type="checkbox"

              name="hobbies"

              value="Dancing"

            />Dancing</label

          >

          <label

            ><input

              type="checkbox"

              name="hobbies"

              value="Playing Baseball"

            />Playing Baseball</label

          >

        </fieldset>

        <button type="submit">Register</button>

      </form>

      <div id="success-message" class="success-message">

        🎉 Thank you for registering! 🎉

      </div>

    </div>

  </body>

</html>

public static List<string> GetAllCheckedCheckboxes(IWebDriver driver, By locator)

{

List<string> selectedValues = new List<string>();

var checkboxes = driver.FindElements(locator);

foreach (var checkbox in checkboxes)

{

if (checkbox.Selected) // If the checkbox is checked

{

selectedValues.Add(checkbox.GetAttribute("value"));

}

}

return selectedValues;

}

UnitTest1.cs

[Test]

public void WorkingWithAdvancedControls()

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("file:///C:/Personal/cmr\_capgemini/Testing/Selenium/Testpage.html");

SeleniumCustomMethods.EnterText(driver, By.Id("name"), "Geetha Eswaramurthi");

SeleniumCustomMethods.EnterText(driver, By.Id("email"), "geethaeswaramurthi@outlook.com");

SeleniumCustomMethods.EnterText(driver, By.Id("doj"), "8-3-2020");

SeleniumCustomMethods.Click(driver, By.CssSelector("input[name='gender'][value='female']"));

SeleniumCustomMethods.SelectDropDownByText(driver, By.Id("city"), "Coimbatore");

SeleniumCustomMethods.EnterText(driver, By.Id("designation"), "Senior Product Engineer");

SeleniumCustomMethods.MultiSelectElements(driver, By.Id("skills"), ["testing", "cloud"]);

var getSelectedOptions = SeleniumCustomMethods.GetAllSelectedLists(driver, By.Id("skills"));

getSelectedOptions.ForEach(Console.WriteLine);

SeleniumCustomMethods.Click(driver, By.CssSelector("input[name='hobbies'][value='Reading Books']"));

SeleniumCustomMethods.Click(driver, By.CssSelector("input[name='hobbies'][value='Playing Baseball']"));

var getSelectedHobbies = SeleniumCustomMethods.GetAllCheckedCheckboxes(driver, By.Name("hobbies"));

getSelectedHobbies.ForEach(Console.WriteLine);

// Clicking the Register Button

SeleniumCustomMethods.Click(driver, By.CssSelector("button[type='submit']"));

// Validation: Ensure form submission redirects to a success page or displays a confirmation message

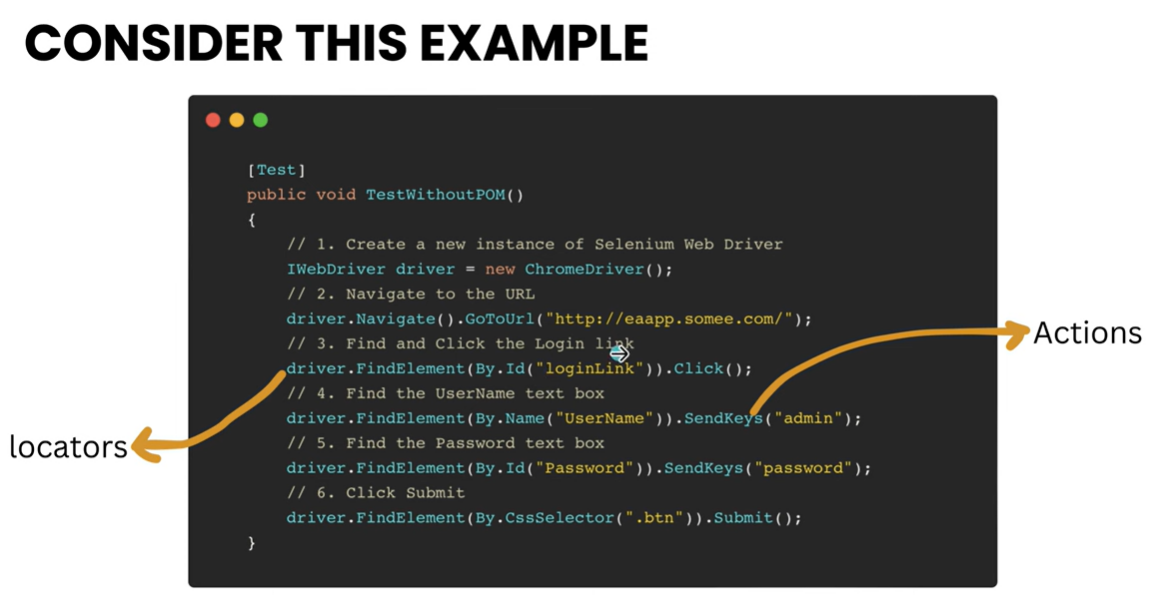
Thread.Sleep(2000); // Wait for form submission response

bool isSubmissionSuccessful = driver.PageSource.Contains("Thank you for registering") || driver.Url.Contains("success");

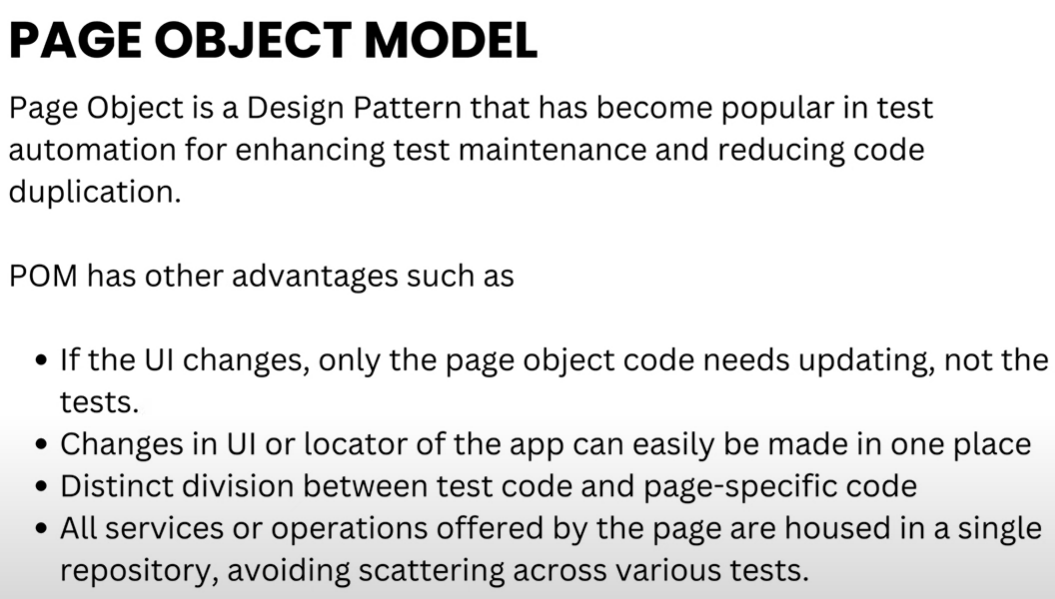
Assert.IsTrue(isSubmissionSuccessful, "Form submission failed.");

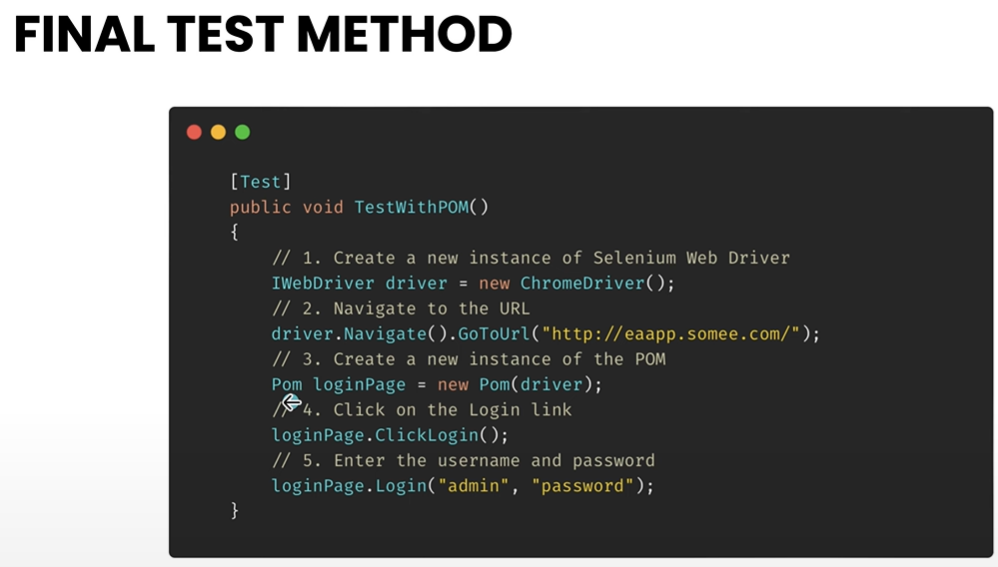
}

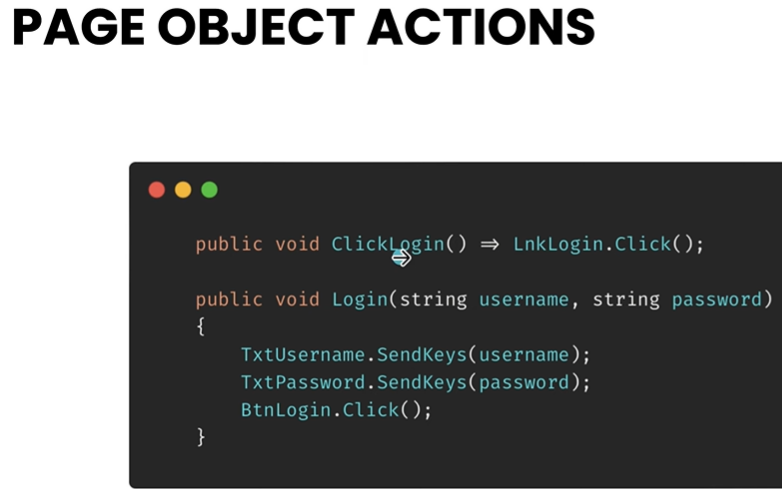
11. Page Object Model











To do the same goto Project

🡺 Add NewFolder

🡺 Pages

Add new class within pages Folder

🡺LoginPage.cs

public class LoginPage

{

private readonly IWebDriver driver;

public LoginPage(IWebDriver driver)

{

this.driver = driver;

}

//locators

//property

IWebElement LoginLink => driver.FindElement(By.Id("loginLink"));

IWebElement TxtUser => driver.FindElement(By.Id("UserName"));

IWebElement TxtPassword => driver.FindElement(By.Id("Password"));

IWebElement BtnLogin => driver.FindElement(By.CssSelector(".btn"));

public void ClickLogin()

{

LoginLink.Click();

}

public void Login(string username, string password)

{

TxtUser.SendKeys(username);

TxtPassword.SendKeys(password);

BtnLogin.Submit();

}

}

Goto UnitTest1.cs

Add new Test Method TestWithPOM() and write the below Method

[Test]

public void TestWithPOM()

{

var driver = new ChromeDriver();

driver.Navigate().GoToUrl("http://eaapp.somee.com/");

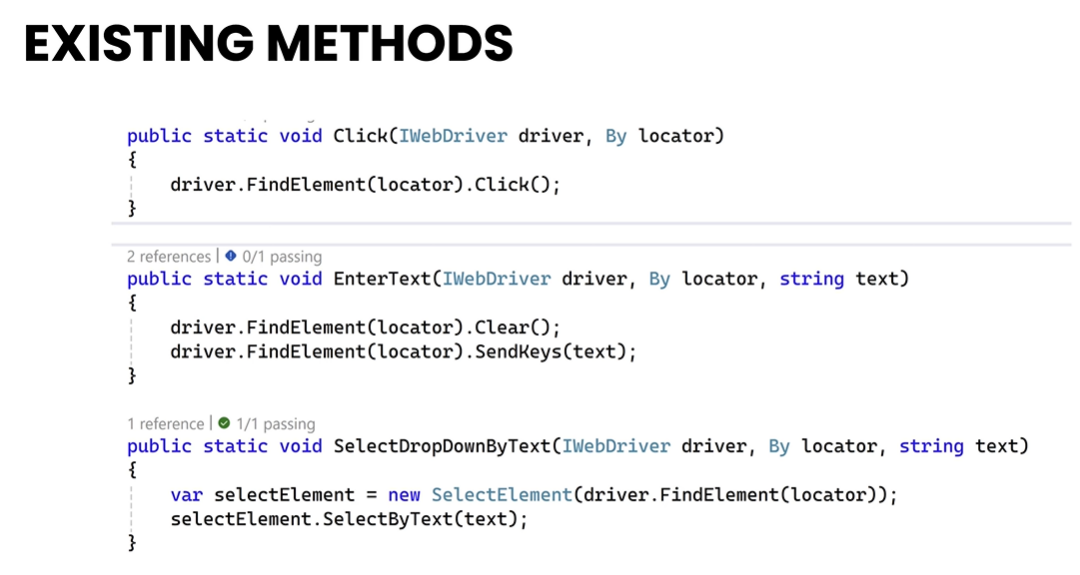
LoginPage loginPage = new LoginPage(driver);

loginPage.ClickLogin();

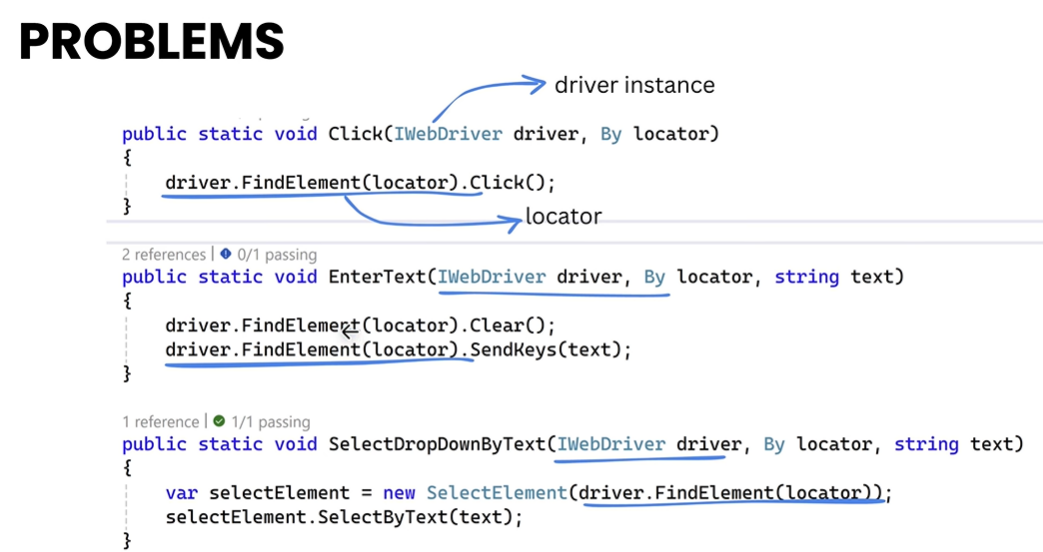
loginPage.Login("admin", "password");

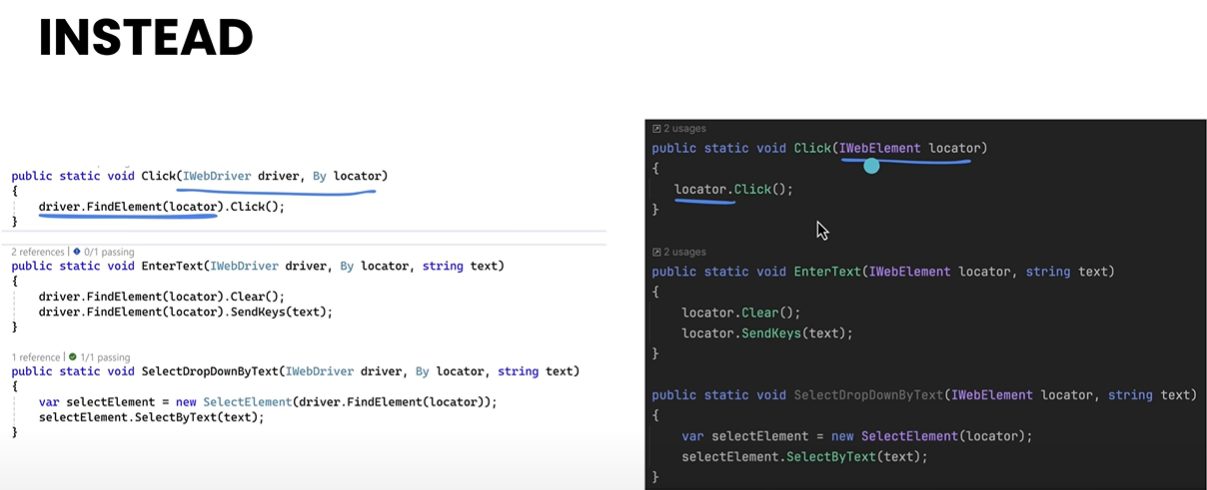
}

**12. Extending Custom Methods with IWebElement in Page Object Model**

****

**Common parameters**





Now Goto SeleniumCustomMethods.cs

//public static void Click(IWebDriver driver,By locator)

//{

// driver.FindElement(locator).Click();

//}

public static void Click(IWebElement locator)

{

locator.Click();

}

public static void Submit(IWebElement locator)

{

locator.Submit();

}

Goto LoginPage.cs

public void ClickLogin()

{

//LoginLink.Click();

SeleniumCustomMethods.Click(LoginLink);

}

public void Login(string username, string password)

{

TxtUser.SendKeys(username);

TxtPassword.SendKeys(password);

//BtnLogin.Submit();

SeleniumCustomMethods.Submit(BtnLogin);

}