

# Automated Test Development with Selenium IDE

## References

- Selenium webpage: <https://www.selenium.dev/selenium-ide/>
- Selenium document: <https://www.selenium.dev/selenium-ide/docs/en/introduction/getting-started>
- Selenium download: <https://www.selenium.dev/downloads/>
- Tutorials:
  - <https://www.tutorialspoint.com/selenium/>
  - <https://expleoacademy.com/int/6-steps-to-create-and-export-selenium-ide-tests/>
  - [https://www.youtube.com/watch?v=m4KpTvEz3vg&ab\\_channel=Simplilearn](https://www.youtube.com/watch?v=m4KpTvEz3vg&ab_channel=Simplilearn)

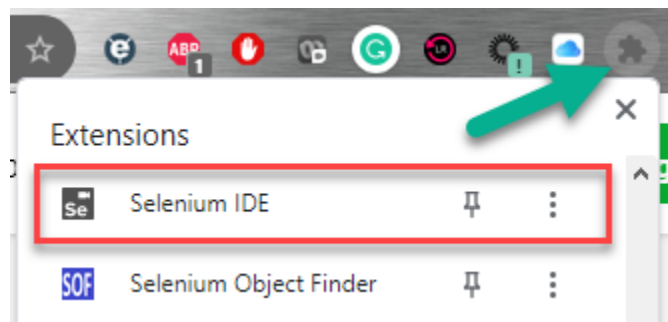
## Software Download

### Download browser plugin

From <https://www.selenium.dev/downloads/>, select download for Chrome. You can also download the plugins for Firefox or Edge or the latest Zip file and install it yourself. In this tutorial, we will just focus on Chrome for the purpose of learning. Other browser implementation may have some variations but with not much deviations from Chrome.

- Download Selenium IDE
- Optional: Download Selenium Object Finder

Now click on the Chrome Extension icon on the right top corner of the browser, you should see the installed extension(s):



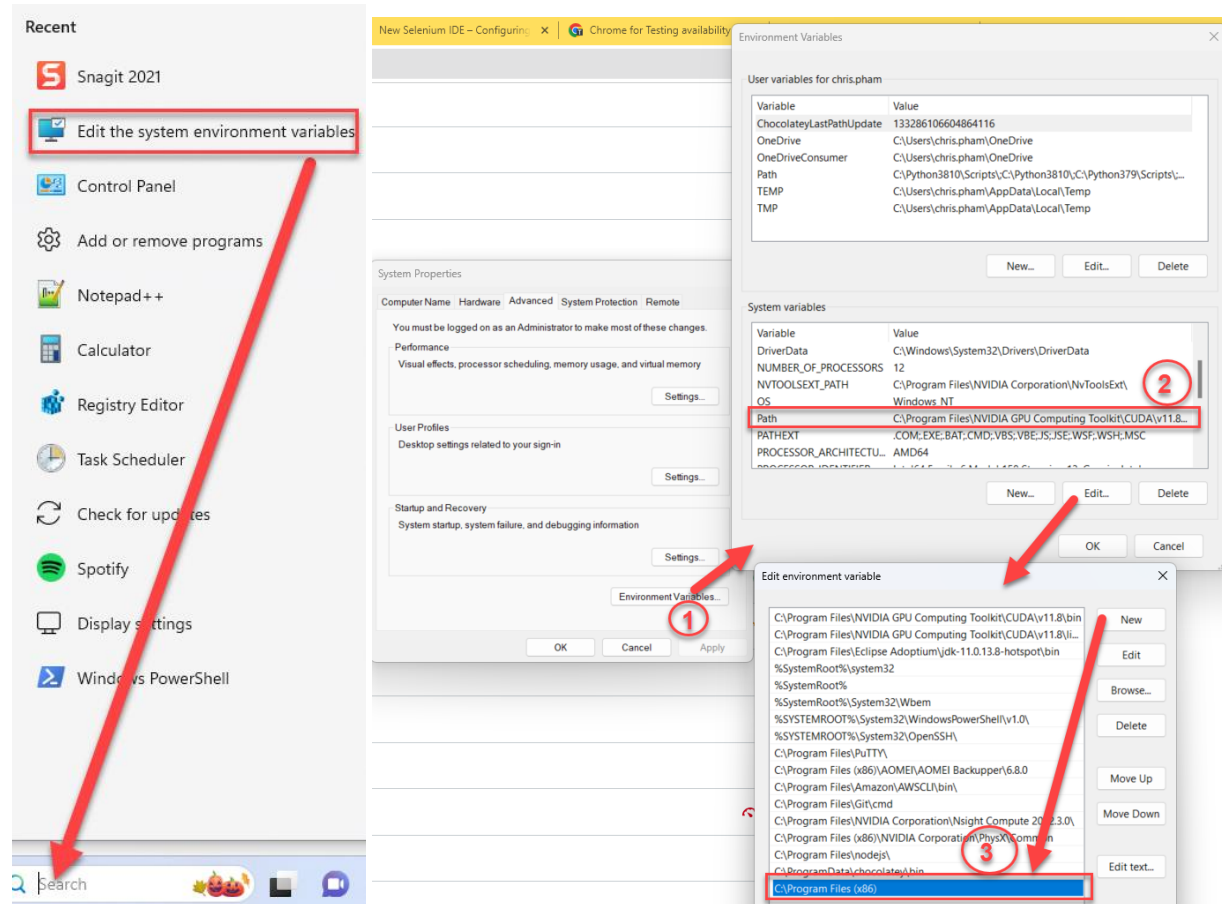
Download the latest WebDriver for your browser

Chrome: <https://chromedriver.chromium.org/downloads>

or [Chrome for Testing availability \(googlechromelabs.github.io\)](https://googlechromelabs.github.io/chrome-for-testing/#stable) or <https://googlechromelabs.github.io/chrome-for-testing/#stable> if you cannot find your Chrome driver on the previous link.

Save the Chrome driver chromedriver.exe to **C:\Program Files (x86)\**

Add **C:\Program Files (x86)\** to Path



## Install Python

- Install Python and add Python path (from python.org)  
Install Python 3.7.9 from <https://www.python.org/downloads/release/python-379/>  
You can use the web-based installer  
Let's do Customized Installation and install it to **C:\Python379** for simplicity
- python.exe -m pip install --upgrade pip
- pip install selenium, pytest, python-dotenv

## Supporting File Creation

### Create .env file

This file is to keep all passwords, license codes or anything else you don't want to show in your python source code.

For example:

```
TEST_PASSWORD = "Password123"
```

```
LICENSE_CODE = "ABC123"
```

### Create a Readme.txt File

This is a text file to capture your instructions and other information about your scripts to give users a perspective of what to expect from your scripts.

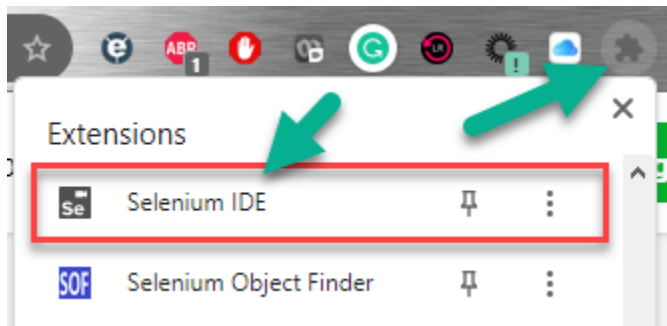
### Create a Test Plan File

This file explains the followings:

- Prerequisites
- A list of tests
- Test details with expected pass/fail conditions and side effects after running that test

### Record a Test

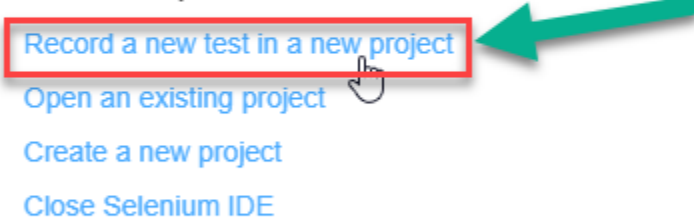
Click on the Chrome Extension icon to show the list of the installed browser extensions, then click on Selenium IDE to invoke the IDE.



You will see the following popup:



What would you like to do?



To learn more on Selenium IDE and how to use it visit the [the Selenium IDE project page](#).

Provide the Project Name and click OK.

## Name your new project

Please provide a name for your new project.

PROJECT NAME

My Automated Test

You can change the name of your project at any time by clicking it and entering a new name.

OK

CANCEL

On the next popup, provide the following website <http://eprint.com.hr/demo/addauser.php> and click START RECORDING to proceed:

## Set your project's base URL

Before you can start recording, you must specify a valid base URL for your project. Your tests will start by navigating to this URL.

BASE URL

<http://eprint.com.hr/demo/addauser.php>

START RECORDING

CANCEL

On the website, you will want to add the followings, but you will let the automation script does that for you:

username: your first name

Password: your own password

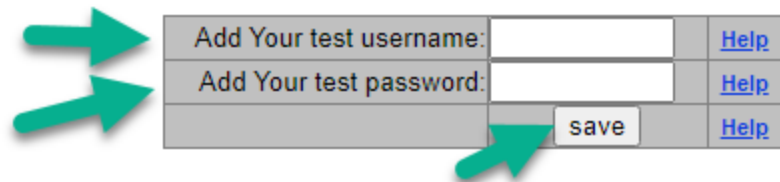
Click "save"

Below is the current single record within the database:

**The username:** admin

**The password:** 123456

Add your own username and password, enter your details below and click save them as you are now.



Add Your test username:	<input type="text"/>	<a href="#">Help</a>
Add Your test password:	<input type="password"/>	<a href="#">Help</a>
<input type="button" value="save"/>		<a href="#">Help</a>

The username: admin1  
The password: 1234

Add your own username and password, enter your details below and click save.  
**DO NOT** use valid details as the information you enter will be displayed above  
and the next visitor will be able to view them as you are now.

Add Your test username:	<input type="text"/>	<a href="#">Help</a>
Add Your test password:	<input type="password"/>	<a href="#">Help</a>
	<input type="button" value="save"/>	<a href="#">Help</a>

When you have added your own username and password move onto the Login page to test it!

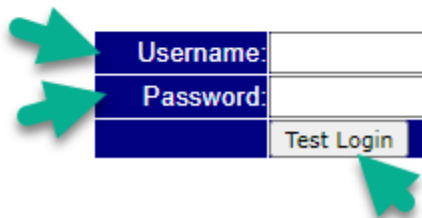
Then click on the link “move onto the Login page to test it!”

## 4. Login

The status was:

**\*\*No login attempted\*\***

Enter your login details you added on the previous page and test the login.  
The success or failure will be shown above.



Username:	<input type="text"/>	<a href="#">Help</a>
Password:	<input type="password"/>	<a href="#">Help</a>
	<input type="button" value="Test Login"/>	<a href="#">Help</a>

Now move on to the final section

Enter your own username and password and press Test Login. If you typed the correct credentials, the Successful Login message will show:

## 4. Login

The status was:

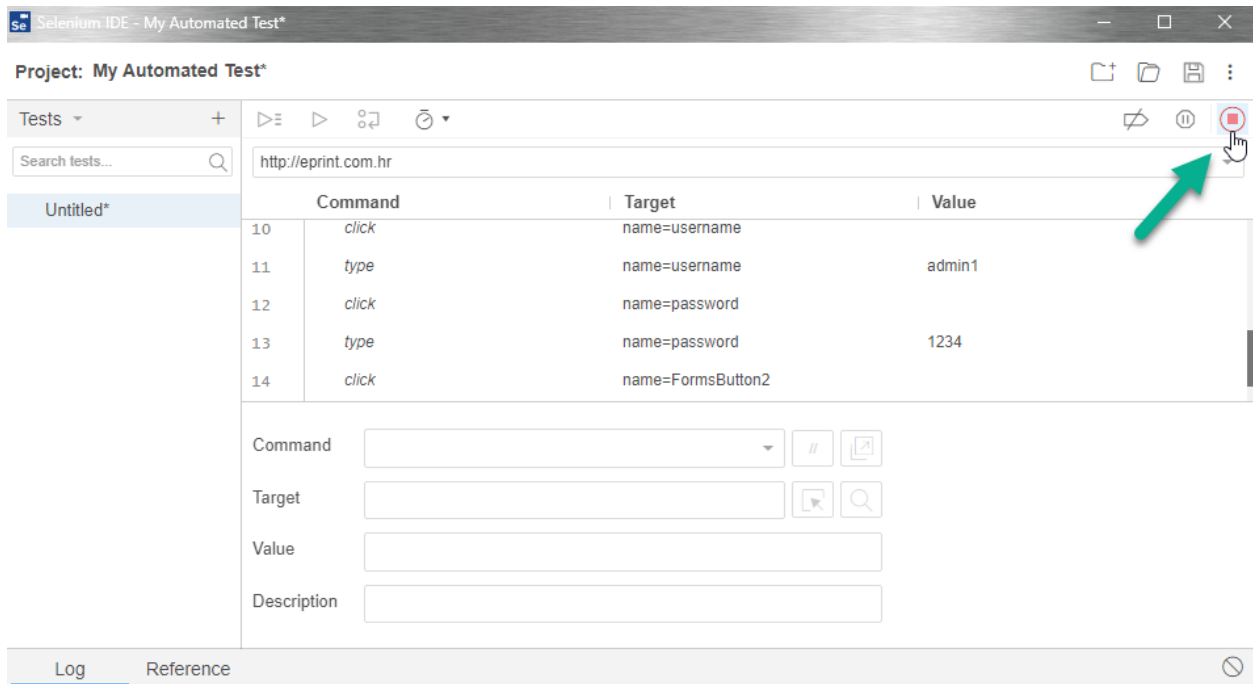
**\*\*Successful Login\*\***

Enter your login details you added on the previous page and test the login.  
The success or failure will be shown above.

Username:	<input type="text"/>	<a href="#">Help</a>
Password:	<input type="password"/>	<a href="#">Help</a>
	<input type="button" value="Test Login"/>	<a href="#">Help</a>

Now move on to the final section

We can stop the recording now:

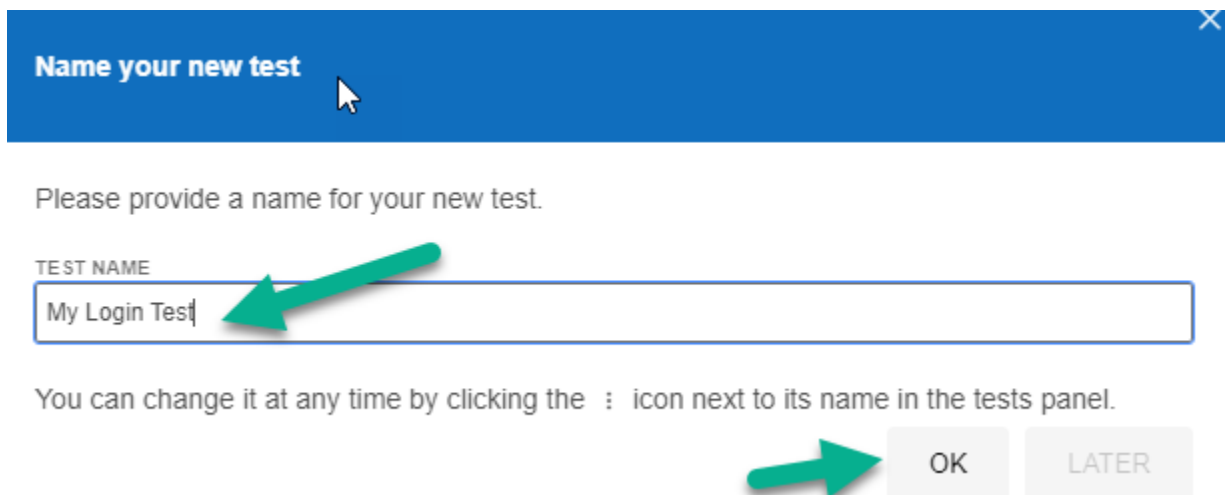


Note: The above window can be behind other windows. You should search for it or click on this icon in

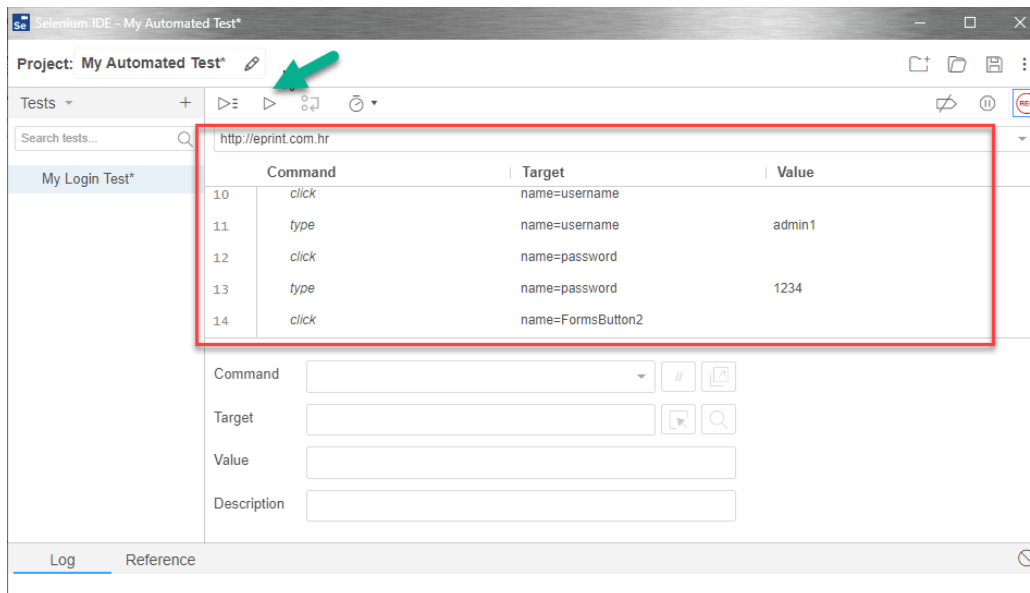
the bottom tray to bring it to the foreground:



Provide a name for your test and click OK:

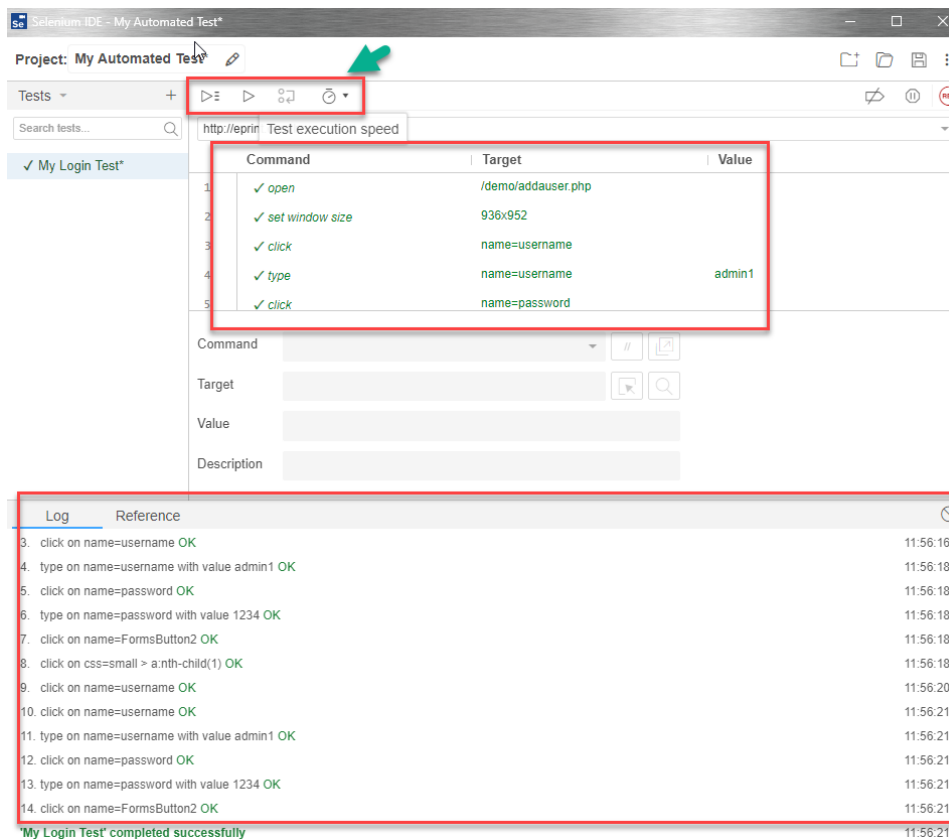


You will see the recorded steps in the Selenium IDE:



You can press the PLAY button to re-play the sequence automatically.

Feel free to change the execution speed and other features of the Selenium IDE to learn more about it.



Save the project as "My Automated Test.side" where SIDE = Selenium IDE.



## Export the Test to Python

Click on the name of the test, then click Export to export your test. Your test will be saved as **test\_myLoginTest.py**.

Project: My Automated Test\*

Tests ▾ +

Search tests... 🔍

✓ My Login Test

Rename  
Duplicate  
Delete  
Export

⏏ ⏩ ⏪ ⏴ ⏵

http://eprint.com.hr

Command

1	✓ open
2	✓ set window size
3	✓ click
4	✓ type
5	✓ click

Command

Target

Value

Select language

☐ C# NUnit

☐ C# xUnit

☐ Java JUnit

☐ JavaScript Mocha

☒ Python pytest

☐ Ruby RSpec

☐ Include origin tracing code comments

☐ Include step description as a separate comment

☐ Export for use on Selenium Grid

EXPORT CANCEL

## Additional Selenium IDE utilities to improve your script

Read by yourself this website to learn more details to improve you're your scripts:

<https://expleoacademy.com/int/6-steps-to-create-and-export-selenium-ide-tests/>

## Modify your exported Python script to include the Chrome driver path

```
# Generated by Selenium IDE
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestMyLoginTest():
    def setup_method(self, method):
        self.driver = webdriver.Chrome(executable_path='C:\Program Files (x86)\chromedriver.exe')
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_myLoginTest(self):
        self.driver.get("http://eprint.com.hr/demo/addauser.php")
        self.driver.set_window_size(936, 952)
        self.driver.find_element(By.NAME, "username").click()
        self.driver.find_element(By.NAME, "username").send_keys("admin1")
        self.driver.find_element(By.NAME, "password").click()
        self.driver.find_element(By.NAME, "password").send_keys("1234")
        self.driver.find_element(By.NAME, "FormsButton2").click()
        self.driver.find_element(By.CSS_SELECTOR, "small > a:nth-child(1)").click()
        self.driver.find_element(By.NAME, "username").click()
        self.driver.find_element(By.NAME, "username").click()
        self.driver.find_element(By.NAME, "username").send_keys("admin1")
        self.driver.find_element(By.NAME, "password").click()
        self.driver.find_element(By.NAME, "password").send_keys("1234")
        self.driver.find_element(By.NAME, "FormsButton2").click()
```

The open a Powershell windows as Administrator into the directory where you have your Python script, then run the following command:

```
PS C:\test> pip install selenium
```

```
PS C:\test> pip install pytest
```

```
PS C:\test> npm install -g selenium-side-runner
```

```
PS C:\test> selenium-side-runner <ProjectName>.side
```