Understanding PlayWright

**Intro-What is PlayWright?**

* PlayWright is used for originally **Javascript programming only** as it is a library created for Node.js , but **now is expanded to other programming languages such as Python**, Java, etc…
* Also PlayWright is…
* An **OpenSource** **automation** **tool/framework for the Web** created by Microsoft that is versatile and can be used for many things such as:

1. **Similar to Selenium** we can use PlayWright to automate testing a website as PlayWright can also simulate end-user actions such as clicking, going through webPages, form submission, etc…
2. Records visual UI’s states by keeping screenshots or videos of specified events
3. Accepted Browsers = Firefox, Google Chrome, and Webkt
4. **And much more check out Works Cited** for more info on the capabilities of PlayWright

**How to start using PlayWright Python Version?**

1. Install PlayWright —  **pip install playwright**
2. Install browser for PlayWright— **playwright Install BrowserName**

**Ex:** playwright install firefox # installs components for playWright to communicate with specified browser(BrowserName)

1. Import for to use in specified file— EX:

**from playwright.sync\_api import sync\_playwright**

# where ‘from playwright.sync\_api’ imports a specified location of playwright

# what/which function inside location ‘import sync \_playwright

# Together ‘from playwright.sync\_api import sync\_playwright’ allows you to write named functions that can be called anytime (synchronous function)

**Simple Example using PlayWright(PYTHON VERSION)**

1. I want to use Google Chrome so in the terminal I type the code:

npm install playwright-chromium

2. In this example, I will open Google Chrome and Go to Youtube.com

from playwright.sync\_api import sync\_playwright

with sync\_playwright() as p: # Setup Playwright/ creates a playWright object named p

browser = p.chromium.launch() # Open browser

context = browser.new\_context() # Create a private browser session

page = context.new\_page() # Create a new tab

page.goto('https://www.youtube.com') # Go to YouTube

# Optional: Perform more simulated Playwright user actions... (Optional additional code)

browser.close() # Close the browser

**Extra(PYTHON):**

[**https://www.youtube.com/watch?v=y8zY14HHiPI&list=PLP5\_A7hnY1Tggph0F0cRqf5iyyZuIBXYC**](https://www.youtube.com/watch?v=y8zY14HHiPI&list=PLP5_A7hnY1Tggph0F0cRqf5iyyZuIBXYC)

**Optional/ JS content**

**Quick Understanding What is Node.JS?**

* An Environment that **allows back-end coding/programming in JavaScript**
* **Back-end Programming =** programming to Process data, as in **storing, retrieving, and managing/manipulating data**, in a program.

**How to start using PlayWright JS VERSION?**

1. **Have an understanding of Javascript**/Typescript(another version of JavaScript) **Recommendation:** read a book/ watch YouTube videos on the foundation of JavaScript and also download an IDE to practice what you’ve learned for example: <https://code.visualstudio.com/download>
2. **Install Node.JS** on your personal computer(Google installing Node.JS if still confused): <https://www.youtube.com/watch?v=eauREi0vwB0>
3. In your IDE **create a Node.JS environment/project(as mentioned in the video above/step 2)**.
4. **Install the PlayWright** Package **using** the command: **npm install playwright**
5. Pick a Browser using the command: ***npm install playwright-***BrowserName Where browser name is the name of the browser you wish to use playWright in.

**Step By Step Explanation using JS but doing the same as the python version:**

//(async () => { //write playWright executions/code here! })();

* //creates an anonymous function(function to only be used once) and by naming it with the keyword ‘async’ (turns it into an asynchronous anonymous function) it allows the anonymous function to do certain commands at certain times using the keyword 'await'. This is optional but allows a more controlled environment for automating the browser using PlayWright**(In this example we will not be using this code to simplify learning)**

**…………………………………………………………………STARTING POINT:**

const { chromium } = require('playwright'); //setup PlayWright

const { chromium } = require('playwright');

* //This code accomplishes **two things** **1.** creates a variable to represent the Google Chrome browser **2.** sets the playwright to be used within the file

const browser = chromium.launch();

* **opens**/launches the **Google Chrome(or specified Browser)**

const context = browser.newContext();

* **Creates a private browser session** that has its own cookies, caches, etc…(won’t affect other browser sessions even if that season is another playWright session) In all this code **essentially separates data from the main browser session.**

const page = context.newPage();

* // **Creates a new Tab** within the chromium variable/browser

page.goto('https://www.Youtube.com');

* // **Goes to** specified **website**/string

browser.close();

* // **Closes browser** session

**Extra help/Example:**

**Intro/Basic(JS)–**

[**https://www.youtube.com/watch?v=4\_m3HsaNwOE&list=PLhW3qG5bs-L9sJKoT1LC5grGT77sfW0Z8**](https://www.youtube.com/watch?v=4_m3HsaNwOE&list=PLhW3qG5bs-L9sJKoT1LC5grGT77sfW0Z8)

**A little more advanced/more specific with examples(JS)–**

[**https://www.youtube.com/watch?v=yOuElUSfAs8&list=PLUDwpEzHYYLsw33jpra65LIvX1nKWpp7-**](https://www.youtube.com/watch?v=yOuElUSfAs8&list=PLUDwpEzHYYLsw33jpra65LIvX1nKWpp7-)

Works Cited

Shrivastva, Arjun M. “The Ultimate Playwright Python Tutorial.” *Browserstack*, 4 May 2023, https://www.browserstack.com/guide/playwright-python-tutorial. Accessed 26 Nov. 2023.

Playwright. “Installation.” *Microsoft*, https://playwright.dev/docs/intro. Accessed 26 Nov. 2023.

Playwright. “PlayWright.” *Microsoft*, https://github.com/microsoft/playwright-python. Accessed 26 Nov. 2023.

SimpliLearn. “Node.js Tutorial.” *SimpliLearn*, 11 Oct. 2023, https://www.simplilearn.com/tutorials/nodejs-tutorial. Accessed 26 Nov. 2023.