

Snipe-IT Playwright .NET Automation Suite

This repository contains a Playwright + NUnit automation suite for validating core asset lifecycle workflows on: <https://demo.snipeitapp.com>

Overview

- Framework: **.NET 10**
- Automation Library: **Microsoft Playwright 1.57.0**
- Test Framework: **NUnit**
- Target Application: **Snipe-IT Demo**
- Credentials: **admin / password**

The suite creates real assets using the prefix `TESTMAC-xxxxxx`, checks them out, searches them, and validates:

- Model, Serial number correctness
 - Asset history (with screenshot evidence)
-

Repository Structure (Important)

After extraction, the relevant paths are:

- Solution Root
`SnipeITAutomationSuite\`
- Test Project (IMPORTANT)
`SnipeITAutomationSuite\SnipeIt.Playwright.Tests\`

*All Playwright installation and build commands **must be run inside the test project folder**.*

HOW TO RUN (MOST IMPORTANT SECTION)

Step 1: Prerequisites (One Time Per Machine)

Install .NET 10 SDK

Download and install .NET 10 SDK from Microsoft.

Verify:

```
dotnet --version
```

Step 2: Extract the Zip

Open **PowerShell** in the directory containing `SnipeITAutomationSuite.zip`

```
tar -xf SnipeITAutomationSuite.zip  
cd .\SnipeITAutomationSuite\SnipeIt.Playwright.Tests\
```

Step 3: Install & Build Playwright (REQUIRED)

Playwright **must be installed and built before running tests**.

Run the following commands **inside** `SnipeIt.Playwright.Tests`:

```
dotnet add package Microsoft.Playwright  
dotnet build
```

If you skip this step, you may see errors such as:

- Playwright not installed

- Playwright tool not built
 - Browser binaries missing
-

Step 4: Install Playwright Browsers

```
dotnet tool update --global Microsoft.Playwright.CLI  
playwright install
```

Running the Tests

There are **two supported execution modes**.

Mode 1: Normal Test Run (Recommended Default)

1. Optional: Run Prerequisite Validation Tests

These tests validate whether Category, Manufacturer, and Model already exist. They are **optional** because the demo environment usually has them.

```
dotnet test --filter "PrerequisitesTests"
```

2. Run Asset Lifecycle Test (Headless)

This is the main test that:

- Creates an asset
- Checks it out
- Validates Model, Serial number, User checkout

```
dotnet test --filter "AssetLifecycleTests"
```

Mode 2: Debug / Visual Mode (Highly Recommended for Review)

Runs Playwright in **headed mode** with **slow motion** and **Playwright Inspector**.

```
$env:PLAYWRIGHT_BROWSER_HEADLESS="false";  
$env:PLAYWRIGHT_SLOW_MO="1000";  
$env:PWDEBUG="1";  
dotnet test --filter "AssetLifecycleTests"
```

When the Playwright Inspector opens:

- Click **Resume** or press **F8**
 - Repeat after each navigation pause
-

Test Artifacts (Screenshots)

After the Asset Lifecycle test completes, a screenshot of the **History tab** is automatically saved.

Location: `SnipeITAutomationSuite\SnipeIt.Playwright.Tests\`

Filename example: `history_checkout_TESTMAC-030058_20251224_110123.png`

What the Tests Validate

- Chromium browser launch (headless or headed)
 - Admin login
 - Asset creation with prefix `TESTMAC-`
 - Asset search and detail validation
 - Asset history verification (with screenshot)
 - Clean PASSED execution
-

Manual Verification

Created assets can be manually verified at: <https://demo.snipeitapp.com/hardware>

Search for: TESTMAC-

Notes

- Playwright installation **must** be done from the test project folder
- Debug mode is recommended for first-time reviews