

How to Upload Files in Playwright

Playwright is an automation tool that can interact with web applications and APIs. While it cannot automate window-based applications, it can automate tasks like uploading files from your local machine to a web application. This can be done using the `setInputFiles()` function.

Three Main Actions with `setInputFiles()`

1. Single File Upload
 2. Multiple Files Upload
 3. Removing Uploaded Files
-

1. Single File Upload

To upload a **single file** from your local machine to a web application, follow these steps:

Steps:

1. Find the locator for the "Choose File" button on the page.
2. Use the `setInputFiles()` function to upload the file.

This will trigger the file upload dialog and select the file from the path you provided.

Example DOM:

```
<input type="file" class="chooseFileButton" />
```

Code Example:

```
const { test, expect } = require('@playwright/test');

test('Single file upload', async ({ page }) => {
    await page.goto('https://example.com'); // Replace with your actual URL
    await page.locator('.chooseFileButton').setInputFiles('tests/uploadFiles/testFile1.pdf'); // Path to your file

    // Verify if the file is uploaded successfully
    await expect(await page.locator('#uploadedFile')).toHaveText('File Uploaded Successfully');
});
```

2. Multiple File Upload

To upload **multiple files** from your local machine, you can use the same `setInputFiles()` function, but this time pass an **array** of file paths.

Steps:

1. Find the locator for the "Choose File" button.
2. Pass an array of file paths to the `setInputFiles()` function.

Example DOM:

```
<input type="file" class="chooseFileButton" />
```

Code Example:

```
const { test, expect } = require('@playwright/test');

test('Multiple file upload', async ({ page }) => {
    await page.goto('https://example.com'); // Replace with your actual URL
    await page.locator('.chooseFileButton').setInputFiles([
        'tests/uploadFiles/testFile1.pdf', // Path to the first file
        'tests/uploadFiles/testFile2.pdf' // Path to the second file
    ]);

    // Verify if the files are uploaded successfully
    await expect(await page.locator('# fileList li:nth-child(1)').toHaveText('testFile1'));
    await expect(await page.locator('# fileList li:nth-child(2)').toHaveText('testFile2'));
});
```

3. Removing Uploaded Files

If you want to **remove** the uploaded files, you can pass an **empty array** to the `setInputFiles()` function.

How Does Passing an Empty Array Remove the File?

By passing an empty array to the `setInputFiles()` function, it clears the input, effectively removing any uploaded files.

Example DOM:

```
<input type="file" class="chooseFileButton" />
```

Code Example:

```
const { test, expect } = require('@playwright/test');

test('Remove uploaded files', async ({ page }) => {
  await page.goto('https://example.com'); // Replace with your actual URL
  await page.locator('.chooseFileButton').setInputFiles([]); // Clear the input to remove files

  // Verify if the files were removed
  await expect(await page.locator('#fileList li:nth-child(1)')).toHaveText('No file attached');
});
```

Conclusion

- **Single File Upload:** Use `setInputFiles()` with a single file path.
 - **Multiple File Upload:** Use `setInputFiles()` with an array of file paths.
 - **Remove Uploaded Files:** Use `setInputFiles()` with an empty array.
-