

# American Airlines booking flight - selecting the first search result

## (Playwright Test Documentation)

### Overview

This Playwright test suite automates the process of booking a flight on the American Airlines website. The test scenario includes multiple test steps with screenshots for every step, a detailed report, and trace files to assist with debugging. Each step captures screenshots to ensure thorough documentation and clarity in auditing/reporting.

### Approach

1. **Modularized Steps:** The tests are broken down into specific steps to ensure each part of the booking flow can be tested and validated individually. Each step includes logging, screenshot capture, and assertions to verify expected outcomes.
2. **Comprehensive Reporting:** The test suite is configured to generate an HTML report and capture screenshots for each test step. Tracing is enabled for every test run, providing in-depth logs and playback for debugging.
3. **Configurations:** The Playwright configuration file ([playwright.config.ts](#)) is set to capture screenshots on all steps, store trace files, and generate HTML reports after each test run.

### Assumptions

1. **Website Consistency:** The American Airlines website layout and locator names are assumed to remain stable. Changes to website structure or locator IDs may require updating selectors in the tests.
2. **Date Selection:** The departure date input assumes the availability of a specific date format (MM/DD/YYYY). Any format change may require modifying the date input format and the date (which is hardcoded in the code) in the test.
3. **Authentication:** No login or session authentication is required for this test scenario. If login becomes a requirement, authentication steps must be added.

### Setup and Prerequisites

1. **Node.js** (v14 or later) should be installed. You can verify your Node.js installation by running:  
`node -v`  
`npm -v`
2. **Playwright Installation:** If Playwright is not installed, you can install it with:  
`npm install @playwright/test`
3. **Browser Dependencies:** Ensure Playwright browsers are installed. You can install these with:  
`npx playwright install`

# Configuration

The configuration file, `playwright.config.ts`, has been customized to ensure:

- **Screenshots** are taken for all steps.
- **Tracing** is enabled for all test runs.
- **HTML Reports** are generated in the `test-results` directory.
- **Headless** is configurable as needed; the current config is set to `false` to run as headed.  
Please refer to the >> [playwright.config.ts](#)

## Test Execution Instructions

1. **Clone or Download the Test Suite:** Clone the repository containing this test suite to your local machine.
2. **Run the Tests:** In the root directory of the project, run:  
`npx tsc`  
`npx playwright test tests/flight-reservation.spec.ts`
3. **View Screenshots:** Screenshots for each step are saved in the `screenshots` folder (or within the `test-results` folder if that is specified in each test step). Screenshots are labeled with step descriptions for easy identification.

## Detailed Test Breakdown

1. **Test setup and Test teardown:**
  - Launching the headless chromium browser.
  - Create New Context and new Page.
  - Closes all the browsers after the completion of execution.
2. **Navigate to Website:**
  - Opens the American Airlines website.
  - Waits until the content is fully loaded.
  - Captures a screenshot for documentation.
3. **Select Flight Type:**
  - Locates and selects the “One way” option.
  - Validates that the option was selected successfully by checking visibility.
  - Captures a screenshot.
4. **Enter Origin and Destination Airports:**
  - Inputs origin and destination airports (ATL and BUF).
  - Logs inputs for validation.
  - Captures a screenshot for each input action.
5. **Select Departure Date:**
  - Entering a specific departure date.
  - Logs confirmation of date selection.
  - Captures a screenshot of the date input field.
6. **Submit Flight Search:**
  - Checks for and clicks the search button.
  - Logs status of button visibility.
  - Captures a screenshot upon clicking the button.

7. **Validate Search Results and Capture Data:**
  - Validates that the flight count displayed is greater than Zero.
  - Logs both UI flight counts and displays in the console the total count.
  - Captures a screenshot of the results page.
8. **Select First Flight:**
  - Chooses the first available flight option.
  - Ensures the selection was successful by checking visibility.
  - Captures a screenshot after selecting the flight.
9. **Choose Basic Economy Fare Option:**
  - Selects the Basic Economy fare.
  - Ensures the selection was successful by checking visibility.
  - Captures a screenshot after selecting the basic fare option.
10. **Skip Upgrade:**
  - Locates and clicks the “No Upgrade” option.
  - Logs confirmation and captures a screenshot.
11. **Verify Trip Summary:**
  - Verifies that the trip summary section is displayed and includes the correct text.
  - Captures a final screenshot of the trip summary.