14.Reporting

supports **built-in** and **custom reporters**.

* Reporters can be configured:
  + Using the --reporter CLI option.
  + Programmatically in playwright.config.ts.
* Multiple reporters can be used simultaneously.
* Reporters differ in verbosity and output format.

**Built-In Reporters**

**List Reporter**

* **Purpose:** Prints each test's status line-by-line.
* **Default:** Local runs (except CI where Dot is default).
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: 'list',

});

* **Options:**
  + printSteps: Shows each step of the test if set to true.
  + PLAYWRIGHT\_LIST\_PRINT\_STEPS: Environment variable for step rendering.
  + PLAYWRIGHT\_FORCE\_TTY & FORCE\_COLOR: Control live terminal and color output.
* **Use Case:** Detailed test output for local debugging.

**Line Reporter**

* **Purpose:** Displays concise output with test progress and inline failures.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: 'line',

});

* **Environment Variables:**
  + PLAYWRIGHT\_FORCE\_TTY: Terminal suitability for live output.
  + FORCE\_COLOR: Enables colored output.
* **Use Case:** Large test suites; balances progress visibility and clutter.

**Dot Reporter**

* **Purpose:** Minimalistic output; represents test status using single characters.
* **Default:** CI environments.
* **Characters Used:**
  + ·: Passed
  + F: Failed
  + T: Timed out
  + °: Skipped
  + ±: Passed after retry
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: 'dot',

});

* **Environment Variables:**
  + Same as Line Reporter.
* **Use Case:** Avoids clutter; ideal for CI.

**HTML Reporter**

* **Purpose:** Generates a shareable HTML report with test results.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: [['html', { outputFolder: 'my-report', open: 'never' }]],

});

* **Options:**
  + outputFolder: Directory to save reports.
  + open: When to auto-open (always, never, on-failure).
  + attachmentsBaseURL: Path for external attachments.
* **Commands:**
  + Generate: npx playwright test --reporter=html.
  + Open last report: npx playwright show-report.
* **Use Case:** Post-run analysis and sharing test results.

**Blob Reporter**

* **Purpose:** Captures detailed test data for merging or further report generation.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: [['blob', { outputFile: './blob-report/report.zip' }]],

});

* **Options:**
  + outputFile: Full path for the blob report.
  + outputDir: Directory to save reports.
  + fileName: Custom filename for the report.
* **Use Case:** Sharded tests and advanced reporting needs.

**JSON Reporter**

* **Purpose:** Generates a JSON object with all test details.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: [['json', { outputFile: 'results.json' }]],

});

* **Options:**
  + outputFile: Path to save JSON.
  + Environment variables like PLAYWRIGHT\_JSON\_OUTPUT\_NAME.
* **Use Case:** Integration with other tools and custom reporting.

**JUnit Reporter**

* **Purpose:** Produces JUnit-style XML reports.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: [['junit', { outputFile: 'results.xml' }]],

});

* **Options:**
  + outputFile: Path to save XML.
  + Additional configs like includeProjectInTestName.
* **Use Case:** CI/CD pipelines that support JUnit reporting.

**GitHub Reporter**

* **Purpose:** Generates annotations for GitHub Actions.
* **Configuration Example:**

javascript

Copy code

export default defineConfig({

reporter: process.env.CI ? 'github' : 'list',

});

* **Use Case:** Debugging test failures on GitHub Actions.

**Custom Reporters**

* **Purpose:** Tailored reporting by implementing the Reporter API.
* **Example:**

javascript

Copy code

import type { Reporter } from '@playwright/test';

class MyReporter implements Reporter {

// Custom methods

}

export default MyReporter;

* **Use Case:** Unique reporting requirements not covered by built-ins.

**File-Specific Notes**

**playwright.config.ts**

* Houses configuration for all reporters.
* Multiple reporters can be combined:

javascript

Copy code

export default defineConfig({

reporter: [

['list'],

['json', { outputFile: 'results.json' }],

],

});

**Command-Line Usage**

* Test runs with specified reporters:

bash

Copy code

npx playwright test --reporter=line

npx playwright test --reporter=html