

Redder Stack Test Plan —Automation Assignment

1. Overview

This plan describes the automation approach for validating key flows in Rudder stack, including creating sources and destinations, sending test events via the HTTP API, and checking delivery through the UI. The automation will be written in JavaScript using Playwright + Cucumber JS and integrated with GitHub Actions for scheduled runs.

2. Objective

- Automate creation of an HTTP Source and Webhook Destination in Rudderstack.
- Send events to the HTTP Source using Rudderstack's API and validate delivery in the UI.
- Include basic positive and negative test coverage.
- Ensure tests run daily in CI without manual intervention.

3. Tech Stack

Category	Tool / Tech
Automation Framework	Playwright
Test Runner	Cucumber JS
Language	JavaScript
CI/CD	GitHub Actions
Env Management	.env files
External Tools	Request Catcher, Temp-Mail
API Client	fetch / axios

4. Test Environment

- **Rudder stack App:** <https://app.rudderstack.com/>
- **Webhook Monitor:** <https://requestcatcher.com>
- **Temporary Email Provider:** <https://temp-mail.org/>
- **API Reference:** <https://www.rudderstack.com/docs/api/http-api/>

Environments managed through .env files:

- DEV
- QA
- PROD

5. Test Scenarios

As per the assignment

6. Test Strategy

UI Testing:

- Implemented with Playwright page object model.
- Cucumber feature files for human-readable test definitions.
- Step definitions in JavaScript using Playwright selectors.

API Testing:

- Use fetch or axios to send events to Rudderstack HTTP source.
- Validate response code and status.

CI/CD:

- GitHub Actions will run tests daily.
- .env values will be injected using GitHub Secrets.
- Test results will be logged in the Actions workflow.

7. Test Data

- Temporary business email account for signup.
- API payload to simulate event data.
- Webhook URL from Request Catcher.

8. Folder Structure (As per the assignment)

```
rudderstack-automation/  
├── features/  
│   ├── login.feature  
│   ├── event-delivery.feature  
│   └── step_definitions/  
├── pages/  
│   └── connectionPage.js  
├── utils/  
│   └── apiClient.js  
└── .env
```

```
├── playwright.config.js
├── cucumber.js
├── package.json
├── .github/
│   ├── workflows/
│   └── daily-run.yml
└── README.md
```

9. Risks & Mitigations

Risk	Mitigation
Temp email expiring or invalidating session	Use long-duration temporary emails
Rudder stack UI changes could break locators	Use stable selectors and retry on test failure
API event delivery takes time	Add retries or polling on Webhook check step
Data Plane URL or Write Key is not visible	Add validation steps or mock fallback

10. Success Criteria

- All steps execute end-to-end without manual intervention.
- At least one successful event is recorded in the Webhook destination.
- Scheduled GitHub Actions job runs daily and passes.