

ā'pěks alpe adria

APEX Alpe Adria

12.04.2024

A regional Oracle APEX conference for everyone #aaapeks24

APEX automated testing - power of automation

Lino Schildenfeld
@LinoSchilde
lschilde.blogspot.com



MASH PROGRAM

Mentor and Speaker Hub

- Our goal is to *connect* speakers with mentors to assist in *preparing* technical sessions and *improving* presentation skills

Interested? Read more and get in touch <https://mashprogram.wordpress.com>

My story

@LinoSchilde

.SYM^{L2}

AUSOUG

- AUSOUG APEX committee chair
- NZ APEX meetup organizer



- APEX blogger
- Conference speaker

- apex.world member of the month x2
- 15+ years of APEX experience

 Oracle ACE
Pro

 MASH

Other sessions

Room 1  10:15 - 11:00	Room 2  10:15 - 11:00	Room 3  10:15 - 11:00
ChatBot and AI: Question your data Dimitri Gielis	Mastering APEX: Unlocking the hidden functionality of native APEX components Petar Simic	Revolutionizing database management: Oracle's SQLcl Liquibase Success Story Rafal Grzegorczyk
Room 1  11:15 - 12:00	Room 2  11:15 - 12:00	Room 3  11:15 - 12:00
APEX 23.2 Working with Application Working Copies Richard Martens	Multilanguage apps? Sí! Yes! Oui! Ja! Da! Да! Zoran Tica	Breaking Barriers: Accessibility in Oracle APEX Tomáš Kucharzyk
Room 1  12:15 - 13:00	Room 2  12:15 - 13:00	Room 3  12:15 - 13:00
Dev Team Insights on APEX Fusion Apps Extensions Steve Muench	Hard to say goodbye to Oracle Forms? Leon Stanko	How to write a perfect PL/SQL code Dejan Topalovic
Room 1  14:15 - 15:00	Room 2  14:15 - 15:00	Room 3  14:15 - 15:00
APEX Blueprint - build your apps the right way Jan Květina	APEX automated testing - power of automation Lino Schildenfeld	Oracle REST Data Services (ORDS) best practices Vili Tajnić
Room 1  15:15 - 16:00	Room 2  15:15 - 16:00	Room 3  15:15 - 16:00
APEX + Cloud: Are you in the cloud yet? Vladislav Uvarov	Cookie Cutter APEX Applications Alex Nijtjen	Linking all my APEX together with REST Enabled SQL David Pataki
Room 1  16:30 - 17:15	Room 2  16:30 - 17:15	Room 3  16:30 - 17:15
Oracle Database 23c for the APEX Developer Roel Hartman	Not so fast, cowboy! - about application features that are not so easy to implement in APEX, but they should be. Adam Kierzkowski	Boost your APEX development workflow Giliam Breems

Today's goal

Sample Database Application

Cancel **Add Customer**

First Name * Last Name *

Street Address Line 2

City State * - Choose State -

Zip Code *

Credit Limit *

Phone Number 999-999-9999 Alternate Number 999-999-9999

Email URL

Tags

Help demo ▾

WHY Automated testing?!

- **Improves software quality**
- **Much faster than manual**
- Saves Time and Money
- Reduces manual errors
- Improves Standards
- Faster time to market
- Faster Feedback
- Improved Productivity
- Faster Debugging
- Return on investment is high
- **Faster APEX Upgrading**
- Improves experience - **Happier clients**

Interesting – decision time

- **Licence cost or no cost option**
- **Support**
- **Usability and maintenance**
- **Skillset**
- **What do I want it to do?!!!**
 - Record a video
 - Upload to the cloud
 - Store and Export results
 - Handle a mobile testing
 - Documentation
 - Easy to use/install

Idea

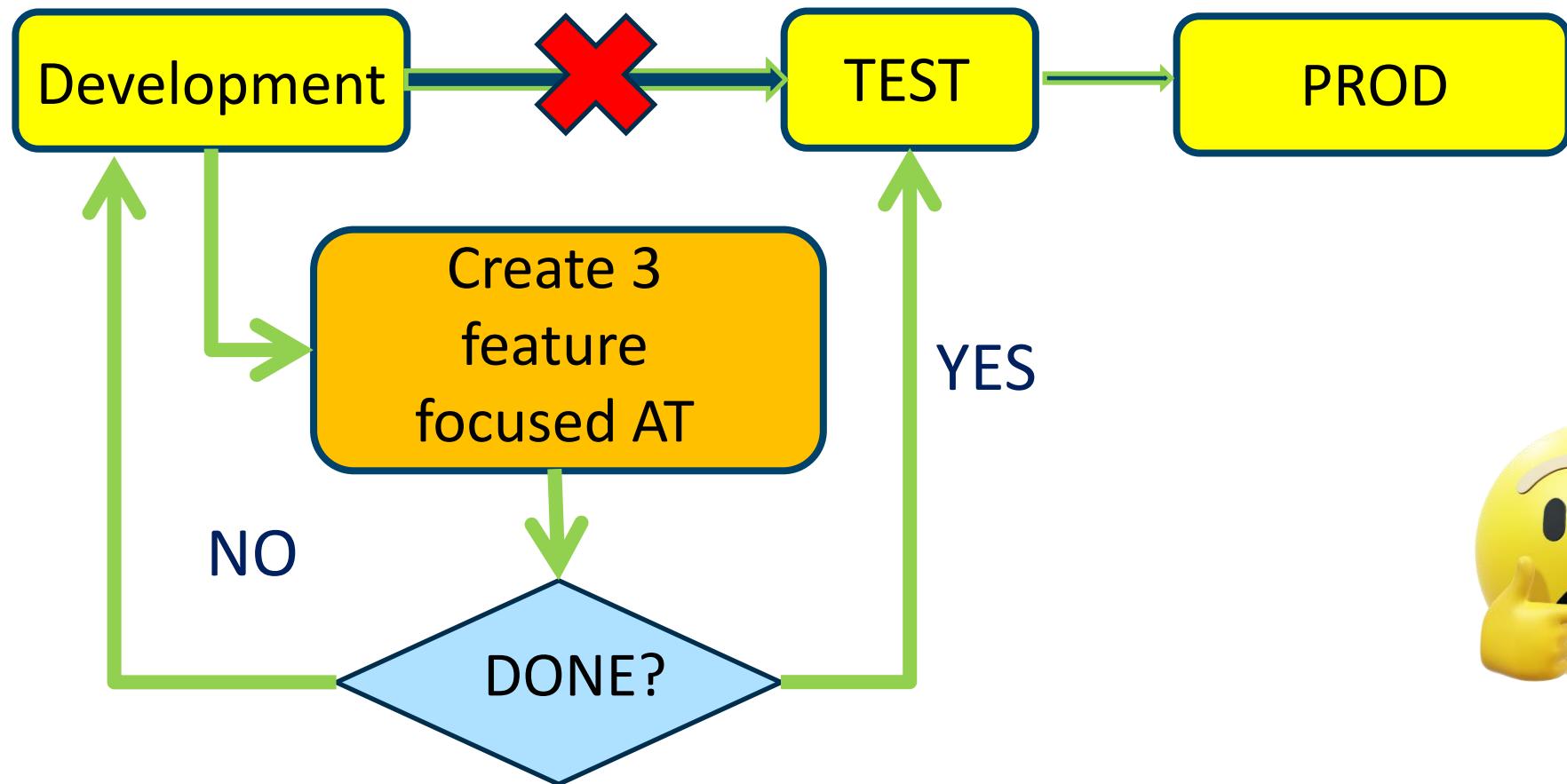


Each new feature developed



3 AUTOMATED TESTS

Consider



Automated testing tools

- Selenium
- **CYPRESS**
- ATAF
- utPLSQL
- Codeconcept.io
- JMeter
- Katalon
- Gatling
- HammerDb
- Appium
- Autoit
- Puppeteer
- **Ghost Inspector**
- Playwright
- LCT

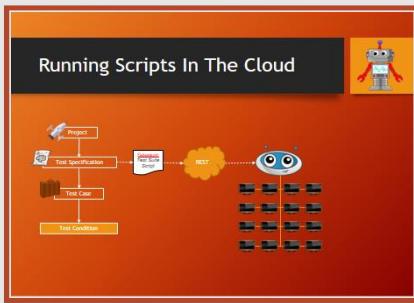
- Rarely run and costly
- Can improve your coding standards
- Built by developers as part of DEV cycle
- Biggest challenge is synchronization (unreliable)

“Web testing is one of the critical parts of modern software lifecycle”

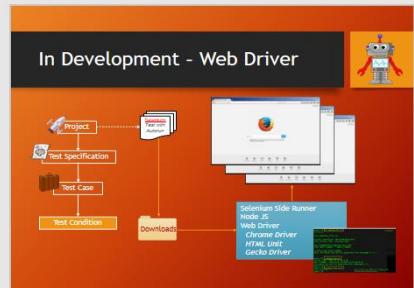
6



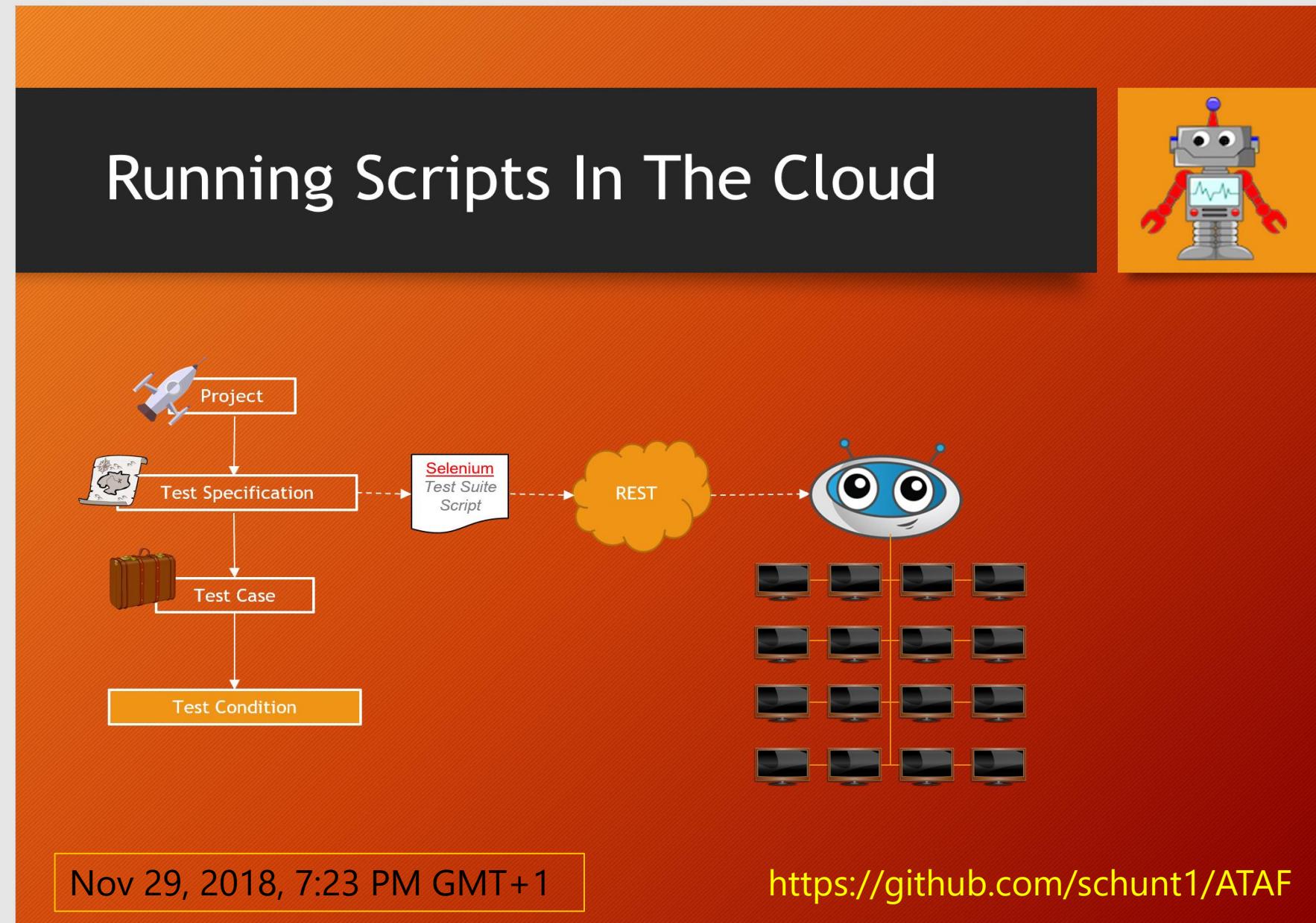
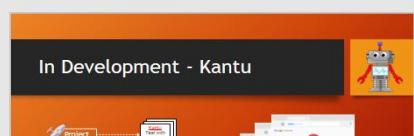
7



8



9



CYPRESS

WHY CYPRESS

- **Easy to get going** and **learn** how to use
- **Good documentation**
- It can video **record** the entire **tests** & takes snapshots
- **Automatically waits** (for commands and assertions before moving on*)
- Supports Firefox, MS Edge, Chrome
- Uses Mocha syntax and Chai assertions
- Smaller community with momentum in last few years



How it works

Installation

- Install node.js
- Create a directory
- Position your self into it
- Run **npm init -y**
- Run

```
npm install cypress --save-dev
```

```
LS@LS-HP-x360 MINGW64 ~/Desktop/CYPRESS
$ npm install cypress -D
npm WARN deprecated har-validator@5.1.5: this library is no longer supported

> cypress@5.1.0 postinstall C:\Users\LS\Desktop\CYPRESS\node_modules\cypress
> node index.js --exec install

Installing Cypress (version: 5.1.0)

  ✓ Downloaded Cypress
  ✓ Unzipped Cypress
  ✓ Finished Installation C:\Users\LS\AppData\Local\Cypress\Cache\5.1.0

You can now open Cypress by running: node_modules\.bin\cypress open

https://on.cypress.io/installing-cypress

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN CYPRESS@1.0.0 No description
npm WARN CYPRESS@1.0.0 No repository field.

+ cypress@5.1.0
added 216 packages from 147 contributors and audited 216 packages in 256.478s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

First run

- We need text editor and a command line
- Run `npx cypress open`
- Check your `package.json` for correct dependencies

- `npx cypress verify`
- `npx cypress info`
- `npx cypress version`

```
LS@LS-HP-x360 MINGW64 ~/Desktop/CYPRESS
$ npx cypress run
```

HelloWorld.js

```
describe('Welcome to Cypress', ()=> {
  | context('Trying out things', () => {
    |   Open Cypress | Set ".only"
    |   | it('Test 1', () => {
    |   |   | expect(1).to.equal(1);
    |   |   | })
    |   | it.skip('Test2', () => {
    |   |   | expect(true).to.equal(true);
    |   |   | });
    |   | });
  | });
});|
```

TestRunner

```
LS@LS-HP-x360 MINGW64 ~/Desktop/CYPRESS
$ npx cypress open
It looks like this is your first time using Cypress: 5.1.0

    ✓ Verified Cypress! C:\Users\LS\AppData\Local\Cypress\Cache\5.1.0\Cypress

Opening Cypress...
```

The screenshot shows the Cypress Test Runner application running in a terminal window. The terminal output indicates that Cypress version 5.1.0 is being opened for the first time. The application interface below the terminal includes a navigation bar with File, Edit, View, Window, Help, and developer information for L Schilde. The main area displays a tree view of integration tests under the 'INTEGRATION TESTS' category, including 'apex_110' with files 'app110.login.js' and 'app110.loginBaseURL.js'. A search bar and a 'Run all specs' button are also visible.

C:\Users\LS\Desktop\CYPRESS

File Edit View Window Help

CYPRESS

Support Docs L Schilde

Tests Runs Settings Chrome 85

Search... Run all specs

INTEGRATION TESTS COLLAPSE ALL | EXPAND ALL

apex_110

app110.login.js

app110.loginBaseURL.js

Key tips

Selectors

- Might require **changes** to your **APEX applications**
- Try to standardize these across your team
- Learning how to do this is a **key challenge**
- Watch out for modals

```
cy.url().should("contain", "110:LOGIN_DESKTOP");
cy.get('.t-Login-title').should("contain", "Sample Database Application");
```

<https://docs.cypress.io/guides/references/assertions.html#Chai>

Key tips

Assertions

- Uses popular **Chai assertion** library (including Sinon & jQuery)
- **Hardest** in web automation testing to get right
- Modal dialogs, reports and interactive grid validation

```
cy.url().should("contain", "110:LOGIN_DESKTOP");
cy.get('.t-Login-title').should("contain", "Sample Database Application");
```

<https://docs.cypress.io/guides/references/assertions.html#Chai>

Final demo

Playwright

```
MINGW64:/c/Users/luchi/Desktop/Playwright
```

```
luchi@LAPTOP-3MI58QCO MINGW64 ~/Desktop/Playwright
$ npm init playwright@latest
npx: installed 1 in 1.692s
Getting started with writing end-to-end tests with Playwright:
Initializing project in '.'
? Do you want to use TypeScript or JavaScript? ...
> TypeScript
  JavaScript                                     JavaScript
✓ Do you want to use TypeScript or JavaScript? · JavaScript
✓ Do you want to use TypeScript or JavaScript? · JavaScript
✓ Do you want to use TypeScript or JavaScript? · JavaScript
✓ Do you want to use TypeScript or JavaScript? · JavaScript
? Where to put your end-to-end tests? » tests
✓ Where to put your end-to-end tests? · tests
? Add a GitHub Actions workflow? (y/N) » false
✓ Add a GitHub Actions workflow? (y/N) · truey
```



All 3 | Passed 3 | Failed 0 | Flaky 0 | Skipped 0

example.spec.js

6.8s

✓ homepage has Playwright in title and get started link linking to the intro page
example.spec.js:4

chromium

3.1s

✓ homepage has Playwright in title and get started link linking to the intro page
example.spec.js:4

firefox

2.0s

✓ homepage has Playwright in title and get started link linking to the intro page
example.spec.js:4

webkit

1.7s

MINGW64:/c/Users/Ischi/Desktop/Playwright

Running 3 tests using 3 workers

[1/3] [chromium] > example.spec.js:4:1 > homepage has Playwright in title and get start
[2/3] [firefox] > example.spec.js:4:1 > homepage has Playwright in title and get starte
[3/3] [webkit] > example.spec.js:4:1 > homepage has Playwright in title and get started

3 passed (6s)

To open last HTML report run:

npx playwright show-report

Ischi@LAPTOP-3MI58QCO MINGW64 ~/Desktop/Playwright

\$ npx playwright show-report

Serving HTML report at http://127.0.0.1:9323. Press Ctrl+C to quit.

JS example.spec.js X

C: > Users > lschi > Desktop > Playwright > tests > JS example.spec.js > ...

```
1 // @ts-check
2 const { test, expect } = require('@playwright/test');
3
4 test('homepage has Playwright in title and get started link linking to the intro page', async ({ page }) => {
5   await page.goto('https://playwright.dev/');
6
7   // Expect a title "to contain" a substring.
8   await expect(page).toHaveTitle(/Playwright/);
9
10  // Create a locator
11  const getStarted = page.locator('text=Get Started');
12
13  // Expect an attribute "to be strictly equal" to the value
14  await expect(getStarted).toHaveAttribute('href', '/docs/intro');
15
16  // Click the get started link.
17  await getStarted.click();
18
19  // Expects the URL to contain intro.
20  await expect(page).toHaveURL(/.*intro/);
21 });
22
```

```
it('Visit Page 2', ()=> {
  cy.get('#t_TreeNav_1 > .a-TreeView-content > .fa').click();

  cy.url().should("contain", "/customers");
  cy.get('[data-cy="create_customer_non_modalButton"]').click();

  const uuid = () => Cypress._.random(0, 1e6);
  const id = uuid();
  const firstName = `DemoCypress${id}`; Cypress

  cy.get("#P18_CUST_FIRST_NAME").type(firstName);
  cy.get("#P18_CUST_LAST_NAME").type("Demo");
  cy.get("#P18_CUST_STREET_ADDRESS1").type("Address1");
  cy.get("#P18_CUST_STREET_ADDRESS2").type("Address2");
  cy.get("#P18_CUST_CITY").type("Chicago");
  cy.get("#P18_CUST_EMAIL").type("xx@myemail.com");
```

Back to Cypress

Manual run

- Good as it can be scripted
- Ability to pass in multiple parameters
- By default `cypress run` **records a video locally**



```
$ npx cypress run --record
```

(Run Starting)

```
Cypress: 5.1.0
Browser: Electron 83 (headless)
Specs: 34 found (apex_110\app110.login.js, apex_110\app110.loginBaseUrl.js, apex_110\app110.loginCommon.js, apex_110\app110.loginPwd.Assertions.js, apex_110\app110.loginPwd.AssertionsPost.js, apex_110\app110.loginPwd.js, apex_110\app110.loginPwdBad.js, apex_110\ap...)

Params: Tag: false, Group: false, Parallel: false
Run URL: https://dashboard.cypress.io/projects/9vk5ep/runs/1
```

```
Running: apex_110\app110.login.js (1 of 34)
```

APEX login page

```
✓ simple login (3564ms)
```

1 passing (4s)

(Results)

```
Tests: 1
Passing: 1
Failing: 0
Pending: 0
Skipped: 0
Screenshots: 0
Video: true
Duration: 3 seconds
Spec Ran: apex_110\app110.login.js
```

```
npx cypress run --config videoRecording=false --spec "C:\..\app101\helloWorld.js"
```

Dashboard

- web-based optional component
- Visualizes and records your runs
- Stores recordings and snapshots
- Runs tests faster
- Efficiency with parallelization
- Automatic load balancing

The screenshot shows the Cypress Dashboard interface. At the top, there's a navigation bar with a user profile icon, the text "Your personal orga...", and a "View all projects" link. Below this is a sidebar with links for "Latest runs", "Analytics", "Run status", "Run duration", "Test suite size", "Top failures", "Slowest tests", and "Project settings". The main content area has tabs for "Specs (34)", "Failures (1)", and "Test Results (11)". A message at the top of the results table says "All specs are complete!". Below this, a section titled "Unnamed" displays a list of test files: "apex_110\app110.login.js", "apex_110\app110.loginAssert.js", "apex_110\app110.loginCommon.js", "apex_110\app110.loginModal.js", "apex_110\app110.loginWeb.js", "apex_110\app110.loginWebIf.js", "apex_110\app110.loginWebAssertions.js", and "apex_110\app110.loginWebAssertionsPosit.js". Each file has a green checkmark icon next to it. To the right of the test list is a detailed timeline showing the execution of each spec, with bars indicating the start and end times for each run.

Getting Started with Cypress Dashboard ➔ [Inbox](#)

Success @ Cypress.io <success@cypress.io> [Unsubscribe](#)
to me ▾

Hi there,

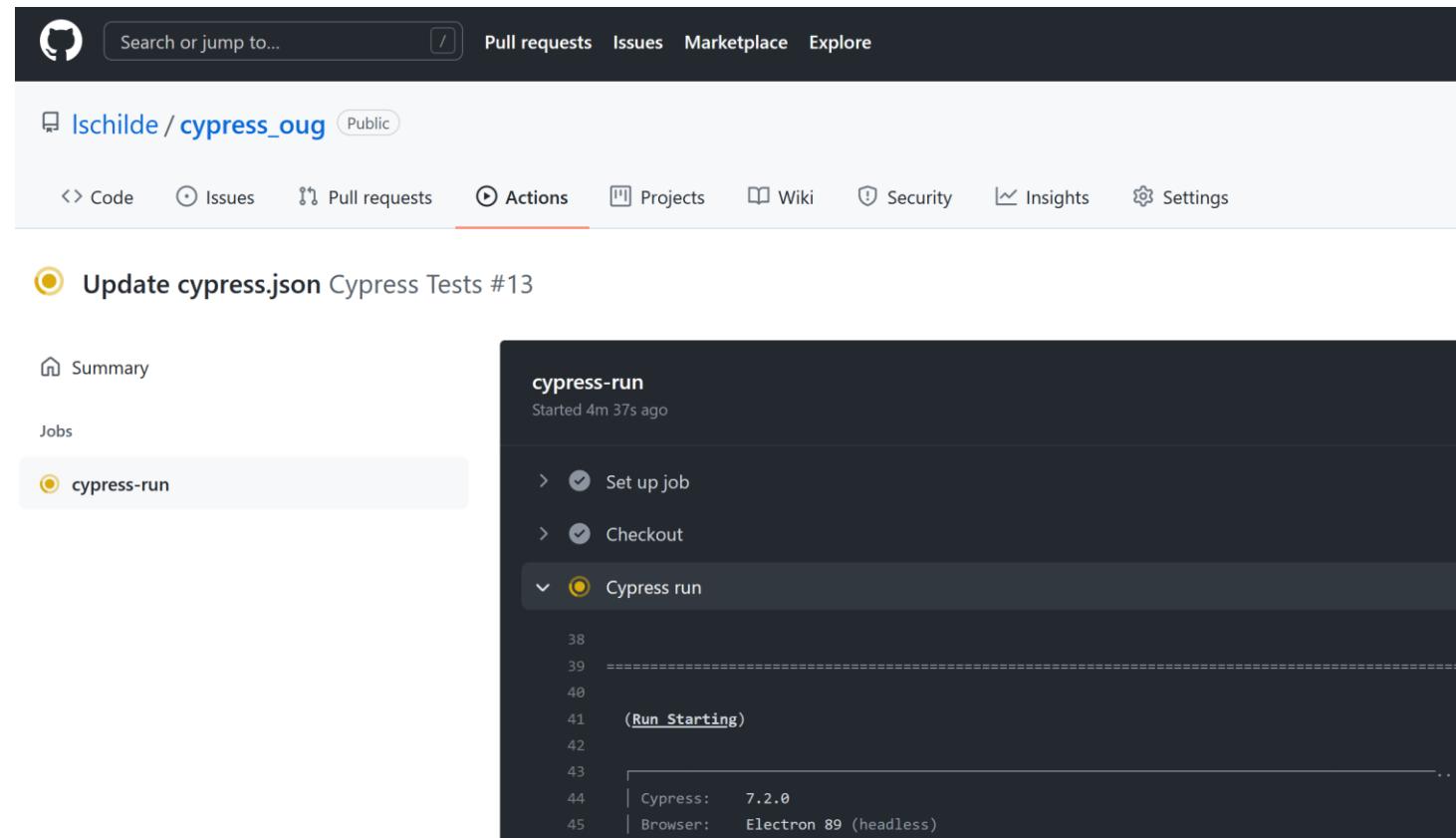
Thanks for signing up for the Cypress Dashboard!

Using the Dashboard will accelerate your testing and unlock insights for your team.

Continuous Integration

- Cypress is compatible with all Continuous Integration (CI) providers

- Integrates with SLACK, GitHub, GitLab, Circle CI



Recorder - beta

Cypress Chrome Recorder

Featured 2.9 ★ (16 ratings)

Extension Developer Tools 9,000 users

Add to Chrome

Watch Later Share

How to use @cypress/chrome-recorder

Quickly create Cypress tests from Chrome DevTools Recordings

How to use the Cypress Chrome Recorder

Watch on YouTube

Overview

This extension allows you to quickly export recordings from Cypress UI tests.

Details

Version 1.1.2

Offered by chromedev

Developer Cypress.io

101 Marietta St NW Suite 2502 Atlanta, GA 30303 US

Trader This developer has identified itself as a trader per the definition from the European Union.

Updated 21 April 2023

Size 33.64KiB

Email

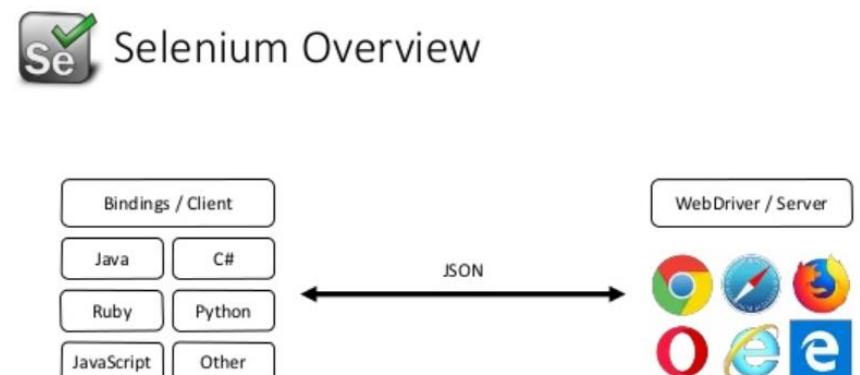
Report a concern

Languages English (United States)

```
→ ~ npx @cypress/chrome-recorder ~/Downloads/{fileName}.json -o=cypress/tests/ui
```

vs. Selenium?

- **Different architecture** (Library vs complete framework)
- **Easier** to set/get test **video recording**/snapshots
- Does not use WebDriver and **runs test directly in the browser**
- 'record your steps' is not supported
- JS vs 9 prog. languages (JS, Java, Perl, PHP, Python, Ruby, and C#..)
- Faster than Selenium
- Not as flexible



Ghost Inspector

Ghost Inspector - Installation

- Installable as Browser add on for Firefox and Chrome



The screenshot shows a web browser window displaying the Ghost Inspector documentation at <https://ghostinspector.com/docs/getting-started/>. The browser's address bar and toolbar are visible. In the toolbar, there is a Ghost Inspector icon (a white ghost inside a blue square) which is highlighted with a red rectangular box. Below the toolbar, a video player interface is shown with a play button, a progress bar indicating 03:59, and the word "vimeo" next to it.

How it works

1. Sign up

To get started, first [sign up for a free trial account](#).

2. Test recorder

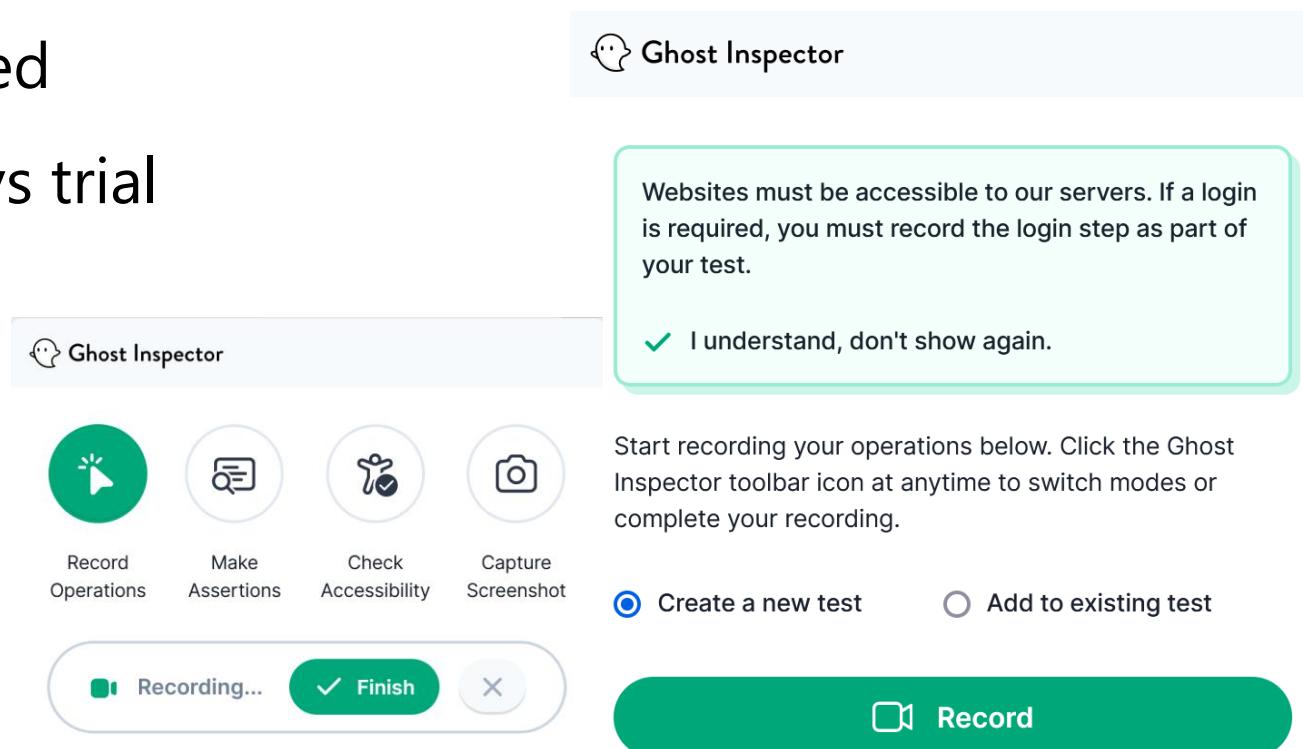
To create tests, you can install our [test recorder](#) in your browser:

[!\[\]\(0a6396f82ac034da0bd772445acbfca2_img.jpg\) Chrome Extension](#) [!\[\]\(cc2f6802f9924c0a41f27e348a1ac693_img.jpg\) Firefox Extension](#)

After installing the extension, a Ghost Inspector icon will appear in your toolbar. Clicking this icon will present you with the Ghost Inspector recording tool. You will need to log into your Ghost

Ghost Inspector – First run

- “Codeless testing”
- Allows you to record and play browser tests quickly and easy
- Very similar how Selenium IDE worked
- No Open Source version only 14 days trial
- We must buy a license



Ghost Inspector - Dashboard

- Test recorder selectors are not 100% there but still great
- Offers familiar selectors CSS and XPath
- Easy to learn
- “Lives in a cloud”

The screenshot shows the Ghost Inspector dashboard interface. At the top, there's a navigation bar with the Ghost Inspector logo, a trial message ("You have 10 days remaining in your trial"), and a "Buy now" button. Below the navigation is a breadcrumb trail: "Dashboard > LinoTests". The main area is titled "LinoTests" and displays a table of test results. The columns are labeled: TEST NAME, STATUS, SCREENSHOT, SCHEDULE, and TESTED. The table contains five rows of test data:

TEST NAME	STATUS	SCREENSHOT	SCHEDULE	TESTED
Create new customer	Passing	Failing	Suite schedule	03/24/22 5:01 PM
Create new customer (timestamp)	Passing	Passing	Suite schedule	03/24/22 5:20 PM
Create new customer Error and fix	Passing	Passing	Suite schedule	03/24/22 3:56 PM
Fix Demo Demo error	Import Only	Unknown	N/A	N/A
Invalid Password	Passing	Passing	Suite schedule	03/24/22 2:30 PM

LCT – Low code testing

LCT – Installation/documentation

- Installation provided by the team
- Easy to get started as training is provided

 Low Code Testing Blog Videos

Testing APEX Apps is now as easy as creating them.



LCT Oracle APEX Testing | Demo Video (12.6 MB)

No more struggling with complex testing frameworks or extensive test code. LCT provides a user-friendly, low-code interface to effortlessly create tests for your Oracle APEX Apps.

 LCT - Testing Oracle APEX
@lct-apex 51 subscribers 9 videos
LCT aims to make testing Oracle APEX apps as easy as creating them. >

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Latest Popular Oldest

 IR/IG Steps LCT 7:24	 Testing APEX Apps is now as easy as creating them LCT 4:23	 Check for Errors LCT 8:11	 Release 0.18 LCT 3:57 https://lct.software
Steps for Interactive Grid and Report Oracle APEX Testing with... 52 views • 4 weeks ago	LCT Oracle APEX Testing - Quick Demo 49 views • 1 month ago	Assert No Visible Errors in Oracle APEX Testing with LCT 78 views • 2 months ago	LCT Release 0.18 Showcase 107 views • 3 months ago
 Trace Viewer LCT	 Test Suites LCT	 Data Preparation LCT	 Variables LCT

<https://lct.software>

LCT – First run

- “Codeless testing”
- Configuring test steps using APEX app
- No recording steps option
- Professional service so support is provided
- Loads of output options

The screenshot shows the LCT - Low Code Testing application interface. The left sidebar contains navigation links: Worksheets, Queue, Import/Export, Test Suites, and Administration. The main area displays a table titled "General" with columns for App ID and Name. The table lists three entries:

	App ID	Name	Results - Percent Successful
	100	Lino Sample app March 2023	9.1 %
	100	Demo - Sample DB Fabian Dev	100.0 %
	100	Sample Test - Sample DB	95.0 %

Log Out

Worksheet

Edit Case

1 - 8 of 8

Select Step

Test Files



Search: All

Filter Category

- Assertion (4)
- Context Change (1)
- Documentation (1)
- Element Interaction (5)
- External (1)
- Keyboard Interaction (1)
- Mouse Interaction (2)
- Navigation (1)
- Visibility (2)
- Waiting (3)

File Name

- execution_variables.js
- lct.spec.js
- lctReporter.js
- playwright.config.js
- pwUtils.js
- run.sh
- test_util.js
- output.log
- results.xml
- after_customer_search.png
- after_login.png
- amount_chosen.png

Sample Database Application

Customers

Customer Name	Address	City	State	Postal Code
Vega, Vincent	-	-	HI	35005

*/

results.xml' }]],

after_customer_search.png

Sample Database Application

```
use: {
  ...devices['Desktop Chrome'],
},
```

Summary

- Learn about selectors and standardize them
- Assertions - Verifying expected actually happened
- Flakiness

Summary

- Make AAT part of your dev lifecycles
 - Reduces time and costs testing your apps
- LCT, Ghost Inspector or Cypress (Playwright or Selenium)
 - You can't go wrong
- Integrate well with any CI tools
- Standardize methods
 - Reuse and share code
 - Passwords
 - Input Variables
 - Consider changing your APEX apps

Q&A

Thank you for attending



Resources

- <https://docs.cypress.io/>
- https://www.youtube.com/watch?v=QFzN_0soxiQ
- <https://dzone.com/articles/selenium-vs-cypress-is-webdriver-on-its-way-out-1>
- <https://automationrhapsody.com/cypress-vs-selenium-end-era/>
- <https://github.com/cypress-io/cypress-realworld-app>
- <https://docs.cypress.io/guides/dashboard/github-integration>
- <https://lct.software>