

APEX automated testing

cypress



Ghost Inspector

hroug 22

Spring | Proljeće

17. - 19.05.2022., Tuhelj

Lino Schildenfeld

@LinoSchilde

lschilde.blogspot.com

APEX Office hours



Part 2: What's new in Oracle APEX 22.1

Marc Sewtz, Chaitanya Koratamaddi, Mónica Godoy

247 views · April 21, 2022



Part 1: Sneak Preview of What's New

Marc Sewtz, Chaitanya Koratamaddi, Mónica Godoy

2,188 views · April 14, 2022

apex.oracle.com/officehours

AUSOUG APEX News



[Become a Member](#) [What's On](#) [Branches](#) [Resources](#) [Past Events](#) [Blog](#) [My Account](#)



AUSOUG
APEX Special Series 2022
ā'pěks
(#orclapex)
2022 SPECIAL SERIES

www.ausoug.org.au/apex-special-series-2022/

My story



- AUSOUG APEX webinars
- NZ APEX meetup organizer
- APEX blogger
- Conference speaker
- Developer since 2006

@LinoSchilder



Automated testing tools

- SELENIUM
- **CYPRESS**
- ATAF
- utPLSQL
- Codeconcept.io
- JMeter
- Katalon
- Gatling
- HammerDb
- Appium
- Autoit
- Eggplant
- Playwright
-

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

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

WHY CYPRESS

- **Easy to get going** and **learn** how to use
- **Automatically waits** (for commands and assertions before moving on*)
- It can video **record** the entire **tests** & takes snapshots
- **Good documentation**

- Version 8.4.0 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 -D (--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 command line
- Run `npx cypress open`
- Check your `package.json` for correct dependencies
- Notice cypress subdirectory (rm examples)
- Create new directory under `integration`

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

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

Config check

```
{ } package.json > ...  
1 {  
2   "name": "CYPRESS",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   ▶ Debug  
7   "scripts": {  
8     "test": "echo \"Error: no test specified\" && exit 1"  
9   },  
10  "keywords": [],  
11  "author": "",  
12  "license": "ISC",  
13  "devDependencies": {  
14    "cypress": "^5.1.0"  
15  }  
16 }
```

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...
```

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

Deeper look

Password

- Hide them! and disable log
- Environment variable or cypress.json

baseUrl

- Don't define new use default

5 ways to set environment variables

- In configuration file config.json
- Create Cypress.env.json
- Export as CYPRESS_*
- Pass in CommandLine as -env
- Set within test configuration

```
{  
  "baseUrl": "http://localhost:8080/ords/",  
  "env": {  
    "demoUsername": "demo",  
    "demoPassword": "demo"  
  }  
}
```

Share code

```
beforeEach(() => {
  // root-level hook
  // runs before every test
})

describe('Hooks', () => {
  before(() => {
    // runs once before all tests in the block
  })

  beforeEach(() => {
    // runs before each test in the block
  })

  afterEach(() => {
    // runs after each test in the block
  })

  after(() => {
    // runs once after all tests in the block
  })
})
```

```
Cypress.Commands.add("login", () => {
  cy.server();
  cy.visit("f?p=110");
  cy.url().should("contain", "110:LOGIN_DESKTOP");
  cy.get('.t-Login-header').should("contain", "Sample Database Application");
  cy.get('#P101_USERNAME')
    .type("demo")
    .should("have.value", "demo");
  cy.get('#P101_PASSWORD').type("demo",{log:false});
  cy.get('#P101_LOGIN').click();
  cy.url()
    .should("contain", "110:1")
    .then($url => {
      window.loggedInPage= $url;
    });
  cy.getCookie("ORA_WWV_PACKAGED_APPLICATIONS").then($cookie => {
    window.appCookie = $cookie.value;
  });
});
```

Fill a form - Assertions

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

Assertions

- Uses popular **Chai assertion** library (including Sinon and 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

Manual run

- Runs all tests 'heedlessly'
- Ability to pass in parameters
- By default `cypress run` **records a video locally**
- Disable video with
`--config video[Recording]=false`
- Record into Dashboard with
`--record --key`

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

Option
<code>--headed</code>
<code>--headless</code>
<code>--help, -h</code>
<code>--key, -k</code>
<code>--no-exit</code>
<code>--parallel</code>
<code>--port, -p</code>
<code>--project, -P</code>
<code>--quiet, -q</code>
<code>--record</code>

(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
```

Dashboard

- Optional web-based companion to the Test Runner
- Visualizes and records your runs
- Stores recordings and snapshots
- Runs tests faster
- Efficiency with parallelization
- Automatic load balancing

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.

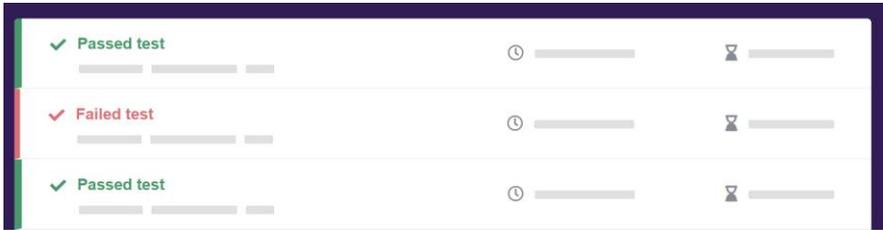
👤 0 📄 0 🟢 10 🚫 1

The screenshot displays the Cypress Dashboard interface. On the left is a dark sidebar with navigation options: 'Your personal orga...' (L. Schilde), 'CYPRESS' (View all projects), 'Latest runs', 'Analytics' (Run status, Run duration, Test suite size, Top failures, Slowest tests), and 'Project settings'. The main content area shows a test run summary for 'Unnamed' with the message 'All specs are complete'. It reports 25 specs cancelled, 8 specs passed, and 1 spec failed. Below this is a detailed spec execution timeline for 'apex_110\app110'. The timeline shows 10 specs, with 9 passing (green bars) and 1 failing (red bar). The failing spec is 'apex_110\app110.LoginPwd_AssertionsPost.js' which failed at 00:09. A notification at the top of the timeline suggests 'Make your tests run 2 minutes faster by adding 8 machines in CI.' and provides a 'See full analysis' link.

Spec Name	Status	Duration	Actions
apex_110\app110.Login.js	Pass	00:03	📄 📷 🔄
apex_110\app110.LoginBaseURL.js	Pass	00:02	📄 📷 🔄
apex_110\app110.LoginCommon.js	Pass	00:11	📄 📷 🔄
apex_110\app110.LoginPwd.js	Pass	00:04	📄 📷 🔄
apex_110\app110.LoginPwdBad.js	Pass	00:04	📄 📷 🔄
apex_110\app110.LoginPwdObj.js	Pass	00:03	📄 📷 🔄
apex_110\app110.LoginPwd_Assertions.js	Pass	00:04	📄 📷 🔄
apex_110\app110.LoginPwd_AssertionsPost.js	Fail	00:09	📄 📷 🔄

Recordings

- records a video for each spec file when running tests during `cypress run`



You could see test recordings here!

Connect to Cypress Dashboard for free:

- Record test runs in CI and debug failed tests with ease
- Understand the health of your tests with test analytics
- Improve testing efficiency with parallelization, load balancing, and more

[Connect to Dashboard](#)

After logging in, you'll see recorded test runs here and in your Cypress Dashboard.

```
cypress run --record --key 5b5a9e17-19e0-4dca-8d37-bc4
```

Set up project

What's the name of the project?

(You can change this later)

CYPRESS_DEMO1

Who should own this project? ?

[Manage organizations](#)

Your personal organization

Who should see the runs and recordings? ?

Public: Anyone has access.

Congrats! You recorded your first run 🎉

Success @ Cypress.io <success@cypress.io>

to me ▾

Hi L,

Congratulations on recording your first test run to the Cypress Dashboard!

Each recording unlocks powerful insights into each of your test runs.

Recording runs in CI can help you dig even deeper, allowing you to:

- See all the tests running in your CI pipeline
- Find the precise error that resulted in a failed test
- Watch a video recording of failed tests

Want to supercharge your testing and increase deployment velocity?

Check out our guide to running Cypress tests in [Continuous Integration](#).

Happy Testing!

- Cypress Success Team

[Set up project](#)

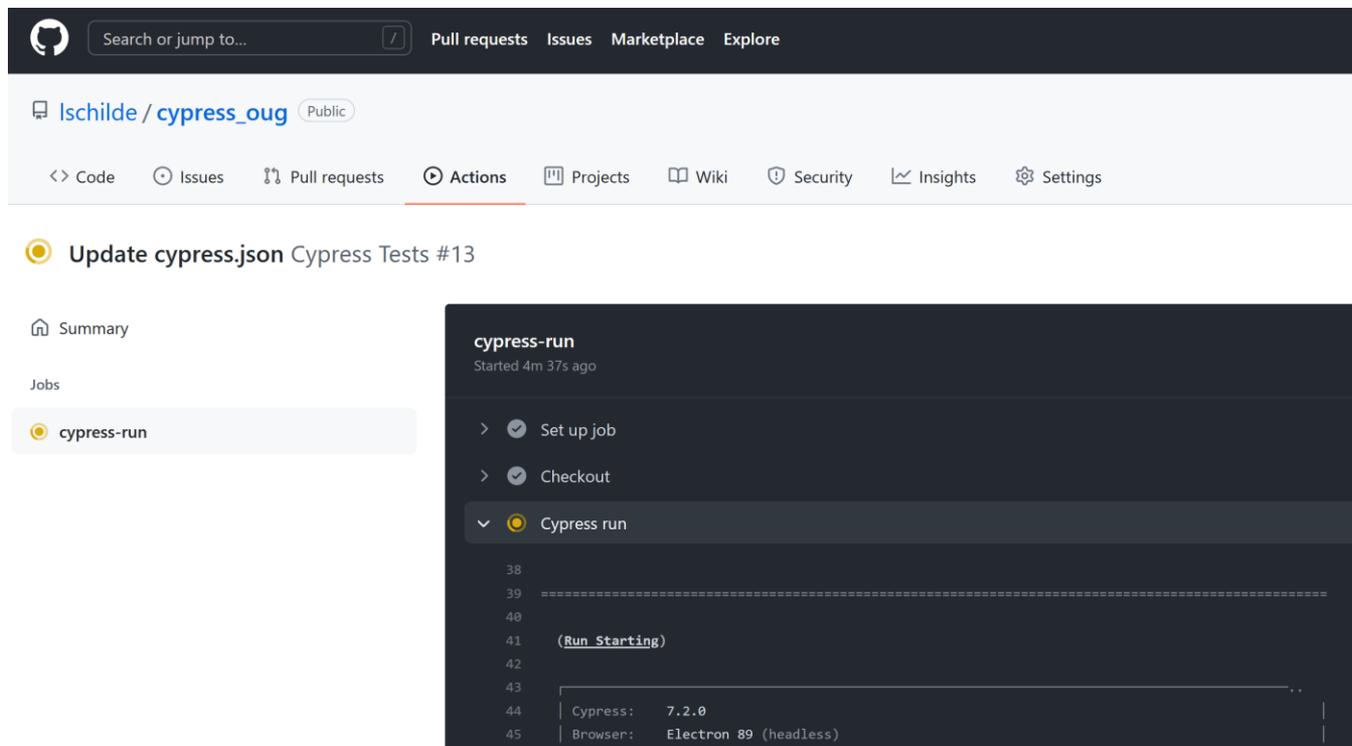
Test results

(Run Finished)

Spec	Tests	Passing	Failing	Pending	Skipped
✓ apex_110\app110.login.js	00:03	1	1	-	-
✓ apex_110\app110.loginBaseUrl.js	00:02	1	1	-	-
✓ apex_110\app110.loginCommon.js	00:11	2	2	-	-
✓ apex_110\app110.loginPwd_Assertions.js	00:04	1	1	-	-
✗ apex_110\app110.loginPwd_AssertionsPost.js	00:09	1	-	1	-
✓ apex_110\app110.loginPwd.js	00:04	1	1	-	-
✓ apex_110\app110.loginPwdBad.js	00:04	1	1	-	-
✓ apex_110\app110.loginPwdObf.js	00:03	1	1	-	-
✓ apex_110\app110.p2aMultiple.js	00:10	2	2	-	-
✓ apex_110\app110.p2BypassLoginX.js	00:10	1	1	-	-

Continuous Integration

- **Cypress is compatible with all Continuous Integration (CI) providers and systems**
- Integrates with SLACK, GitHub, GitLab, Circle CI

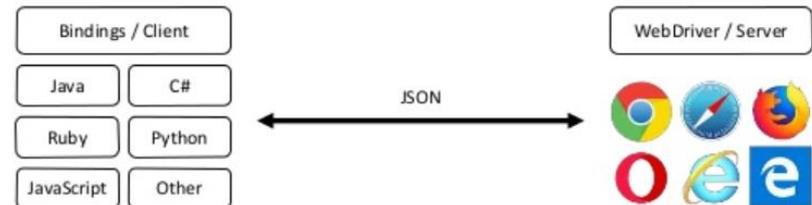


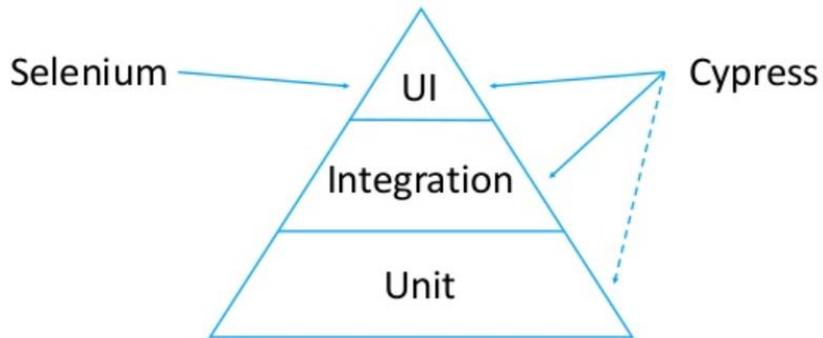
The screenshot shows a GitHub Actions workflow run for the repository 'lschilde / cypress_oug'. The workflow is titled 'Update cypress.json Cypress Tests #13'. The 'Actions' tab is selected, showing a list of jobs: 'Summary', 'Jobs', and 'cypress-run'. The 'cypress-run' job is expanded, showing a list of steps: 'Set up job', 'Checkout', and 'Cypress run'. The 'Cypress run' step is expanded, showing the output of the Cypress command. The output indicates that Cypress 7.2.0 is being used with Electron 89 (headless).

```
38
39
40
41 (Run Starting)
42
43
44 Cypress: 7.2.0
45 Browser: Electron 89 (headless)
```

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#..)
- **Fewer browser support** (cross browser testing limitation)
- Faster than Selenium
- Not as flexible
- Still market place for both





Selenium

- Java
- C#
- Ruby
- Python
- JavaScript
- PHP, Perl, Golang, Haskell, Objective-C, R, Elixir, etc.

Cypress

- JavaScript

Selenium

- ID
- Name
- Class Name
- Tag Name
- (Partial) Link Text
- CSS Selector
- XPath Selector

Cypress

- jQuery

Selenium



And many more ...

Cypress



<https://automationrhapsody.com/cypress-vs-selenium-end-era/>

Testing tool - Ghost Inspector



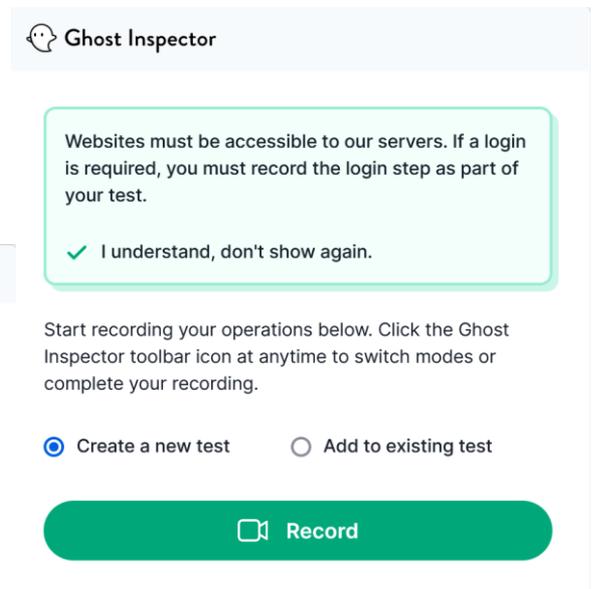
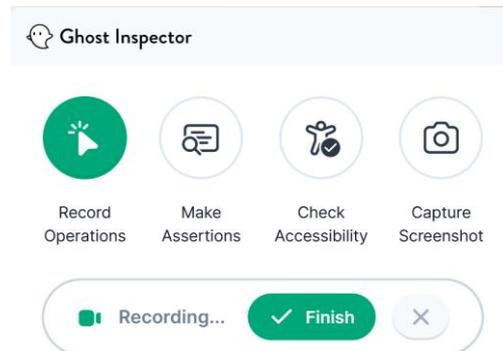
Ghost Inspector

Getting Started



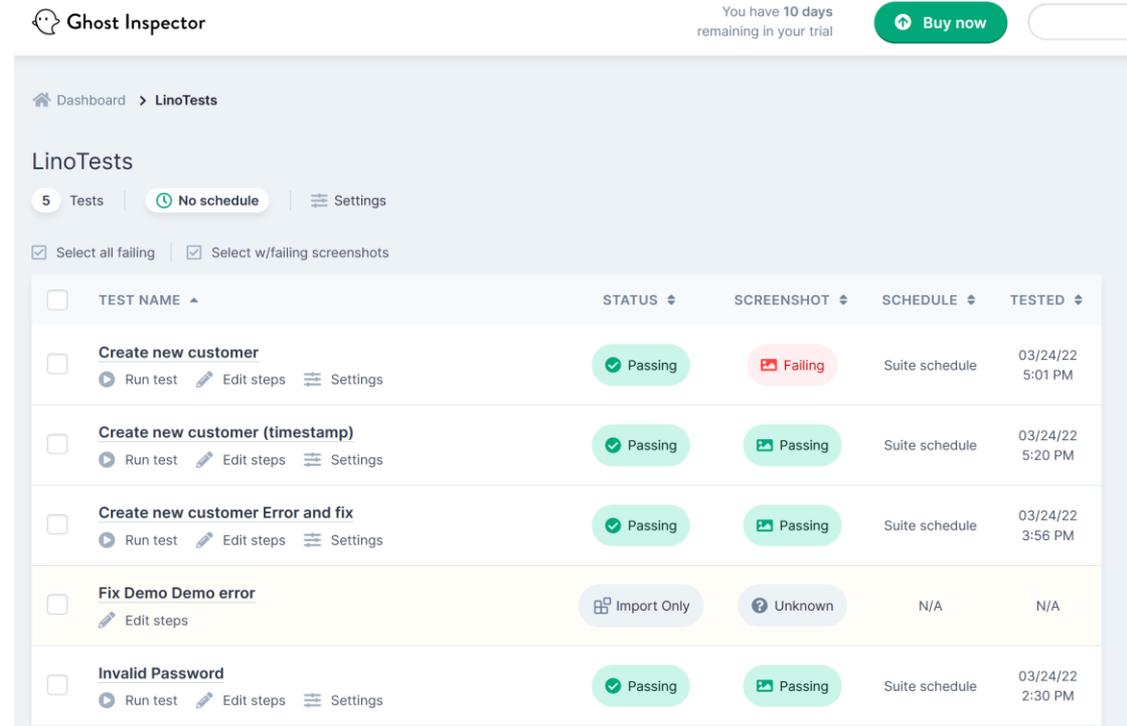
Why - Ghost Inspector

- Installable as Browser add on for Firefox and Chrome
- "Codeless testing"
- Allows you to record and play browser tests quickly and easy
- Very similar how Selenium IDE worked
- No opensource version only 14 days trial
- Must buy a license



Testing tool - Ghost Inspector

- Test recorder selectors are not 100% there but still great
- Offers familiar selectors
- Easy to learn
- There is no local execution



The screenshot displays the Ghost Inspector web interface. At the top, there is a navigation bar with the Ghost Inspector logo, a trial status indicator ('You have 10 days remaining in your trial'), and a 'Buy now' button. Below the navigation bar, the breadcrumb path is 'Dashboard > LinoTests'. The main content area is titled 'LinoTests' and shows a summary of 5 tests, with a 'No schedule' indicator and a 'Settings' link. There are two checkboxes: 'Select all failing' (checked) and 'Select w/failing screenshots' (checked). Below this is a table of test results.

<input type="checkbox"/>	TEST NAME ▲	STATUS ▾	SCREENSHOT ▾	SCHEDULE ▾	TESTED ▾
<input type="checkbox"/>	Create new customer Run test Edit steps Settings	Passing	Failing	Suite schedule	03/24/22 5:01 PM
<input type="checkbox"/>	Create new customer (timestamp) Run test Edit steps Settings	Passing	Passing	Suite schedule	03/24/22 5:20 PM
<input type="checkbox"/>	Create new customer Error and fix Run test Edit steps Settings	Passing	Passing	Suite schedule	03/24/22 3:56 PM
<input type="checkbox"/>	Fix Demo Demo error Edit steps	Import Only	Unknown	N/A	N/A
<input type="checkbox"/>	Invalid Password Run test Edit steps Settings	Passing	Passing	Suite schedule	03/24/22 2:30 PM

Testing tool - Ghost Inspector

- Saves videos and snapshots
- Supports geolocations
- Multiple screen sizes
- Multiple browser versions
- It can send notifications to:
 - Emails
 - REST webhook
 - MS Teams, Slack or PagerDuty

The screenshot displays the Ghost Inspector interface for creating a new customer. The main window shows a test scenario with two steps, both marked as 'Passing'. Step #1 involves clicking on a username input field. Step #2 involves assigning a value 'demo' to the same input field. The interface includes a 'Start URL' field with a long URL, and various action buttons like 'Add condition', 'Move', 'Copy', 'Add above', 'Add below', and 'Delete'. A dropdown menu is open, showing options for 'Click', 'Mouse over', 'Drag and drop', 'Keyboard', and 'Other'. The 'Click' option is selected, and the 'Left Click' radio button is checked. The interface also shows a 'Cancel' button and a 'Save changes' button.

Summary

- Make AT be part of your dev lifecycles
 - Reduces time and cost testing your apps
- Cypress, Ghost Inspector or Selenium
 - You can't go wrong
- Learn and know your selectors
- Protect your passwords
 - Using environment variables
- Standardize methods
 - Reuse and share code

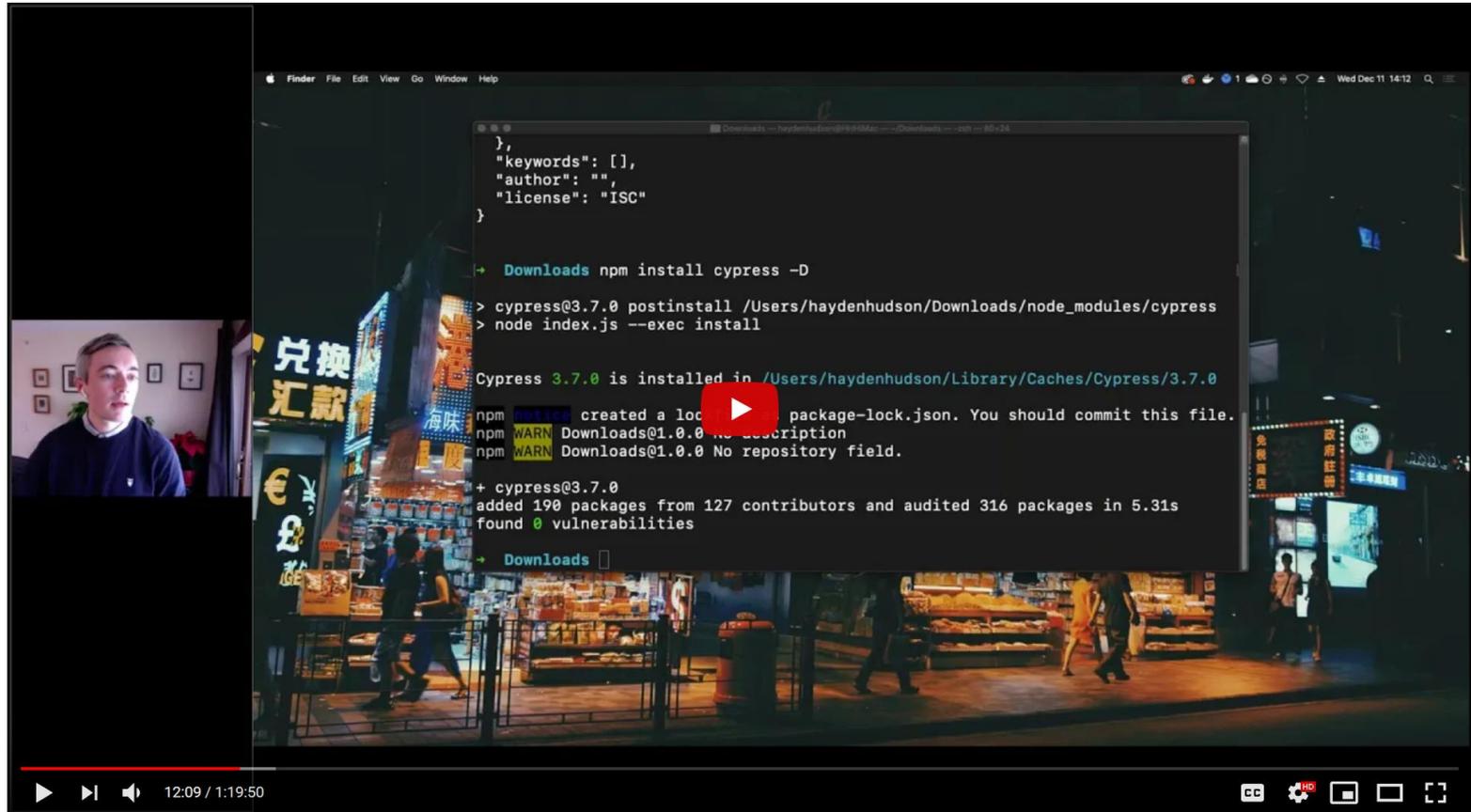
Q&A

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://ghostinspector.com/docs/api/>



Hayden Hudson - https://www.youtube.com/watch?v=QFzN_0soxiQ