End-to-End Testing Tools

Demo on how to use Cypress and Cucumber to test front-end and back-end code.

Lakshi Villavarayen,
Senior Quality Engineer

Manale Henini,
Software Engineer I

PROS

Why test?

Thorough testing is necessary to discover bugs/defects that need to be resolved prior to delivery to client. This ensures the dependability, security, and quality of the software.

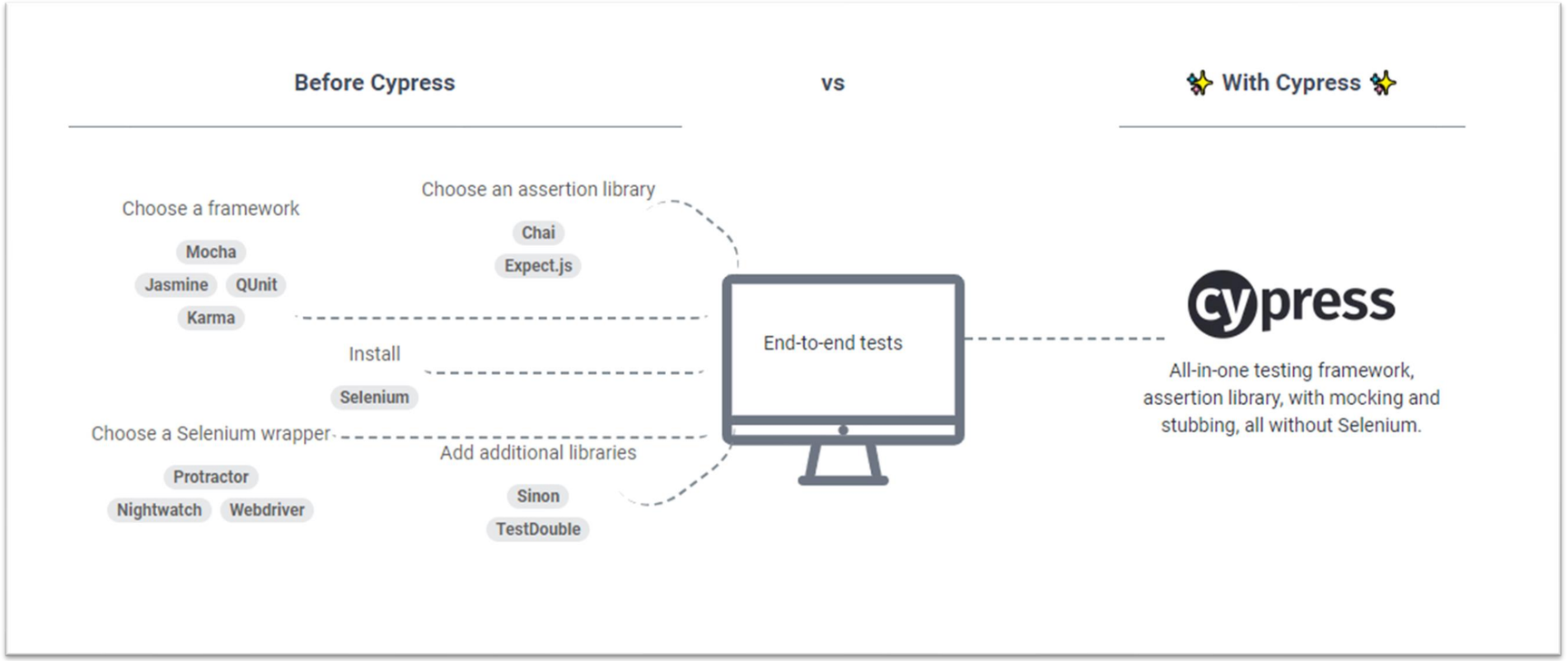
PROS

CYPRESS

PROS

What’s Cypress?

A complete-end-to-end testing experience



Why Cypress?

- Works on any front-end framework or website.
- For Developers and QA Engineers.
- Supports Time Travel and Real Time Reloads.
- Cypress runs much, much faster.
- With Cypress it’s possible to:

- Set up tests

Write tests

Run tests

Debug tests

Getting Set Up.

Install Node (version 14)

Install Cypress

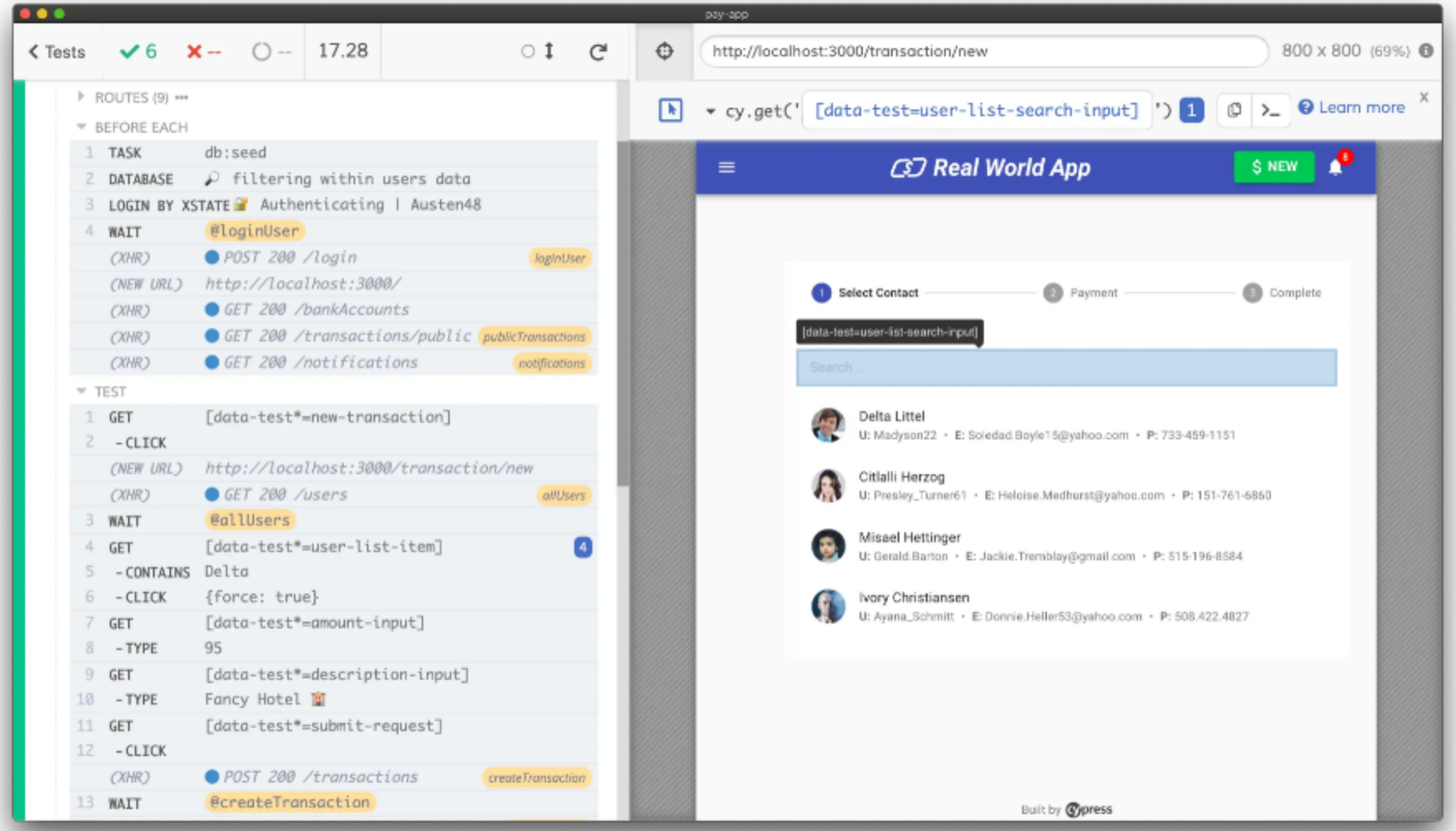
Install the cypress-realworld-app

Test status menu

See how many tests passed or failed, and how long they took to run

Viewport sizing

Test responsive layouts by changing your app's viewport size



Command Log

Hover over a command in the log and the app preview shows you exactly what it looked like when the test ran. Our users call it time travel ✨

App preview

While test commands execute, see what happens in your app in real time. Use your DevTools to inspect or debug each command

Cypress Resources

For more info on how Cypress works, including documentation and tutorials, please visit https://www.cypress.io/

For instructions on installing the Real World App please visit https://github.com/cypress-io/cypress-realworld-app

CUCUMBER

PROS

First, Behavior-Driven Development

- Also known as BDD
- An agile software development process
- Focuses on expected behavior
- Encourages collaboration across whole team
- Goal: agree on a how an application should behave based on concrete examples.

BDD Process

Discovery

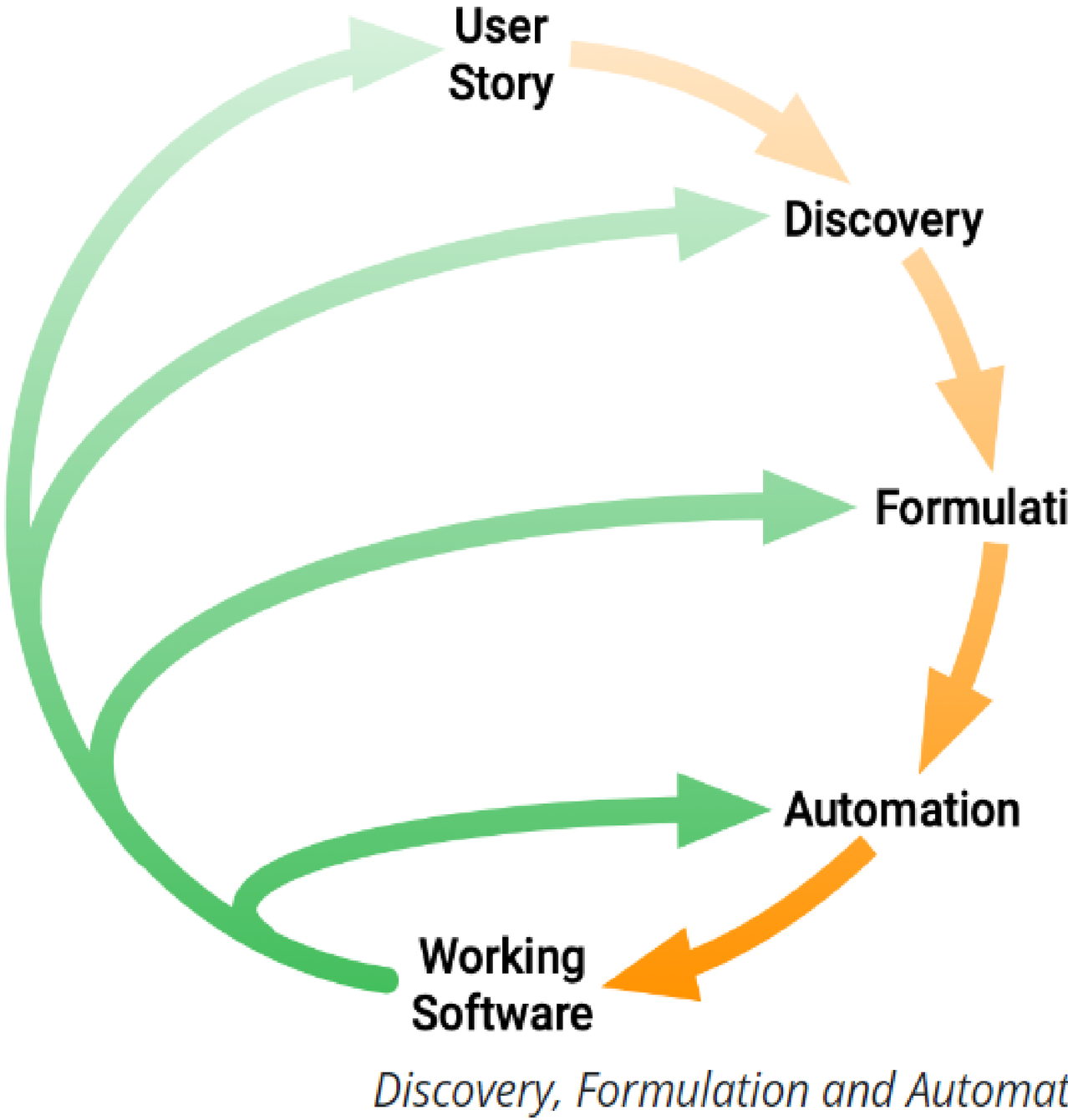
What it *could* do: Agree on concrete examples from User Story.

Formulation

What it *should* do: Document the examples.

Automation

What it *actually* does: Implement the documented behavior.



Discovery, Formulation and Automat

Now, What's Cucumber?

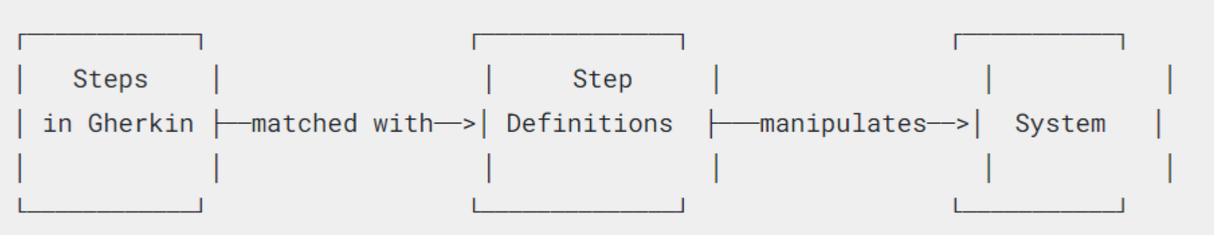
- Cucumber is a tool that supports the BDD process.
- Reads plain text specifications and verifies software follows them.
- Specifications are made of *examples,* or *scenarios*, ex:

```
Scenario: Breaker guesses a word
  Given the Maker has chosen a word
  When the Breaker makes a guess
  Then the Maker is asked to score
```

- Scenarios are made of *steps*.

Cucumber Scenarios

- Scenarios have 2 parts: Gherkin syntax and Step Definitions
- What's Gherkin?
  - A set of grammar rules to give plain text structure
  - Gherkin documents are stored in .feature text files
- What are Step Definitions?
  - Connect Gherkin steps to programming code



  - Can be written in many programming languages. JavaScript example:

```
When("{maker} starts a game", function(maker) {
  maker.startGameWithWord({ word: "whale" })
})
```

Cucumber Resources

- For more info on how Cucumber works, including documentation and tutorials, please visit https://cucumber.io/
- To access the complete code for the cucumber tutorial and these slides please visit https://github.com/mhenini/CucumberDemo

Or scan

Thank You

pros.com