

# End-to-end Angular Testing with Cypress

---

Introducing Cypress



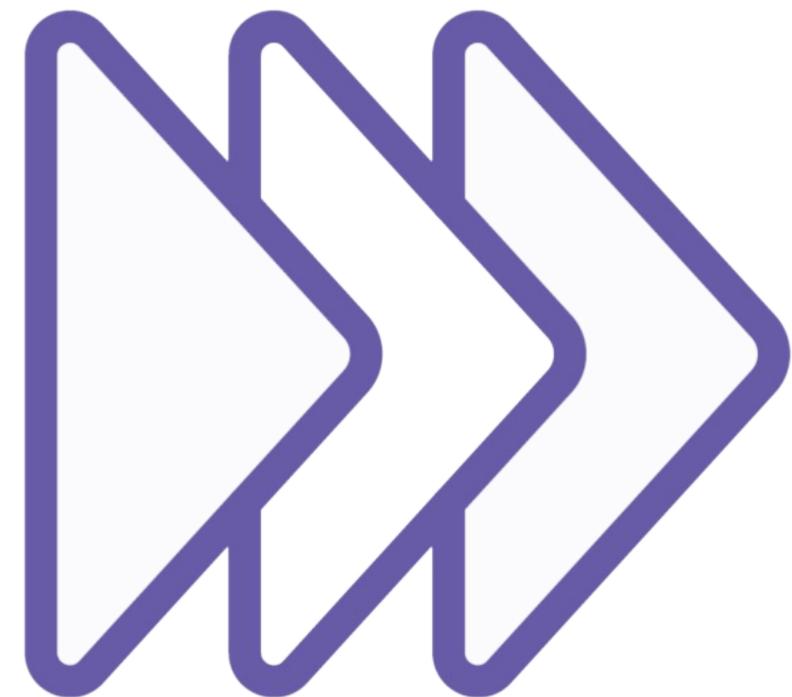
**Marko Vajs**  
Software Development Engineer in Test



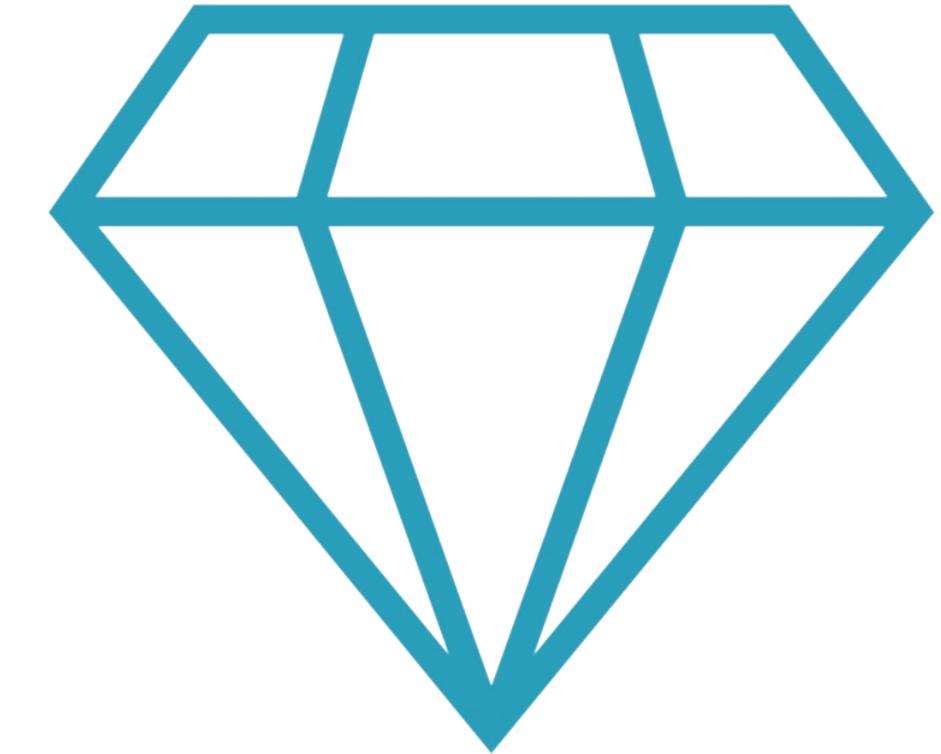
# Benefits of Automated Testing



**Provides early  
feedback on quality**



**Speeds up the  
development**



**Increases software  
quality**



You will learn how to write end-to-end automated tests using Cypress to test Angular applications.





**JavaScript  
end-to-  
end testing  
framework**

**Testing with  
Cypress is easy**

**Modern  
architecture**



# Course Overview



- Introduce Cypress
- Set up a development environment
- Learn how to locate elements on a page
- Learn how to interact with elements
- Explore core concepts of Cypress
- Use App Actions and Page Object Model
- Access Angular components from Cypress
- Use additional features of Cypress



# End-to-end Angular Testing with Cypress

---

Version Check



# Version Check



**This version was created by using:**

- Cypress 9.2.1
- Angular CLI 13.1.1
- Node 16.13.2



# Version Check



**This course is 100% applicable to:**

- Cypress 6.0 through Cypress 9.2.1
- Angular CLI 13.1.1
- Node 16.13.2



# Introducing Cypress

---

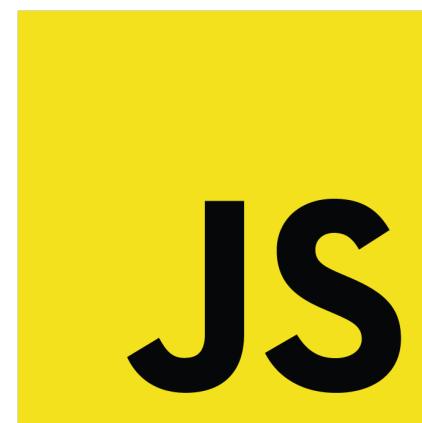




# What Is Cypress?

**JavaScript-based front-end testing tool.**





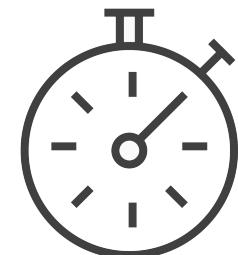
Cypress tests can be written in any language  
that compiles down to JavaScript.



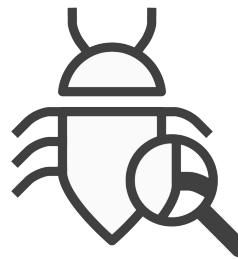
Cypress addresses many pain  
points when testing web  
applications.



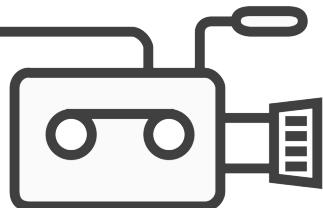
# Why Cypress?



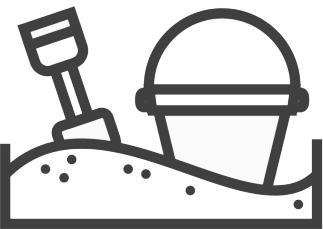
**Automatically waits for elements to become available**



**Simplifies debugging by taking snapshots of test execution**



**Saves videos and screenshots of test execution**



**Contains a built-in selector playground**



**Includes unique features because it runs directly in the browser**





**Cypress is not based on Selenium  
WebDriver**



**Cypress runs directly in the  
browser**



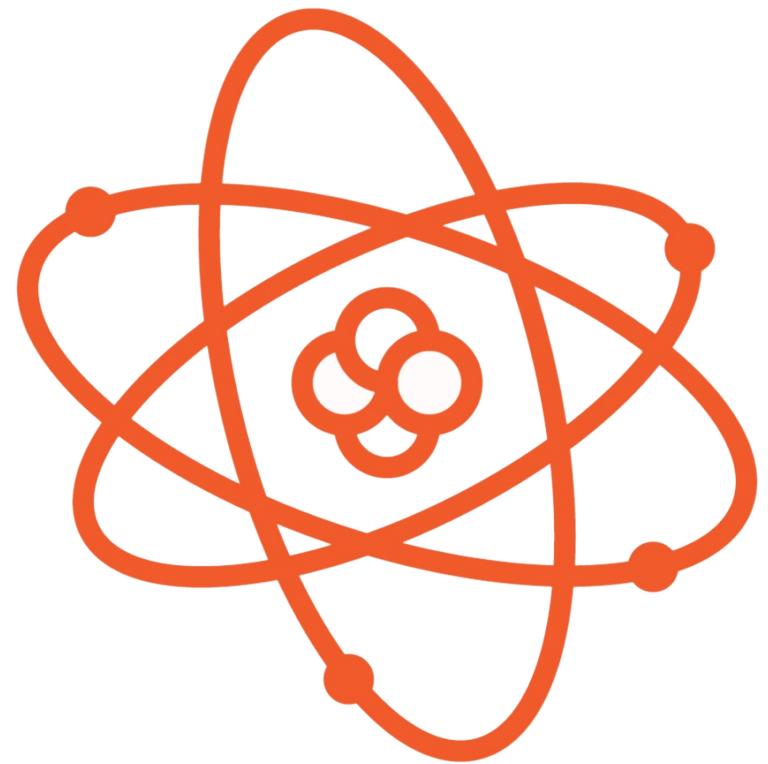
# Browser Support



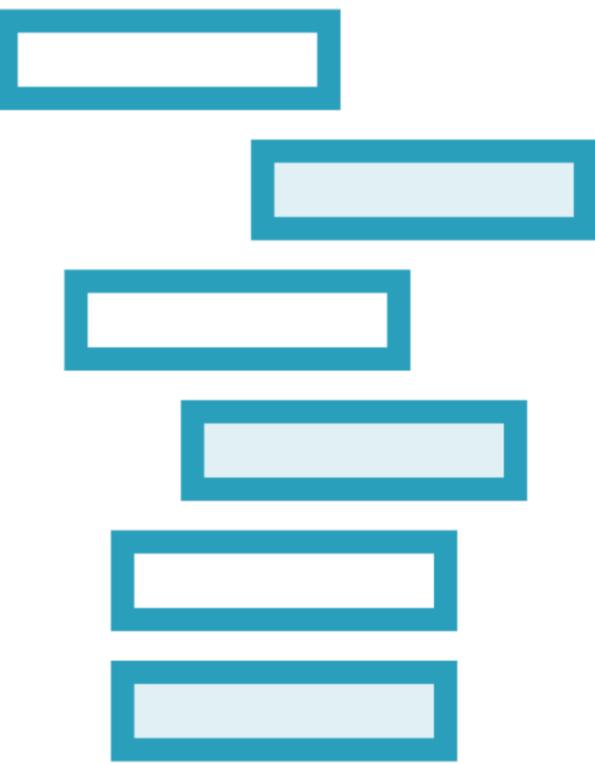
**Supports Chrome-family of browsers  
and Firefox**



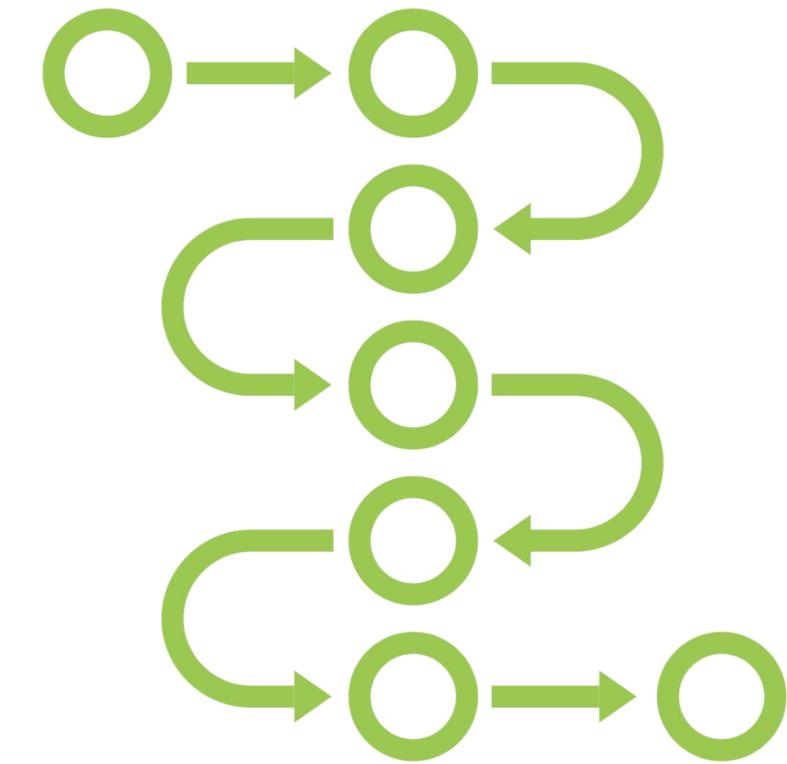
# Types of Tests



**Unit**



**Integration**



**End-to-end**

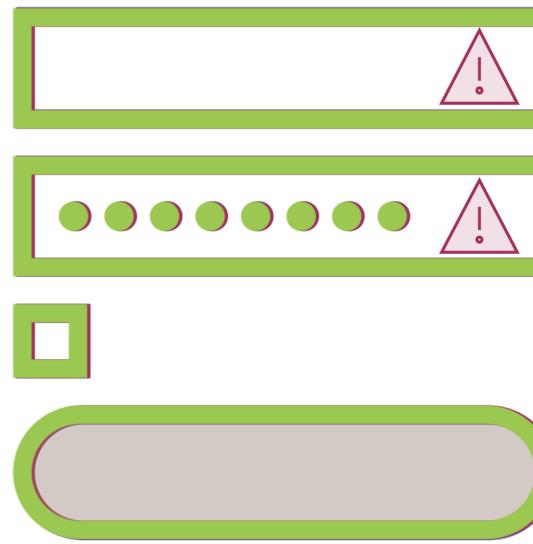


# End-to-end Testing

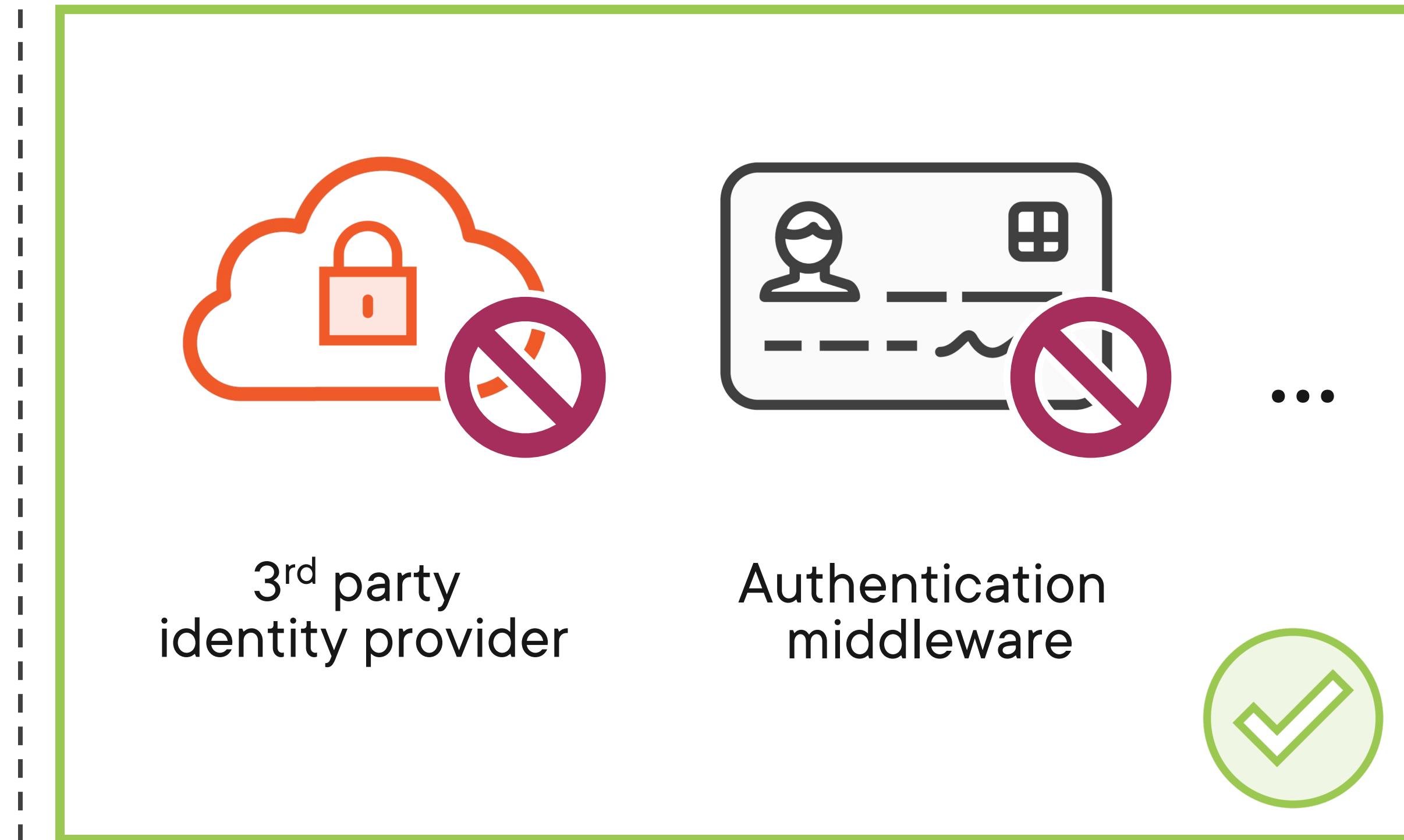
**A type of testing in which business processes are tested from start to finish under production-like circumstances.**



# Example of Login



Login page



**UI  
testing**



**End-to-end  
testing**



# Exploring Trade-offs

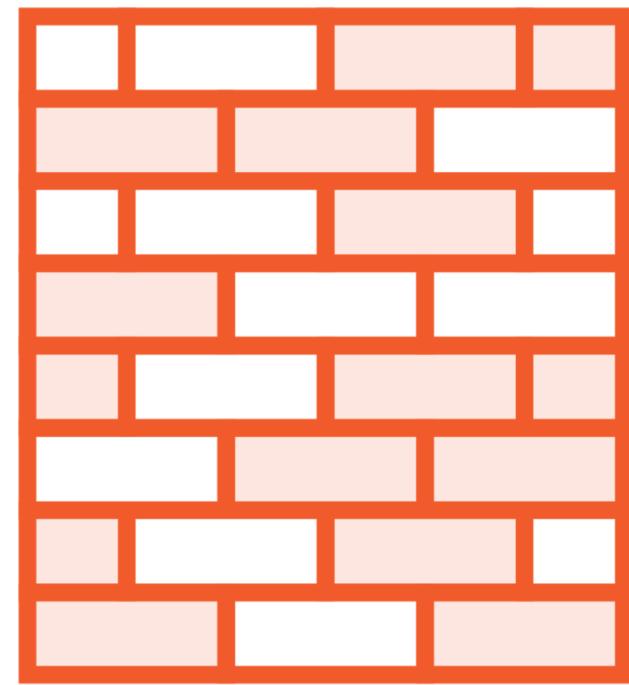
---



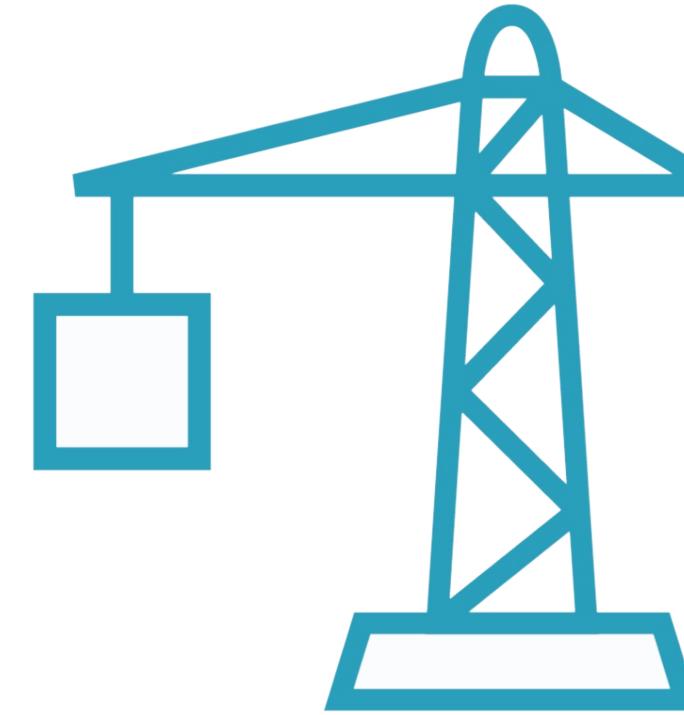
Trade-offs can be seen as good boundaries that are there to prevent us from writing bad tests.



# Types of Trade-offs



**Permanent**



**Temporary**



# Permanent Trade-offs



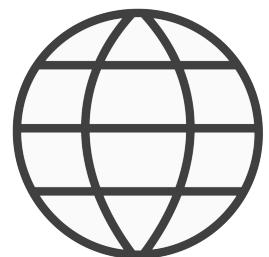
**Cypress is not used to index web pages, perform performance testing or to script 3<sup>rd</sup> party sites**



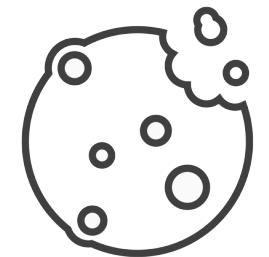
**Communication with back-end like server or database is limited**



**There is no multi-tabs support**



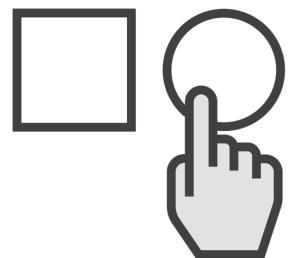
**Cypress does not support multiple browsers open at the same time**



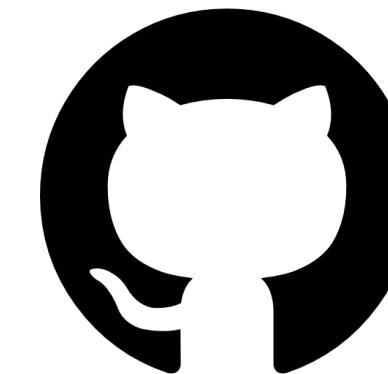
**Cypress tests are limited to the same origin**



# Temporary Trade-offs



**Workaround is required for a cy.hover() command**



GitHub page contains list of issues where the progress of temporary trade-offs can be tracked.





## Workarounds

It's better to re-think the test case design before looking for a workaround.

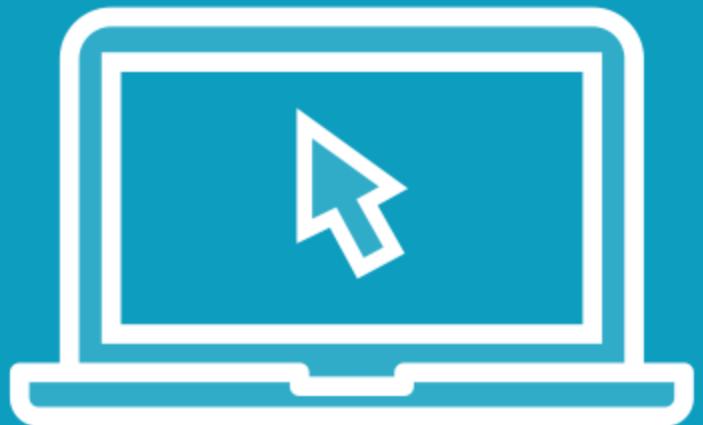


# Demo: Setting up Development Environment

---



Demo



**Install Node 16.13.2**

**Download and run demo application**

**Add Cypress using Angular CLI**



# Module Summary

---



# Module Summary



**Cypress is an end-to-end testing framework for web applications**

**Cypress has known trade-offs and some are permanent and intentional**

**There are workarounds for most of the trade-offs**

**Angular CLI supports adding Cypress to existing projects**



Up Next:  
Testing Angular Application

---

