

LORENZO SAVINI

Software developer

# Smoke testing con Cypress

Develer TPS-8

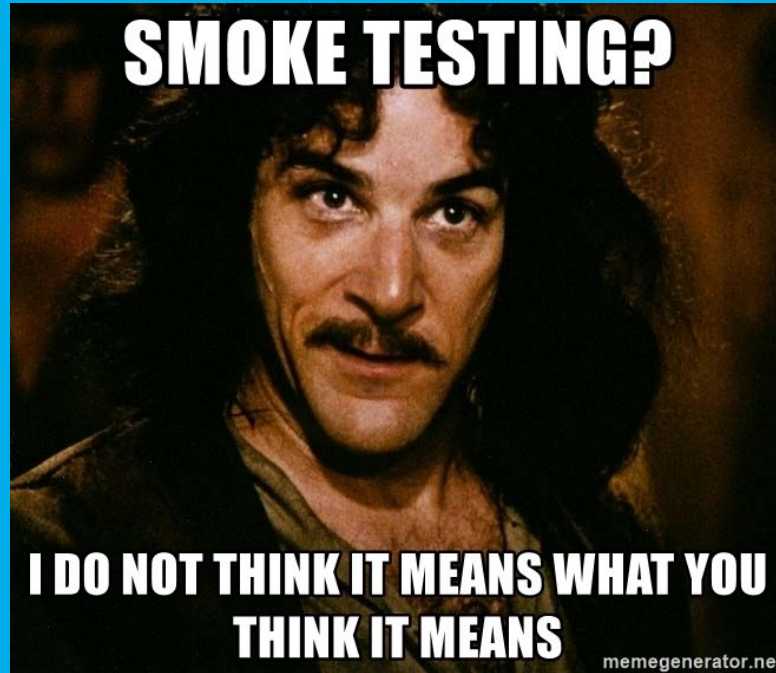
26/11/2020

The logo for Develer, featuring the word "develer" in a white, lowercase, sans-serif font. A horizontal orange bar is positioned above the letter "e".

## I TPS

- Le TPS (Technical Project Stories) sono incontri interni tra Develerian\*.
- Sono momenti di condivisione di idee ed esperienze.
- Parliamo di un po' di tutto: dai post mortem di incidenti in produzione o progetti conclusi, a tool che abbiamo usato in un progetto e che vogliamo far conoscere a colleghi di altri team.
- Le TPS rientrano nella 80 ore di formazione che Develer finanzia annualmente a ciascuno di noi.

## ■ Cos'è lo smoke testing?





Smoke testing is preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release.

[cit. [Wikipedia](#)]



# PROGRAMMA DELLA TPS

- | Cypress
- | Selenium? Puppeteer?
- | Distribuirlo ai team
- | Best practices

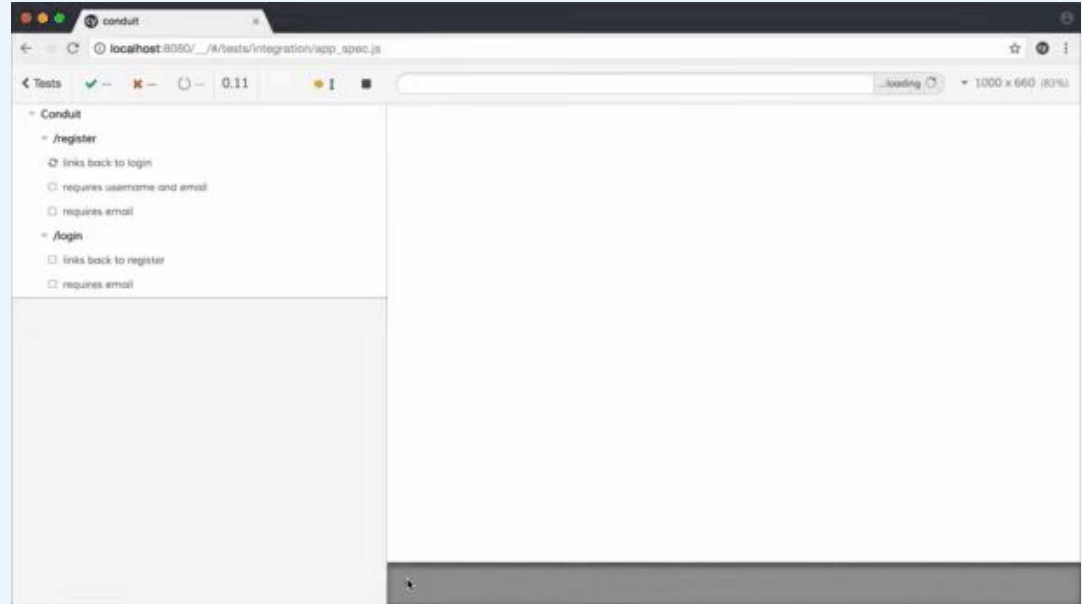
**Cypress**



# Cypress

Cypress è un tool per test front-end, che gira su browser.

Viene eseguito nello stesso run loop della tua applicazione.



[Credits [Cypress.io](https://cypress.io) website]

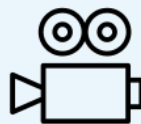
## Feature principali

JS/  
TS

Test in JavaScript  
o TypeScript



Supporto a  
Mocha reporters



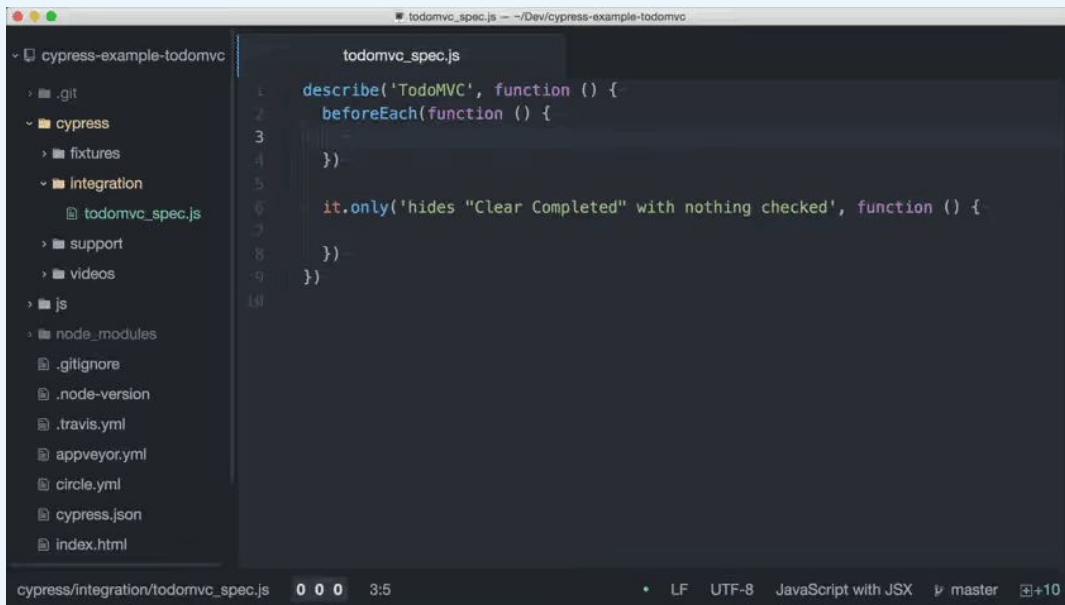
Screenshot e  
Video



Cross browser  
Testing



# Cypress



```
1 describe('TodoMVC', function () {
2   beforeEach(function () {
3     //
4   })
5
6   it.only('hides "Clear Completed" with nothing checked', function () {
7     //
8   })
9 })
```

Permette di scrivere test in JS o TS.

Niente driver binding, nessun nuovo linguaggio esoterico.

[Credits [Cypress.io](https://www.cypress.io) website]

# Cypress - Mocha reporters

[esempio di un estratto del file cypress.json]

```
{
  [...]
  "reporter": "mochawesome",
  "reporterOptions": {
    "reportDir": "output/reports",
    "overwrite": false,
    "html": false,
    "json": true
  },
  [...]
}
```

Cypress permette di configurare Mocha [reporters](#) per customizzare il formato dei report dei test.

Con questi parametri nel cypress.json, ci ritroviamo un json per test file

Cypress

# Features che non usiamo [ancora]



## Features di Cypress che ancora non usiamo



Spies, Stubs, e  
Clocks



Controllo del  
network



Cypress  
[Dashboard](#)



Cypress  
[plugins](#)

# Comparazione con altre alternative



# Selenium

Selenium



# Comparazione con Selenium

## Selenium

---

- Viene eseguito fuori dal browser
- Servono anche un wrapper per Selenium, un framework, una libreria di assertions. Minimo
- I test tendono ad essere instabili
- Se usi Docker, devi avere 2 containers: Selenium e browser

## Cypress

---

- Cypress gira nel browser stesso
- Include Mocha come test framework e Chai per le assertions, con mocking e stubbing
- È in grado di capire meglio lo stato della tua app
- Se usi Docker, devi solo lanciare Cypress in un container

# Comparazione con Selenium (e Nightwatch)

## Selenium con Nightwatch

```
function login () {
  return navigate()
    .setValue("@usernameInput", username)
    .setValue("@passwordInput", password)
    .click("@submitButton")
    .waitForElementVisible(
      ".my-private-app-class",
      (time = 25000),
      (abortOnFailure = false),
      (message = "Redirect to private area")
    )
    .getTitle(function (title) {
      if (title !== "Private area") {
        console.log("Login failed");
        exit();
      }
    });
}
```

## Cypress

```
export const login = (host, username, password) => {
  cy.visit(host);
  cy.get("input[name='username']").type(username);
  cy.get("input[name='password']").type(password);
  cy.get("form").submit();
  cy.url().should("include", "/private-area");
};
```



# Puppeteer



# Comparazione con Puppeteer

## Puppeteer

---

- Gira solo con Chrome
- Servono anche un test framework (Jest?), ecc...
- Niente registrazione video

## Cypress

---

- Supporta Chrome e Firefox
- Include test framework e altro
- Video mp4 di ciascun test

**Distribuirlo ai  
team**



## Distribuirlo ai team - Feature principali



App e test nella  
stessa  
repository



Immagine  
Docker  
ready-to-go



Upload degli  
artifacts su S3



BuildKite &  
Jenkins2



Integrazione  
con Slack e  
PagerDuty



DataDog  
metrics

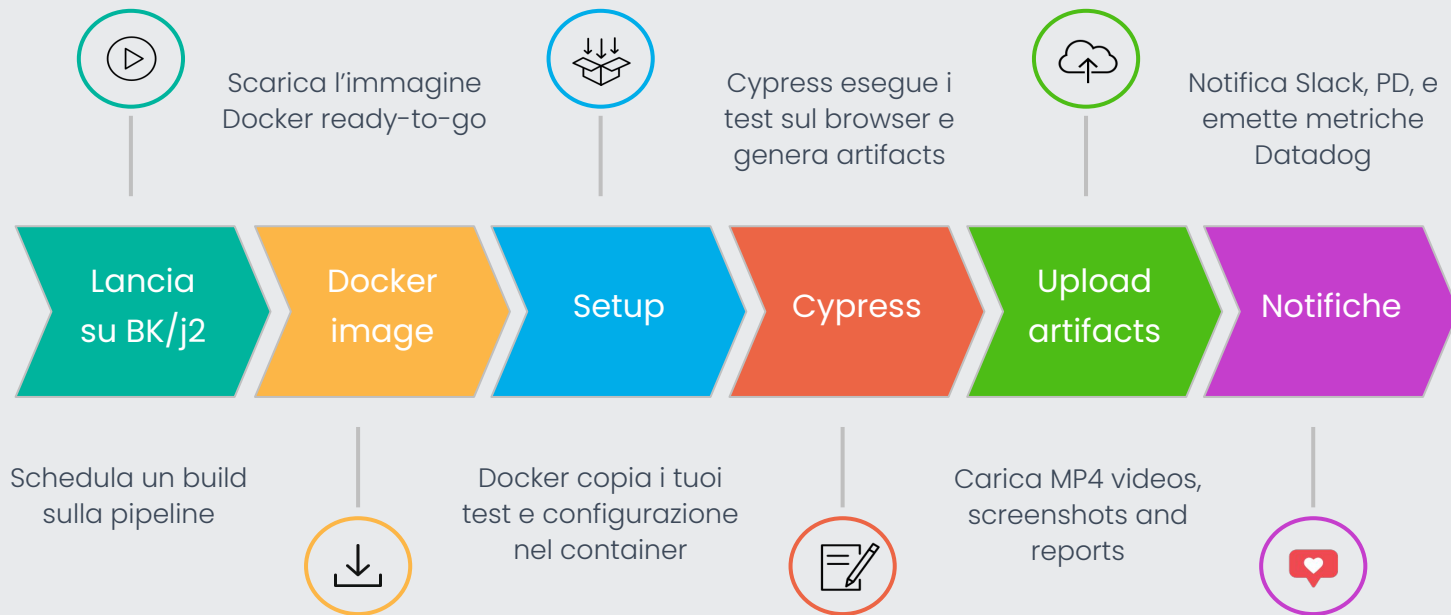


Supporto a  
environment  
multipli

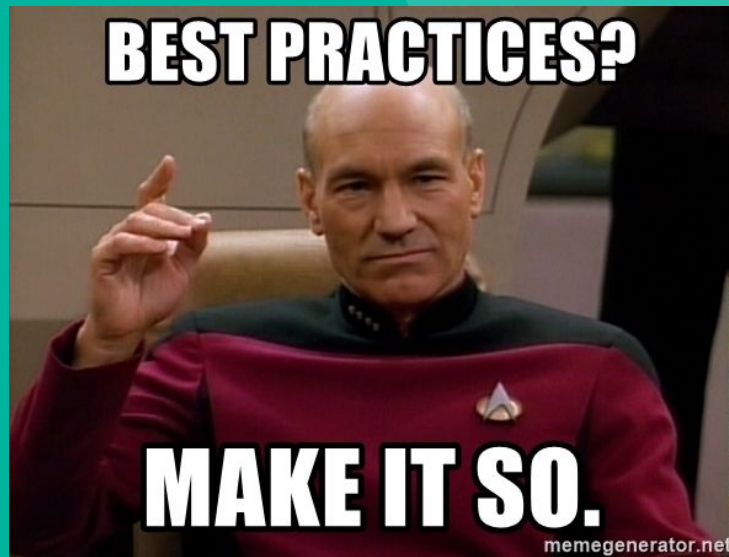


Possibilità di  
eseguire  
specifici test

## Distribuirlo ai team - How it works



# [Le nostre] Best practices



## I Classificazione dei test - Best practices

P0

UI non  
carica



Sezione  
business critical  
non funzionante

Impossibile  
completare  
un acquisto

P1

Sezione  
secondaria  
non  
funzionante

Azione  
minore non  
funzionante

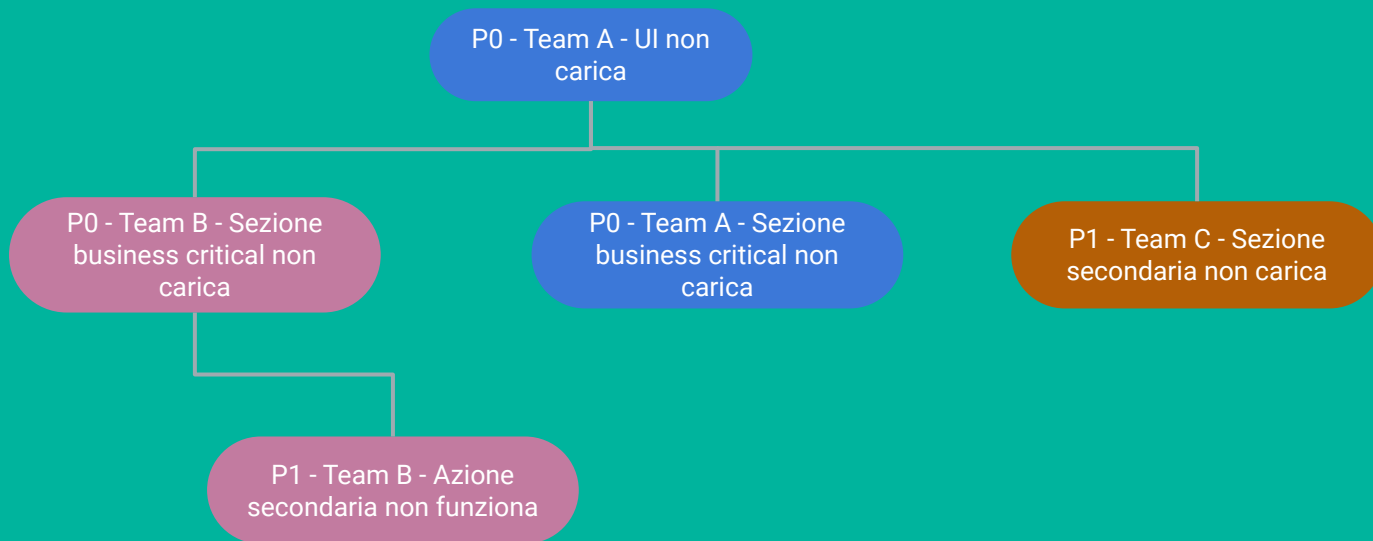
## I **Multi-team e ownership** – Best practices

Quando un'app è gestita da vari team:

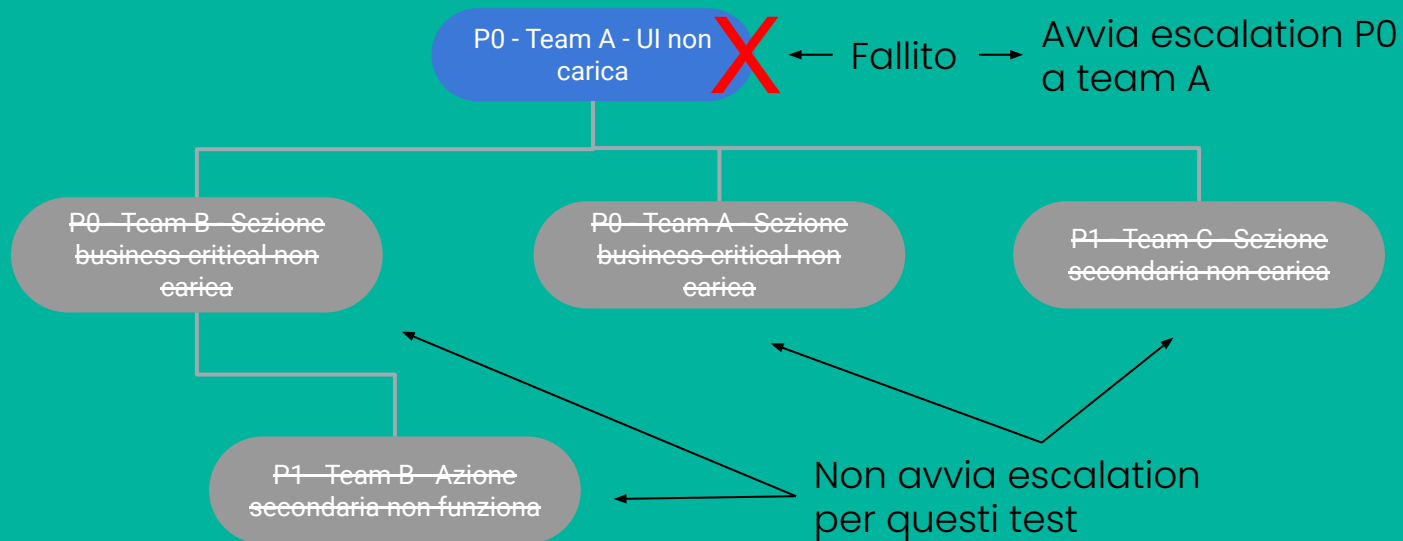
- Dividi i test per ownership
- Strutturare i tuoi test in maniera gerarchica permette di limitare notifiche per problemi fuori dal proprio dominio
- Permetti configurazioni multi-team, per notificare su canali distinti



## I Multi-team e notifiche - Best practices



## I Multi-team e notifiche - Best practices



## I Monitoraggio di production – Best practices

- Esecuzione indipendente, schedato per girare su CI
- Cypress è veloce ma non dimentichiamo i tempi della CI per avviare il job e i tempi di Docker
- TL;DR: lancia soltanto i test più critici

## I Testing post-deploy – Best practices

- Configura la tua pipeline di deploy (Staging e Production) per eseguire test post-deploy.
- Testare Staging prima di rilasciare su Production?  
Prevedi di poter escludere facilmente gli smoke test:
  - Outage di Staging
  - Hotfix a Production
- Lancia tutti i test!

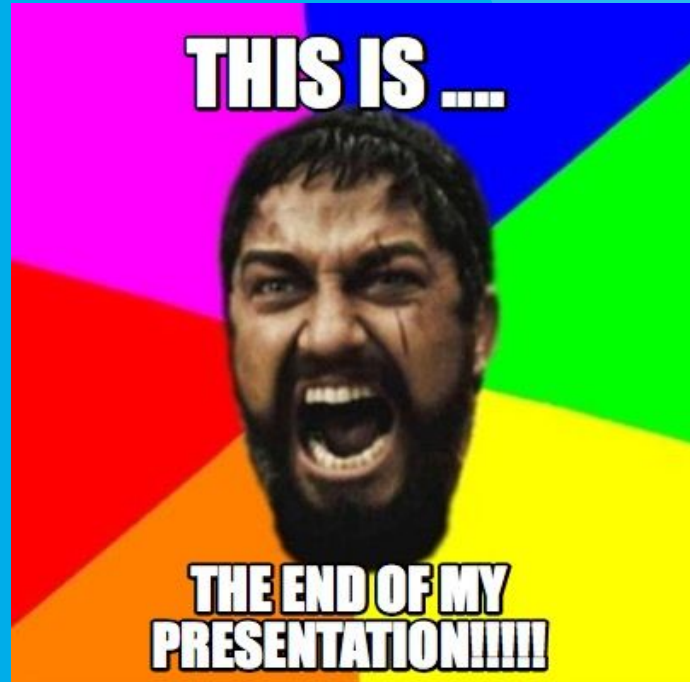


## I Smoke testing – Best practices



Questo tipo di smoke testing **NON** deve essere visto come un'alternativa a unit testing, functional/end-to-end testing, monitoring delle API HTTP usate dalla UI, ...

Però potrete smettere di ricaricare con ansia 3/4 [mila] sezioni random dopo ogni deploy su Production.



Grazie di aver partecipato!

# Lorenzo Savini

savo@develer.com

Vuoi rimanere aggiornato sugli eventi Develer?  
Seguici nei nostri canali social:



**develer**

[www.develer.com](http://www.develer.com)