

# Testing

---



# Angular testing

---

- Tools and environment
- @angular/.../testing
- Testing strategies

# Testing - Tools and environment

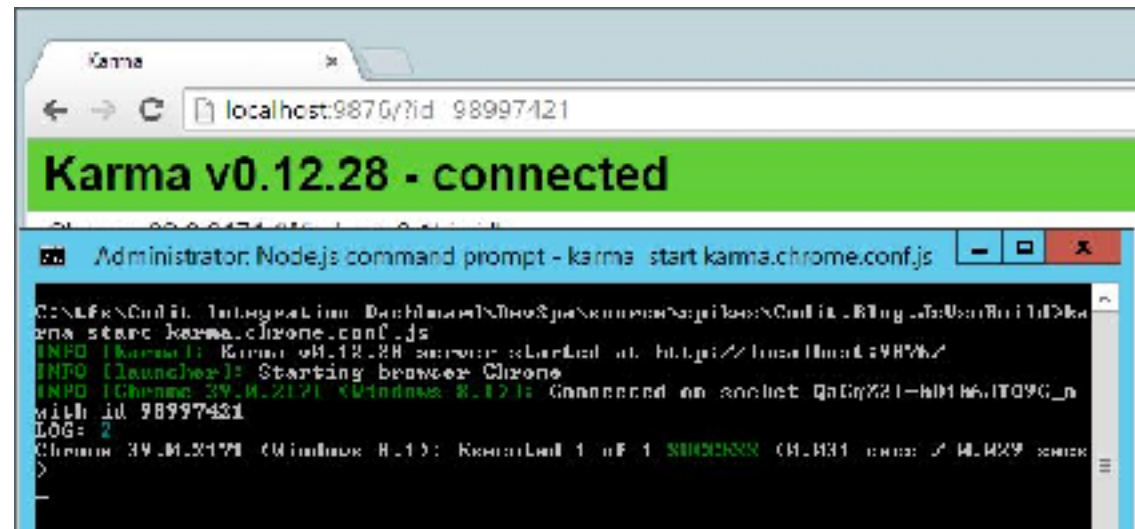
---



# Testing - Karma

---

- Unit test runner
- Loads files
- Starts browser
- Reporters



# Testing - Protractor

---

- e2e test runner
- API for interaction with browser
- Jasmine
- page objects



# Testing - Jasmine

---

- Testing SDK
- Test structure
- Assertions
- Matchers

```
Jasmine 1.2.0 revision 1337005947 finished in 0.075s
* * * * *

Passing 13 specs

username validation
  should return true for valid usernames (with "valid values" using abc)
  should return true for valid usernames (with "valid values" using longusername)
  should return true for valid usernames (with "valid values" using john_doe)
  should return false for invalid usernames (with "invalid values" using ab)
  should return false for invalid usernames (with "invalid values" using name_too_long)
  should return false for invalid usernames (with "invalid values" using no spaces)
  should return false for invalid usernames (with "invalid values" using invalid)

camel case converter
  should return camel case string (with "dashes" using foo, Foo)
  should return camel case string (with "dashes" using foo-bar, FooBar)
  should return camel case string (with "dashes" using foo-bar-baz, FooBarBaz)
  should return dashed string (with "camel case" using Foo, foo)
  should return dashed string (with "camel case" using FooBar, foo-bar)
  should return dashed string (with "camel case" using FooBarBaz, foo-bar-baz)
```

# Testing - Jasmine

---

## Test structure

```
describe('The tested feature', () => {  
  beforeEach(() => { // setup code });  
  
  describe('could be nested', () => {  
    it('is a test', () => { ... });  
  });  
});
```

# Testing - Jasmine

---

## Assertion and matchers

```
expect(array.length).toBe(2);           // ===  
expect(array).toEqual([]);              // ==  
expect('some text').toMatch(/[A-Z]{4}/); // regex  
expect('some text').toContain('some');  // indexOf
```



# @angular/.../testing

---



# @angular/.../testing

---

- TestBed
- Helper modules
  - RouterTestingModule
- inject() / TestBed.get()
- async() / fakeAsync()

# Testing - TestBed

---

- Configuration class for tests
- A new instance for each test
- Handle dependencies and mocks
- Create Components
- Change detection control

# Testing - Services

---

- Mocking
- inject() / TestBed.get

```
// mocking van services
{ provide: BookService, useClass: MyBookMockService }

it('Instance has been injected',
  inject([ BookService ], (service: BookService) => {
    expect(service).toBeDefined();
  }));
```

# Testing - Services

---

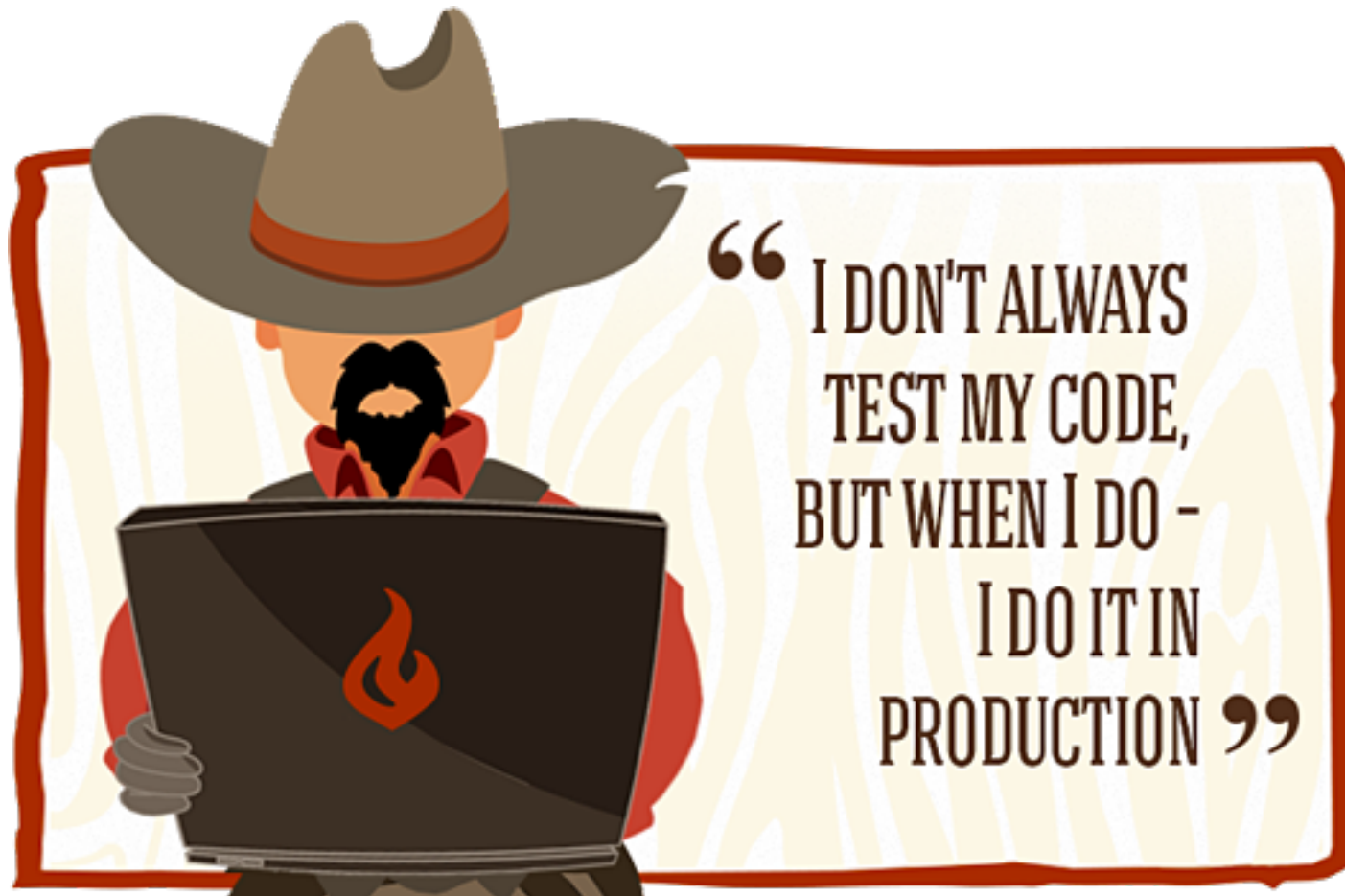
- Mocking
- inject() / TestBed.get

```
// mocking van services
{ provide: BookService, useClass: MyBookMockService }

it('Instance has been injected',
  inject([ BookService ], () => {
    const bookService = TestBed.get(BookService);
    expect(bookService).toBeDefined();
  }));
```

# Test strategies

---



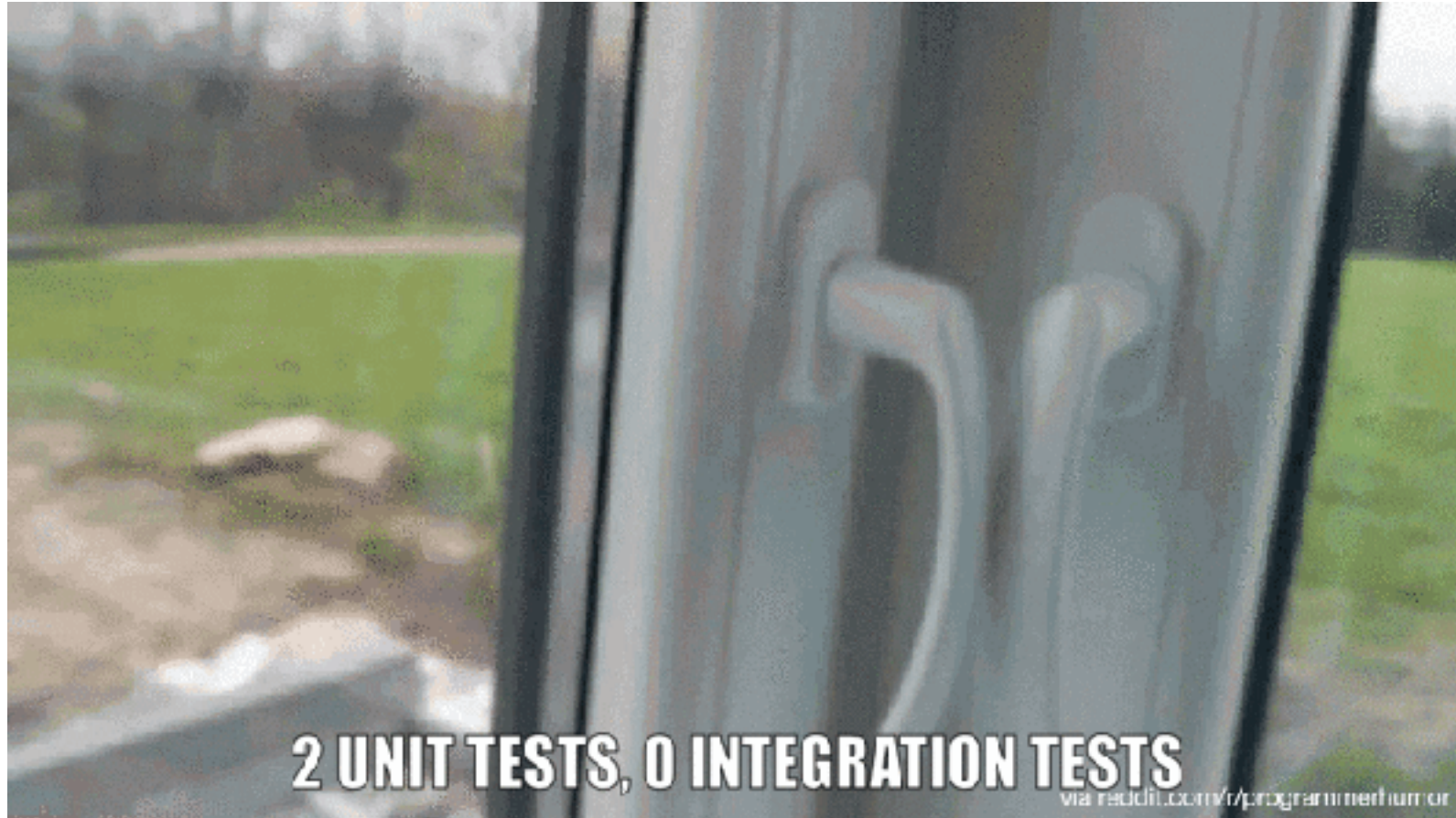
# Test strategies

---

- Dumb Components
- Smart Components
- Lifecycle events
- Services

# Test strategies

---





# Look back - Testing

---

- Karma + Jasmine + Protractor
- @angular/.../testing
- TestBed
- inject()
- Test strategies

# Testing - Demo

---

