



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Testing

4 - Component & Integration Tests Advanced

Reducing Boilerplate



Test Setup



Approaches

1. beforeEach

- a. Default Configuration by Angular CLI
- b. All tests with same setup
- c. Simple situations

2. Nested describes, aka. Contexts

- a. Advanced Scenarios
- b. Limited amount of TestBed configuration

3. Factory methods

- a. The test has full control - not the Test Suite
- b. Most Flexible



Popular Libraries



Testing Library



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Spectator vs. Testing Library

Spectator

- Great Support for Mocking
- Applicable for various testing types
- Much more common in Angular

Testing Library

- Enforces a Testing Philosophy
- Superior UI Debugging Tools
- find* removes Asynchronicity and Change Detection
- userEvent with real-world behaviour
- Common API for multiple frameworks (Cypress,...)
- Host Components very easy



Harnesses

Taming the Beast...



Test Harnesses

- Page Object Models for Component Tests
- Available since Angular v9
- Provide a test abstraction for components
- Developed by @angular/material
- Full coverage for material since v11
- Reduces code size significantly
 - Better Readability
 - Better Maintainability



Address Validation

Address

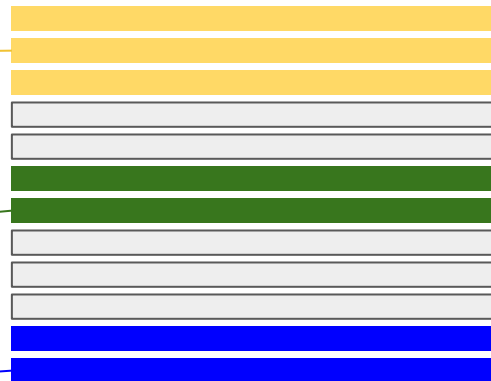
Domgasse 5



Please enter your address

Submit

Address found



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Address Validation

Address

Domgasse 5

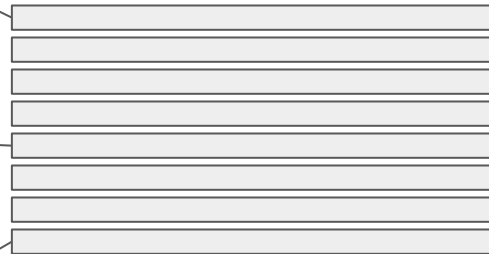
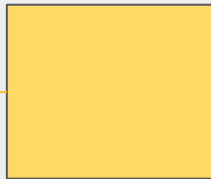


Please enter your address

Submit

Address found

Harness



- Element Selection
- Change Detection
- Asynchronicity
- Rendering



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Creating a Harness

```
export class RequestInfoComponentHarness extends ComponentHarness {  
    static hostSelector = "app-request-info";  
  
    protected getButton = this.locatorFor("button[type=submit]");  
  
    async submit(): Promise<void> {  
        const button = await this.getButton();  
        return button.click();  
    }  
}
```



Using a Harness

```
it("should use the harness", async () => {  
  // setup TestModule...  
  
  const harness = await TestBedHarnessEnvironment.harnessForFixture(  
    fixture,  
    RequestInfoComponentHarness  
  );  
  await harness.submit();  
  // expect something  
});
```



Lab Time



Towards full Integration Tests

