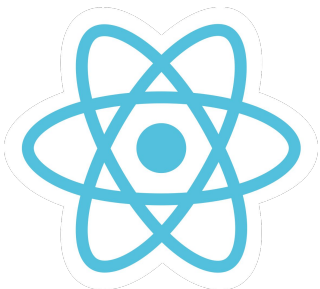


Effektives Testen von Browser-Anwendungen

Werkzeuge und Vorgehen

Jax 2019 - Mainz



Fokus

Überblick zur Technik

Nicht in diesem Talk:
komplementäre Praktiken
Code-Review / Pull-Requests / Pairing

Karsten Sitterberg

- Entwickler, Trainer, Berater
 - sitterberg.com
 - @kakulty
- Java User Group Münster
- Frontend Freunde Münster



Qualitätsziele

- Keine Fehler
- Gut verständliches Programm
- Neue Features leicht zu ergänzen
- Langfristige Wartbarkeit sichergestellt
- Hohe Entwicklerproduktivität

Qualitätsziele

- Keine Fehler
- Gut verständliches Programm
- Neue Features leicht zu ergänzen
- Langfristige Wartbarkeit sichergestellt
- Hohe Entwicklerproduktivität

Vorsicht

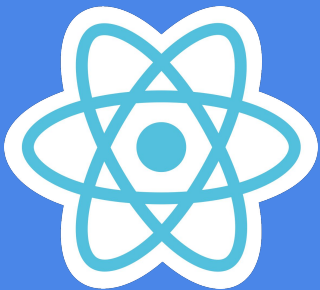
Nicht gratis



Test Driven Development

Tests

Funktioniert doch!



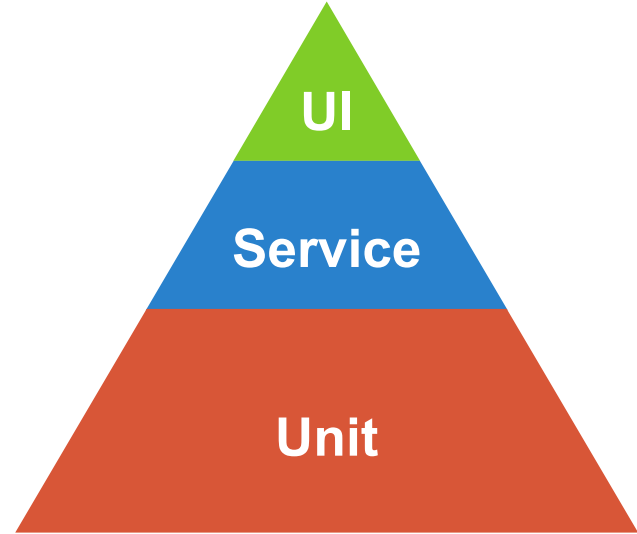
Funktionale Tests

Teil der Codebasis

Teil der **Dokumentation**

Erster Code Re-Use !

Schlüssel zu langfristiger Wartbarkeit



€€€



€

Coverage

Performance

Unit Tests

E2E Tests

Visual Diff

Jest

Cucumber

Testcafe

AyeSpy

Karma

Appium

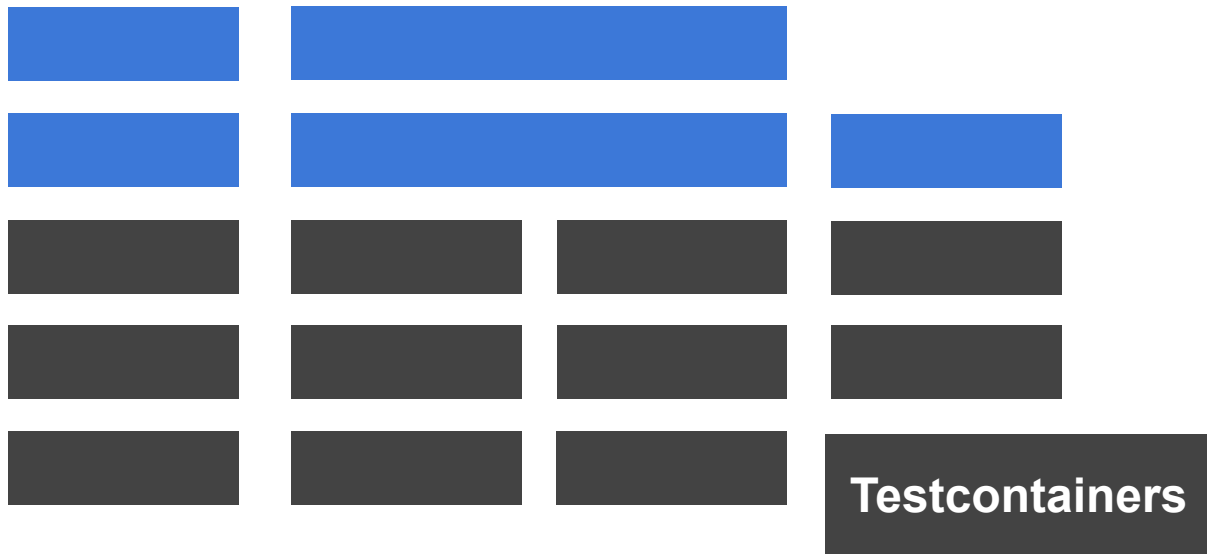
Cypress

backstop.js

Jasmine

Selenium

puppeteer



Wiremock

CI Server

Docker

Input

**Conditional
Rendering**

Demo Anwendung

Routing

Redux

Welcome to angular-quality!

Name

Kundennummer

Aktuelle Angebote

Commit: 7567cc2d3029a84ad2eb65141bab7913e20378db

Build: Wed Sep 19 09:54:37 UTC 2018

Welcome to angular-quality!



Gummibären

Gratis

Nur solange der Vorrat reicht

[Hier](#)



Äpfel

2,50€

In rot, gelb, grün und blau!

[Hier](#)



3 gelbe Bananen

4,70€

In dick, dünn, lang und kurz!

[Hier](#)

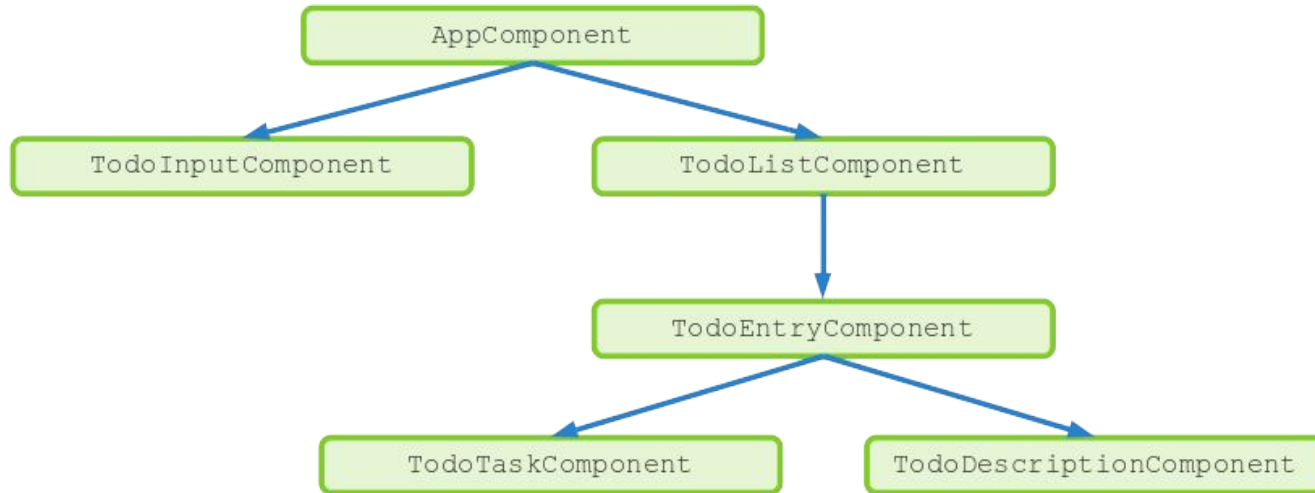
Premiumangebote



Testen von **Browser-** Anwendungen

Was muss getestet werden

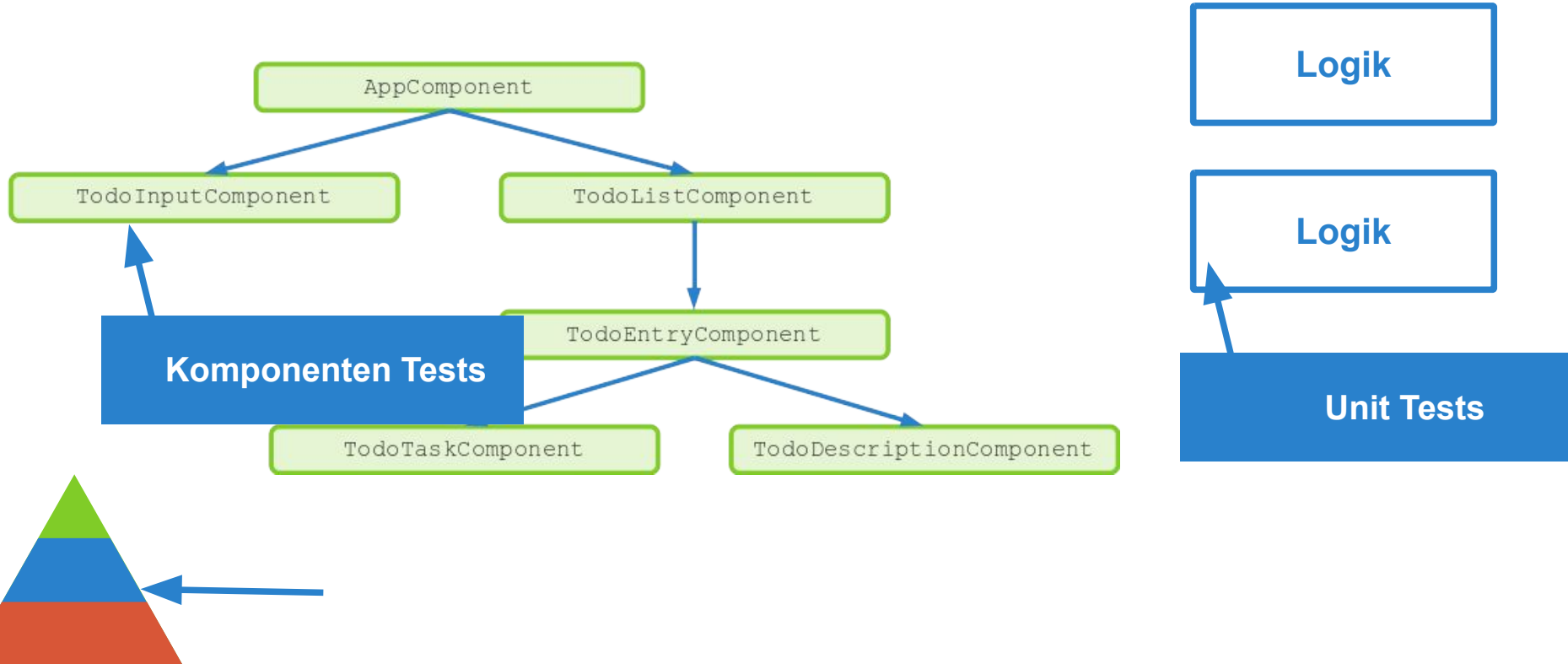
Struktureller Aufbau Browser Anwendung



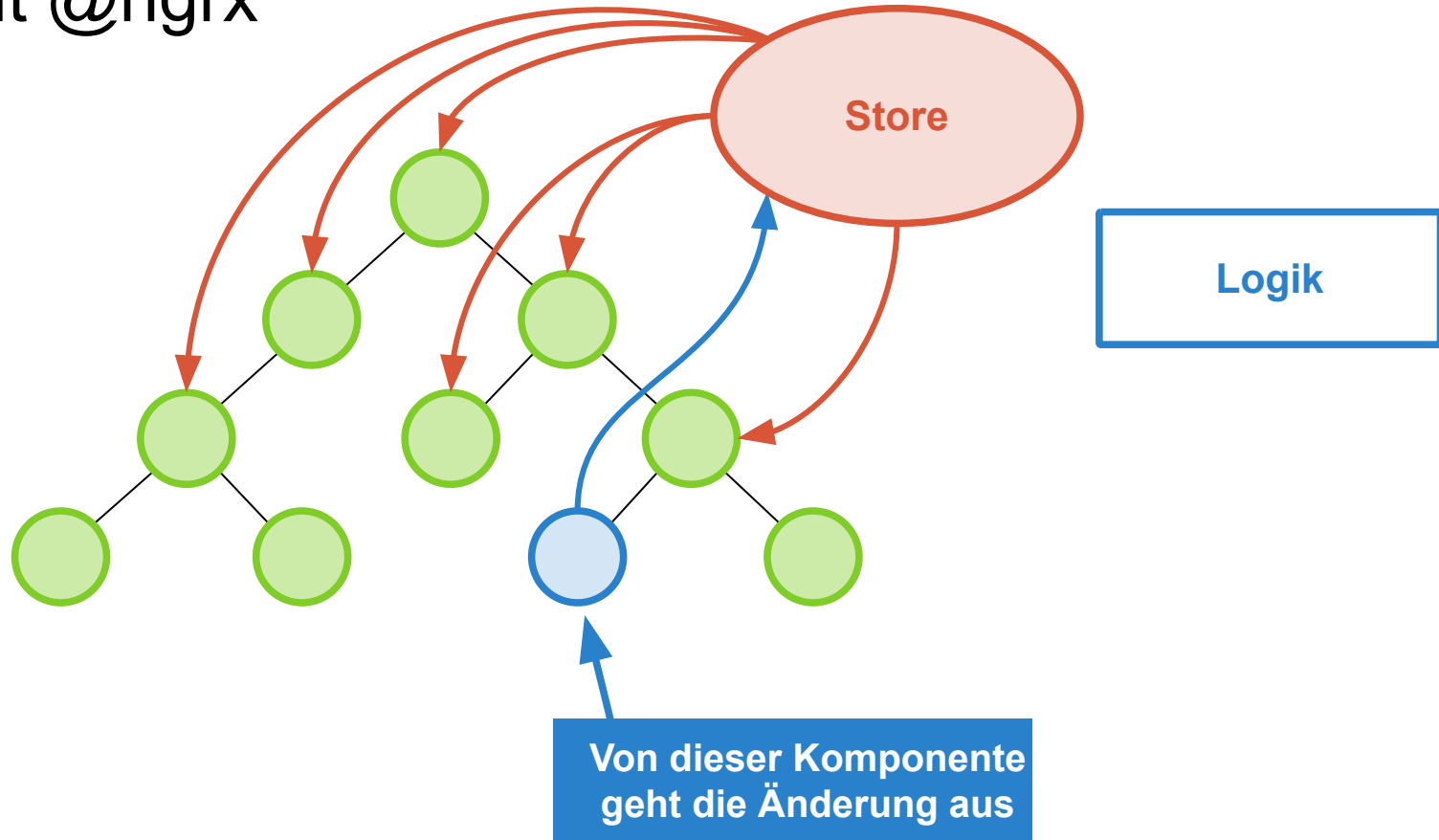
Logik

Logik

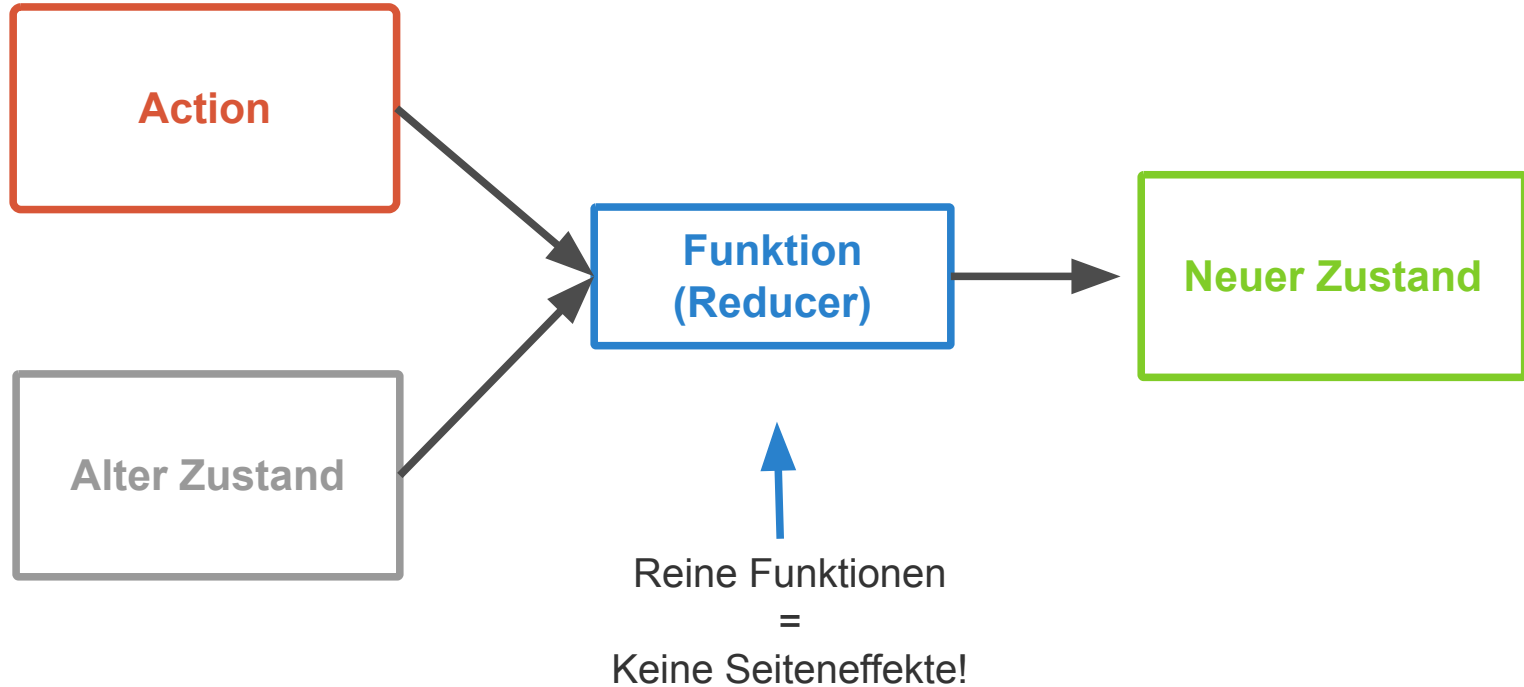
Struktureller Aufbau Browser Anwendung



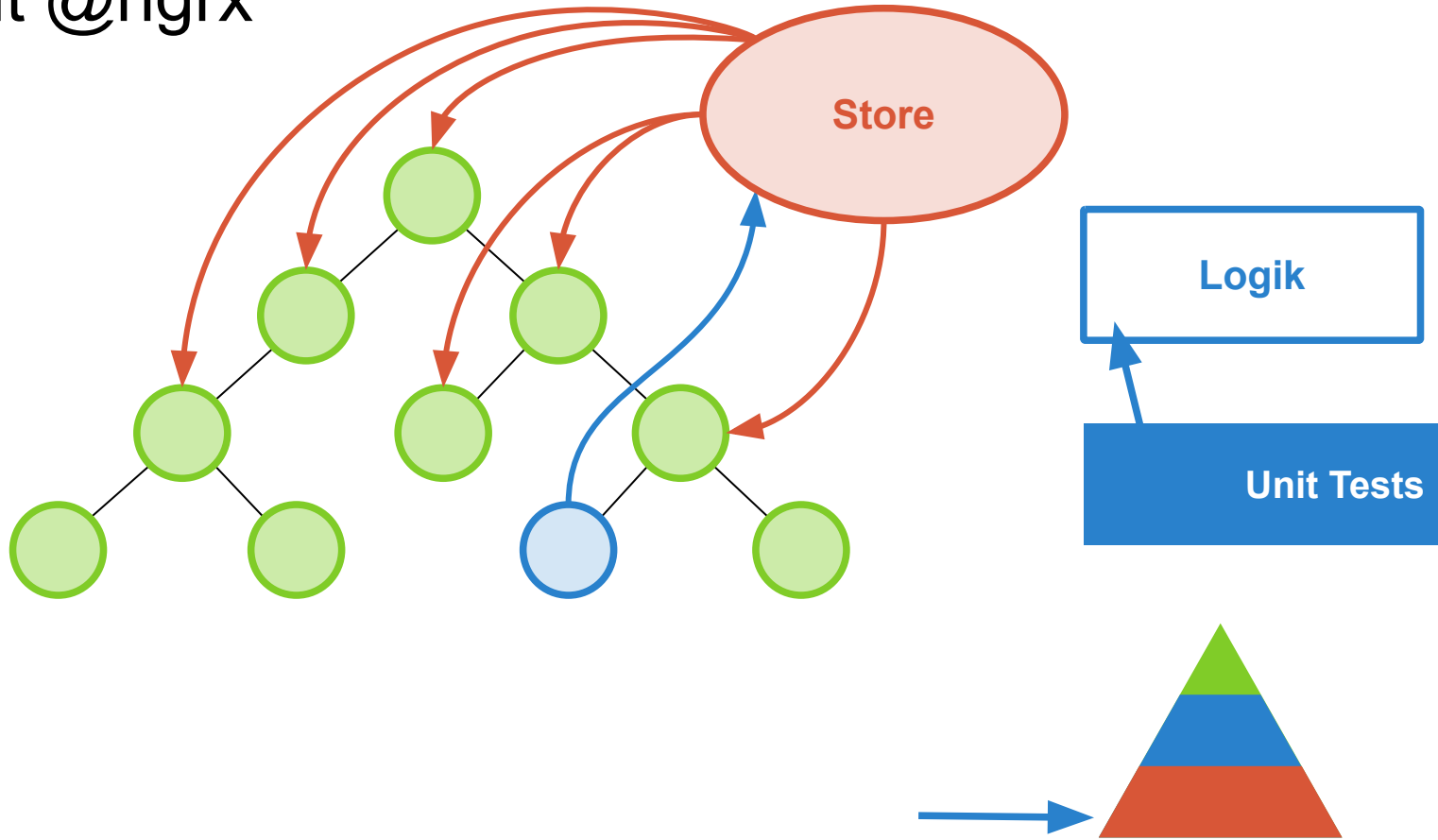
Redux mit @ngrx



Aktualisierung des Zustands

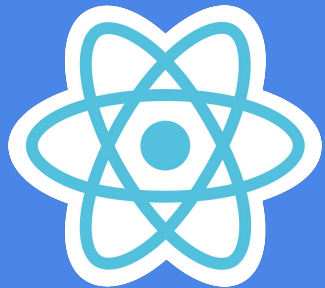


Redux mit @ngrx



Karma & Jasmine

Unit Tests



TypeScript





Test-Runner

Lässt Tests im Browser laufen

Unterschiedliche Browser möglich

Auch Devices

Vergleichbar: JUnit



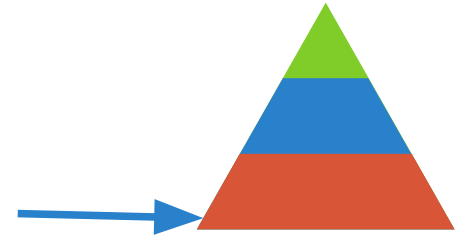
Test-Library

Stellt Test-Syntax zur Verfügung

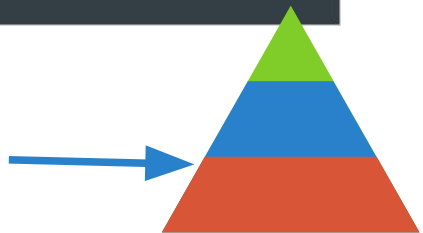
Stellt Assertions zur Verfügung

Vergleichbar: AssertJ, Hamcrest

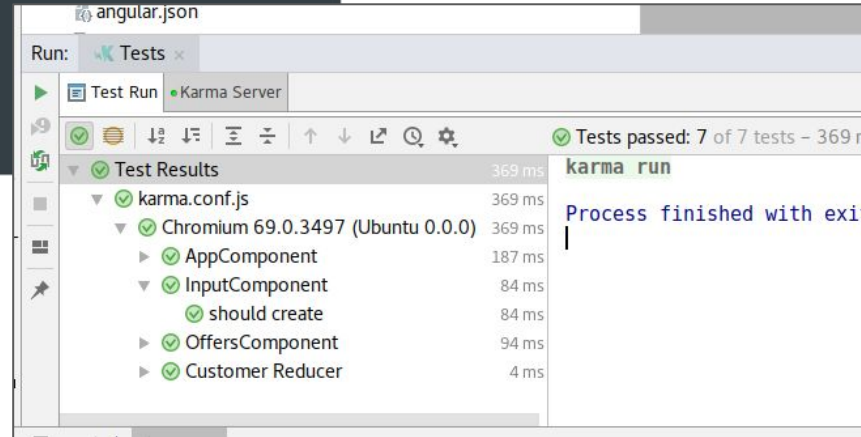
```
it(`should have `, async(() => {  
  Reiner javascript testfall ohne komponente  
}));
```



```
it(`should have as title 'angular-quality'`, async(() => {  
  const fixture = TestBed.createComponent(AppComponent);  
  const app = fixture.debugElement.componentInstance;  
  expect(app.title).toEqual('angular-quality');  
}));
```



```
it('should have as title 'angular-quality'', async(() => {
  const fixture = TestBed.createComponent(AppComponent);
  const app = fixture.debugElement.componentInstance;
  expect(app.title).toEqual('angular-quality');
}));
```



The screenshot shows a web browser window with the 'Run' tab open, displaying test results for 'angular.json'. The test runner is 'Karma Server'. The results show that all 7 tests passed successfully in 369 ms. The tests are categorized under 'karma.conf.js' and 'Chromium 69.0.3497 (Ubuntu 0.0.0)'. The individual tests and their durations are:

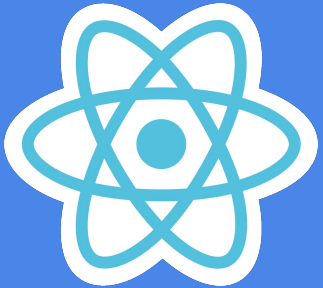
Test Name	Duration (ms)
Chromium 69.0.3497 (Ubuntu 0.0.0)	369
AppComponent	187
InputComponent	84
should create	84
OffersComponent	94
Customer Reducer	4

Summary: Tests passed: 7 of 7 tests - 369 ms. The console also shows the command 'karma run' and the message 'Process finished with exit code 0'.

```
+ angular-quality git:(master) ng test --watch false --progress false
19 09 2018 21:37:17.723:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
19 09 2018 21:37:17.727:INFO [launcher]: Launching browser Chrome with unlimited concurrency
19 09 2018 21:37:17.767:INFO [launcher]: Starting browser Chrome
19 09 2018 21:37:21.185:INFO [Chromium 69.0.3497 (Ubuntu 0.0.0)]: Connected on socket K4CYxak0M2Ha11GpAAAA with id 89934482
Chromium 69.0.3497 (Ubuntu 0.0.0): Executed 7 of 7 SUCCESS (0.323 secs / 0.304 secs)
+ angular-quality git:(master) █
```

Jest

Geteilt von Facebook



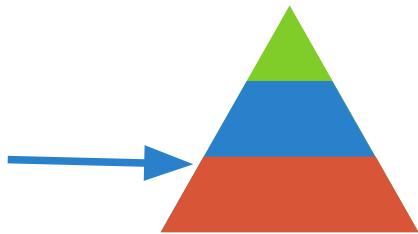
Jest

Für react entwickelt

Kein Browser nötig

Nutzbar mit Angular

Snapshot-Tests



```
> jest -c jest.config.js
```

```
PASS src/app/input/input.component.spec-jest.ts  
  InputComponent  
    ✓ should create (141ms)
```

```
Test Suites: 1 passed, 1 total
```

```
Tests:       1 passed, 1 total
```

```
Snapshots:  1 passed, 1 total
```

```
Time:        1.401s, estimated 2s
```

```
Ran all test suites.
```

```
→ angular-quality git:(master) x █
```

```
it('should create', () => {  
  expect(component).toBeTruthy();  
  expect(fixture.debugElement.nativeElement).toMatchSnapshot();  
});
```

Jest Snapshots

- Werden beim ersten Testlauf erzeugt
- Sollten mit eingechecked werden
- Update mit `–u``

```
input.component.spec-jest.ts.snap ×
1 // Jest Snapshot v1, https://goo.gl/fbAQLP
2
3 exports[`InputComponent should create 1`] = `
4 <div
5   id="root0"
6   ng-version="6.1.7"
7 >
8   <section>
9     <form
10       class="ng-untouched ng-pristine ng-invalid"
11       ng-reflect-form="[object Object]"
12       novalidate=""
13     >
14       <div
15         class="form-group"
16       >
17         <label
18           for="customerName"
19         >
20           Name
21         </label>
22         <input
23           class="form-control ng-untouched ng-pristine ng-invalid"
24           formcontrolname="name"
25           id="customerName"

```

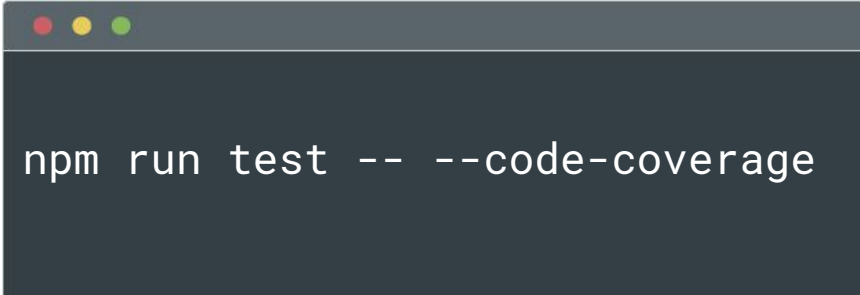
```
it('should create', () => {
  expect(component).toBeTruthy();

  expect(fixture.debugElement.nativeElement).toMatchSnapshot();
});
```

Code Coverage

Code Coverage

Mit Karma und Jest



```
npm run test -- --code-coverage
```



```
npm run test:jest -- --coverage
```

Code Coverage

Mit Jest und Karma

```
→ angular-quality git:(master) ng test --watch false --code-coverage
10% building modules 4/4 modules 0 active19 09 2018 21:24:38.885:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
19 09 2018 21:24:38.889:INFO [launcher]: Launching browser Chrome with unlimited concurrency
19 09 2018 21:24:38.993:INFO [launcher]: Starting browser Chrome
0.3497 (Ubuntu 0.0.0)]: Connected on socket yEs63v7qWvl-zANwAAAA with id 35954550
Chromium 69.0.3497 (Ubuntu 0.0.0): Executed 7 of 7 SUCCESS (0.357 secs / 0.333 secs)
```

```
==== Coverage summary =====
Statements : 90.18% ( 101/112 )
Branches   : 58% ( 58/100 )
Functions  : 68.42% ( 13/19 )
Lines      : 94.74% ( 90/95 )
=====
```

```
→ angular-quality git:(master)
```

```
> jest -c jest.config.js --coverage
```

```
PASS src/app/input/input.component.spec-jest.ts
  InputComponent
    ✓ should create (219ms)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	73.81	45	30	77.78	
app/input	83.33	75	50	80	
input.component.html	100	100	100	100	
input.component.ts	82.35	75	50	78.57	28,32,33
app/store	65.22	25	16.67	75	
customer.actions.ts	60	100	50	60	9,11
customer.reducer.ts	50	0	0	57.14	15,18,22
index.ts	80	20	0	100	14,17
environments	100	100	100	100	
environment.ts	100	100	100	100	

```
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 passed, 1 total
Time:        2.081s
Ran all test suites.
```

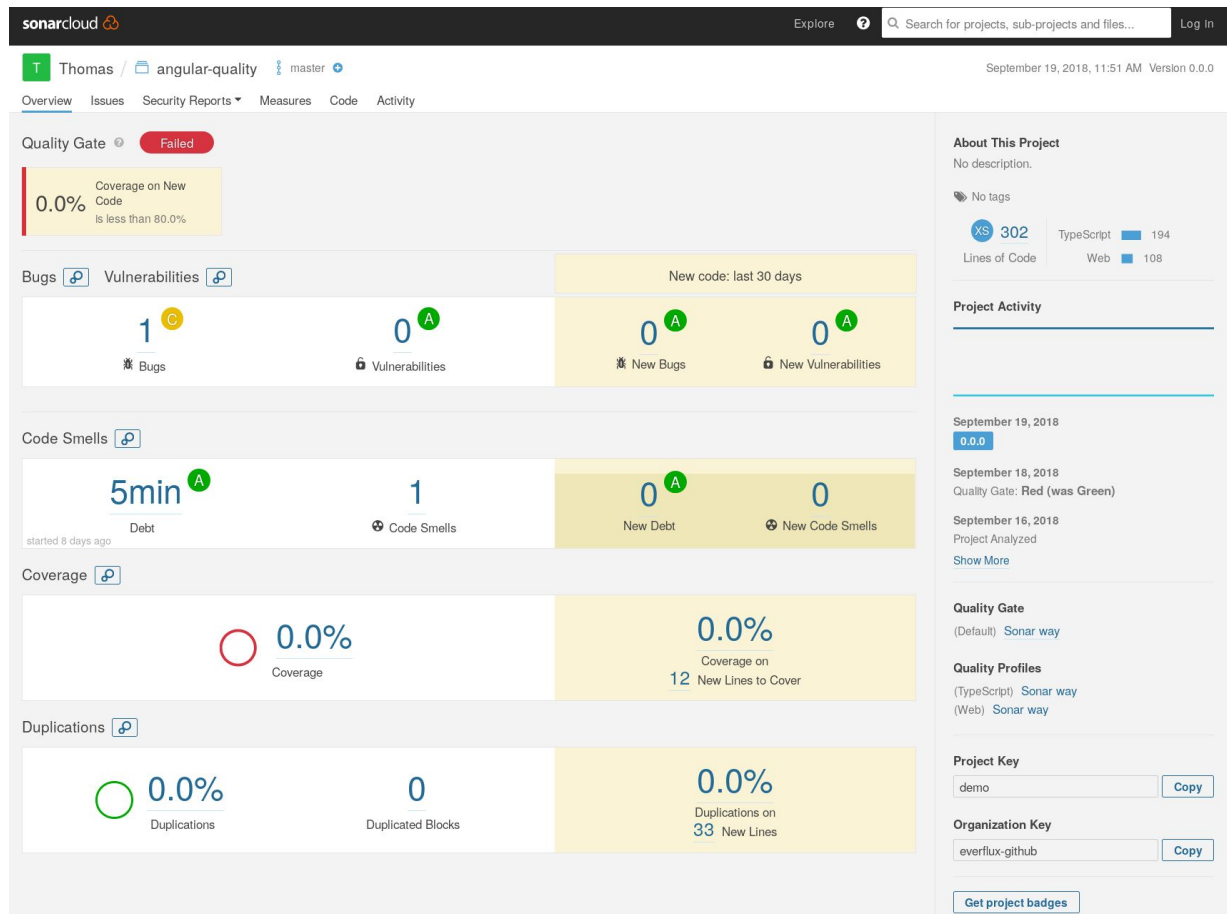
```
→ angular-quality git:(master)
```

SonarQube

Metrikdaten zur Qualität
im Projektverlauf

Dashboard für Projektleiter

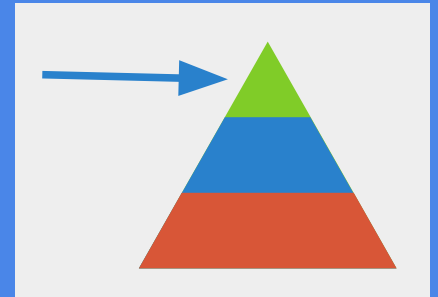
Qualitygate





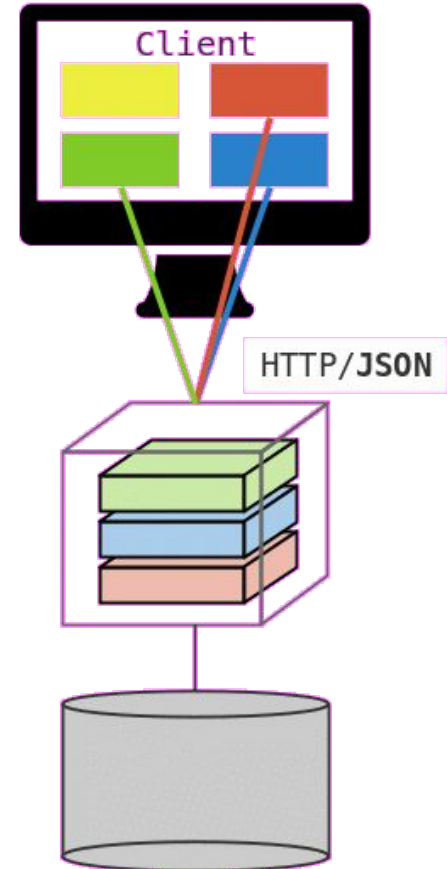
Testen der ganzen Anwendung

Ende zu Ende Tests (e2e)



End-2-End Tests

- Oberfläche mit Integration von Backend
- Typischerweise mit Fake/Mock Backend
- Tipp: Eher wenige Tests



Fake Backends

Wiremock

Wiremock

```
npm i -D wiremock-standalone
```

```
wiremock --root-dir ./mock
```

```
{  
  "request": {  
    "method": "POST",  
    "url": "/api/order"  
  },  
  "response": {  
    "status": 200,  
    "headers": {  
      "Access-Control-Allow-Origin": "http://localhost:4200",  
      "Access-Control-Allow-Methods": "POST"  
    },  
    "jsonBody": {  
      "items": ["3 gelbe Bananen"]  
    }  
  }  
}
```

Protractor

Und Selenium



Protractor

```
import { browser, by, element } from 'protractor';
import { AppPage } from './app.po';

describe('workspace-project App', () => {
  let page: AppPage;

  beforeEach(() => {
    page = new AppPage();
  });

  it('should show Premium-Angebote if Premium-Customer', () => {
    page.navigateTo();
    page.fillNameInput('Helene Birne');
    page.fillCustomerNumberInput(300000);
    page.getSubmitButton().click();
    expect(page.getPremimOffers().getText()).toContain('Premiumangebote');
  });
});
```

Protractor E2E Tests

```
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.83 MB [initial] [re
[22:04:11] I/config_source - curl -o/home/karsten/job/projects/fronten
age.googleapis.com/
i 「wdm」: Compiled successfully.
[22:04:12] I/update - chromedriver: file exists /home/karsten/job/proj
[22:04:12] I/update - chromedriver: unzipping chromedriver_2.42.zip
[22:04:12] I/update - chromedriver: setting permissions to 0755 for /h
2.42
[22:04:12] I/update - chromedriver: chromedriver_2.42 up to date
[22:04:12] I/launcher - Running 1 instances of WebDriver
[22:04:12] I/direct - Using ChromeDriver directly...
Jasmine started

workspace-project App
  ✓ should display welcome message, submit button disabled
  ✓ should enable welcome button, if data is entered
  ✓ should show alert if customerNumber is bigger 10000
  ✓ should show Premium-Angebote if Premium-Customer
  ✓ should show only standard offers if normal customer

Executed 5 of 5 specs SUCCESS in 4 secs.
[22:04:17] I/launcher - 0 instance(s) of WebDriver still running
[22:04:17] I/launcher - chrome #01 passed
+ angular-quality git:(master) █
```

```
Run: E2E Tests x
Tests passed: 5 of 5 tests - 3 s 608 ms
protractor.conf.js (chrome 69.0.3497.81) 3 s 608 ms
  workspace-project App 3 s 608 ms
    ✓ should display welcome message, submit button disabled
    ✓ should enable welcome button, if data is entered 681 ms
    ✓ should show alert if customerNumber is bigger 10000 418 ms
    ✓ should show Premium-Angebote if Premium-Customer 753 ms
    ✓ should show only standard offers if normal customer 728 ms

protractor protractor.conf.js
[22:07:41] I/launcher - Running 1 instances of WebDriver
[22:07:41] I/direct - Using ChromeDriver directly...
Jasmine started

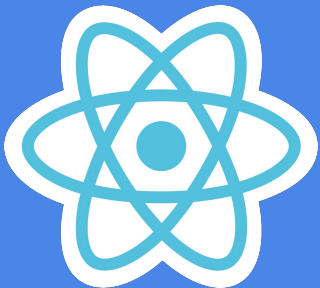
workspace-project App
  ✓ should display welcome message, submit button disabled
  ✓ should enable welcome button, if data is entered
  ✓ should show alert if customerNumber is bigger 10000
  ✓ should show Premium-Angebote if Premium-Customer
  ✓ should show only standard offers if normal customer

Executed 5 of 5 specs SUCCESS in 4 secs.
[22:07:46] I/launcher - 0 instance(s) of WebDriver still running
[22:07:46] I/launcher - chrome #01 passed

Process finished with exit code 0
```

Cypress

Mit node.js





Cypress

Test-Runner + Assertion-Library, vor allem E2E

Basierend auf Electron, Chai, Mocha, Sinon

Schnelle Test-Ausführung

Videos, Screenshots (Perceptual Diff)

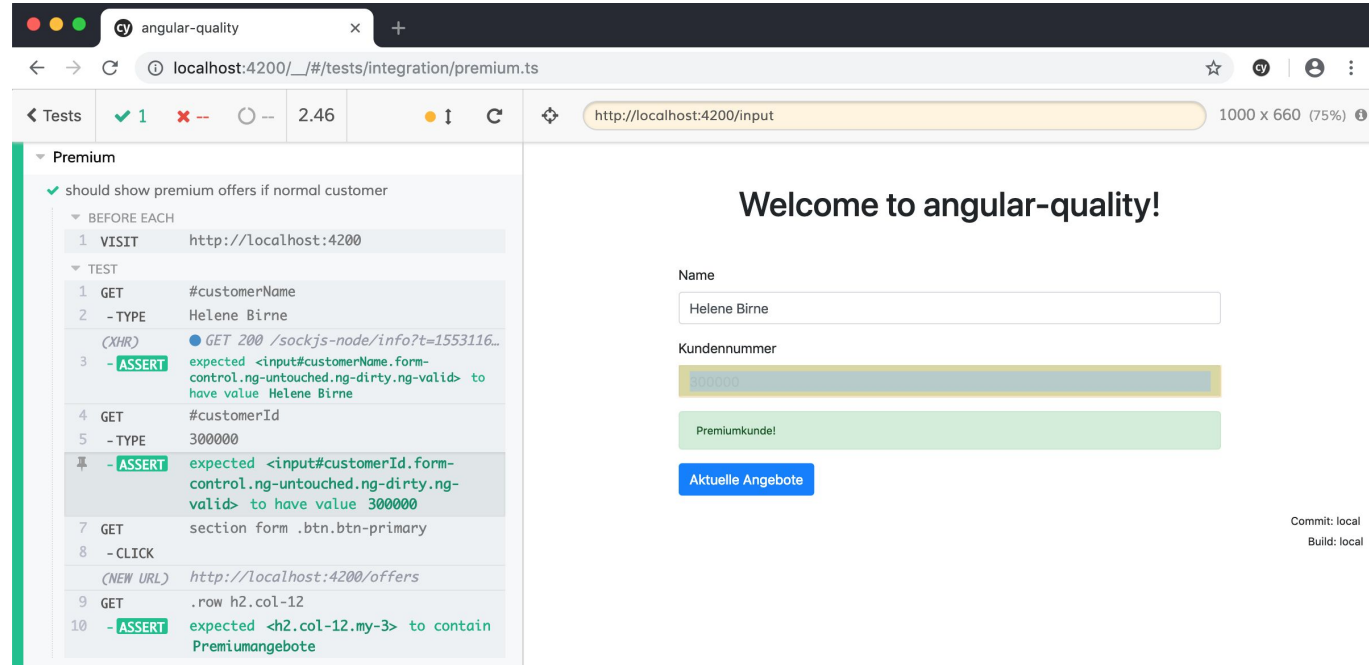
Debugging, HTTP-Mocking

Allerdings: Eingeschränkter Browser-Support

Cypress

Eingebaute "IDE"

Time-Travel Funktion



The screenshot displays the Cypress test runner interface. On the left, the test runner shows a test suite named "Premium" with a test "should show premium offers if normal customer". The test is currently failing. The command log shows the following steps:

- 1 VISIT http://localhost:4200
- 2 -TYPE Helene Birne
- 3 -ASSERT expected <input#customerName.form-control.ng-untouched.ng-dirty.ng-valid> to have value Helene Birne
- 4 GET #customerId
- 5 -TYPE 300000
- 6 -ASSERT expected <input#customerId.form-control.ng-untouched.ng-dirty.ng-valid> to have value 300000
- 7 GET section form .btn.btn-primary
- 8 -CLICK
- 9 GET .row h2.col-12
- 10 -ASSERT expected <h2.col-12.my-3> to contain Premiumangebote

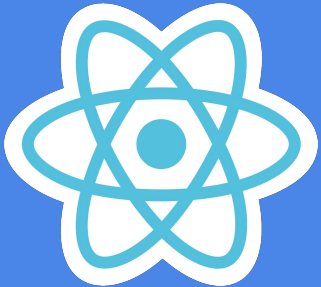
On the right, the web application is shown. The page title is "Welcome to angular-quality!". There is a form with a "Name" field containing "Helene Birne" and a "Kundnummer" field containing "300000". Below the form, there is a green message "Premiumkunde!" and a blue button labeled "Aktuelle Angebote".

Commit: local
Build: local

DOM Snapshot

TestCafe

Mit node.js



TestCafe



Test-Runner + Assertion-Library, vor allem E2E

Schnelle (parallele) Test-Ausführung

Extrem leichtes Setup, guter Browser-Support

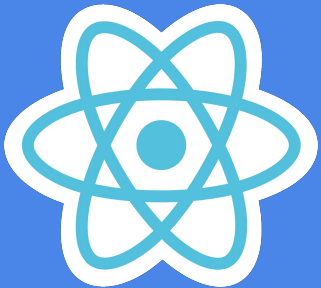
Debugging, HTTP-Mocking

Screenshots/Videos

Device-Emulation

Cucumber

BDD



Cucumber



Behaviour-Driven Development

- Definition: **Kein Testing-Tool**, Collaboration/Analyse-Tool
- Dokumentation von groben Features
- Explizite Beispiele, die zeigen, was Software machen soll
- Szenarios werden vor Produktions-Code geschrieben
 - Acceptance Driven Development

Cucumber + Gherkin

Nutzt spezielle Grammatik:

“Gherkin”

Feature: Show Premium Items
Display the title

Scenario: Home Page - Premium

Given I am on the home page

And I am "Hilde Hirsch" with ID "100000"

When I click submit

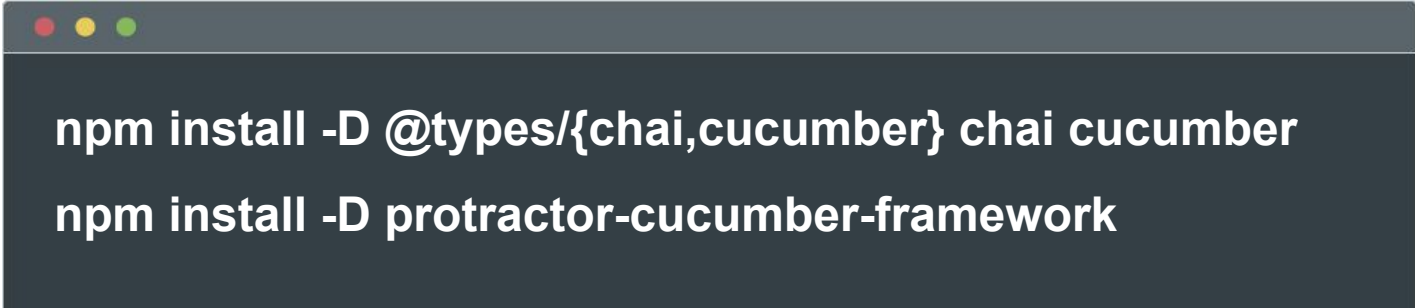
Then I should see Premium-Offers

Cucumber + Gherkin

It is usually counterproductive to let product owners and business analysts write Gherkin.

Instead, we recommend they participate in Example Mapping sessions and approve the Gherkin documents after a developer or tester has translated it to Gherkin.

Cucumber + Gherkin

A terminal window with a dark background and a light gray title bar. The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal contains two lines of white text representing npm commands.

```
npm install -D @types/{chai,cucumber} chai cucumber  
npm install -D protractor-cucumber-framework
```

- Bei erstem Testlauf: Cucumber gibt Test-Implementierung vor

Selfie

Screenshots der App

Protractor Screenshots

- Protractor besitzt API für Screenshots
- Manuell getriggert
- Einfach zu erweitern
 - Bei jedem Testfall
 - Im Fehlerfall

```
const fname =
  `${testDescription.replace(/\s/g, '_')}.png`;

browser.takeScreenshot()
  .then((png) => {
    const stream = fs
      .createWriteStream(path.join(dir, fname));
    stream.write(new Buffer(png, 'base64'));
    stream.end();
  });
```

puppeteer

- Verwendet Chrome API
- Default: headless-mode
- Kann zur Umsetzung von e2e-Tests und Screenshots genutzt werden

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  page.setViewport({width: 1280, height: 768});
  await page.goto('http://angular-quality.com/',
    {waitFor: 'networkidle2'});
  await page.screenshot({
    path: 'shots/startpage.png'
  });
  await page.pdf({
    path: 'shots/startpage.pdf',
    format: 'A4'
  });
  await browser.close();
})();
```

Visual Diff

Ayes

Visual Diff

Hilfe bei:

- Technischem Update (Framework)
- Style Änderungen, "Wo tut sich was?"

```
npm i -g aye-spy  
  
ayespy init
```

```
{  
  "gridUrl": "http://localhost:4444/wd/hub",  
  "baseline": "./aye/baseline",  
  "latest": "./aye/latest",  
  "generatedDiffs": "./aye/generatedDiffs",  
  "report": "./aye/reports",  
  "scenarios": [{  
    "url": "http://localhost:4200/",  
    "label": "label",  
    "onReadyScript":  
      "./aye/scripts/premium.js",  
    "viewports": [{  
      "height": 1500,  
      "width": 1024,  
      "label": "large"  
    }]  
  }]  
}
```

Visual Diff



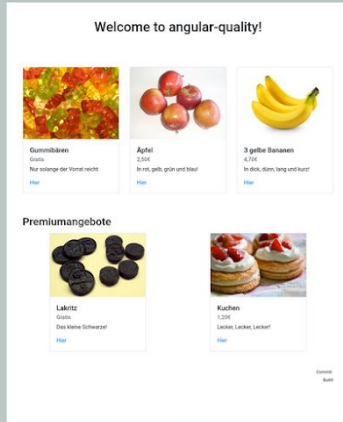
```
ayespy snap --browser chrome --config config.json
```

```
ayespy update-baseline --browser chrome --config config.json
```

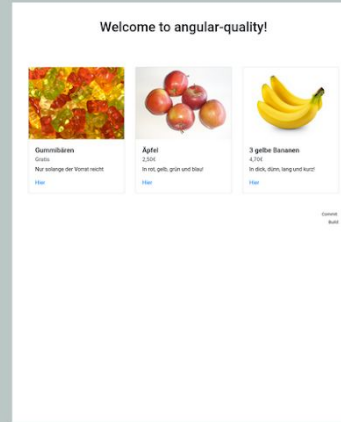
```
ayespy compare --browser chrome --config config.json
```

Scenario: label-large

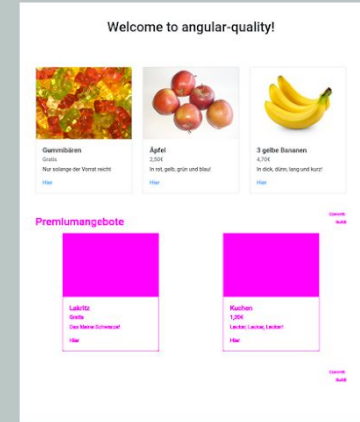
Baseline



Latest



Difference



Weitere Optionen:

- Elemente Entfernen (z.B. Werbebanner etc.)
- Nur Bestimmten Seitenausschnitt aufnehmen

Visual Diff - Anwendungsfälle

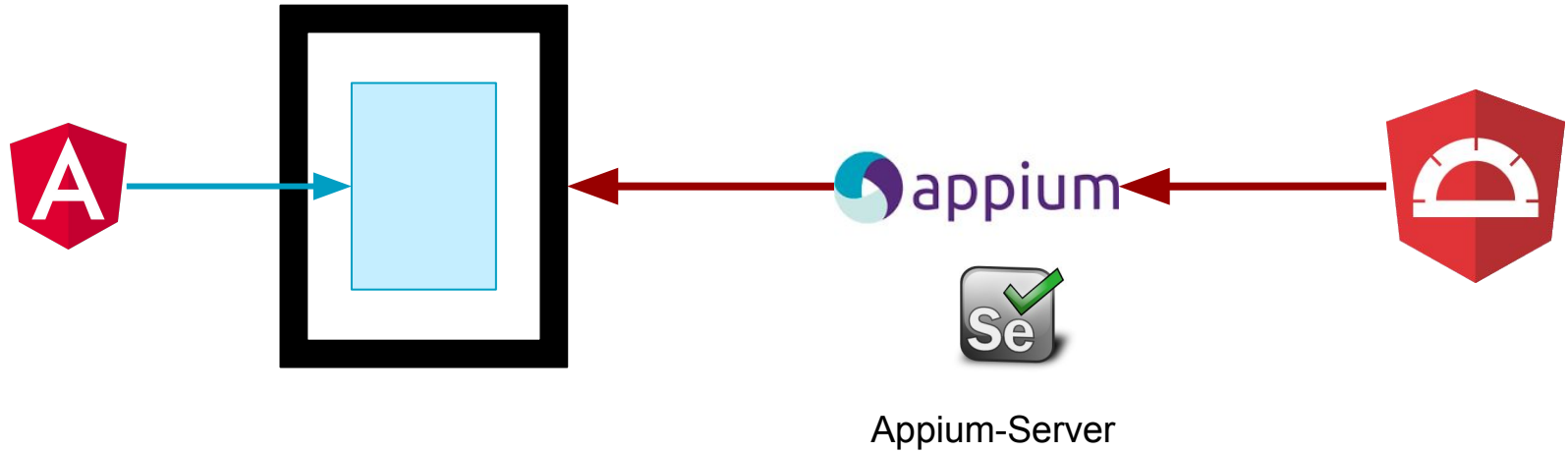
- Anwendungsteile, welche schwer funktional testbar sind
 - Durch Anwendung generierte Grafiken (Canvas, Graphen)
 - CSS-Styling (z.B. Diff einer Kitchen-Sink)
- Erfordert reproduzierbare/stabile Ausgabe der Anwendung
 - Statische Ausgabe oder Umssysteme faken/mocken

Devices

Testen mit Appium

Appium

Architektur



Appium

```
npm i -g appium
```

```
appium
```

```
ng serve --host 0.0.0.0
```

```
protractor ./e2e/protractor.conf.js
```

```
multiCapabilities: [  
  {  
    browserName: 'Safari',  
    platformName: 'iOS',  
    platformVersion: '12.1',  
    deviceName: 'iPhone X',  
    automationName: 'XCUITest'  
  },  
  {  
    avd: 'Nexus_5X_API_25',  
    browserName: 'Chrome',  
    platformName: 'Android',  
    platformVersion: '7.1.1',  
    deviceName: 'Android Emulator',  
    automationName: 'Appium',  
    skipDeviceInitialization: true,  
    'goog:chromeOptions': {  
      args: [  
        "--no-first-run",  
        "--disable-fre"  
      ]  
    }  
  }  
]
```

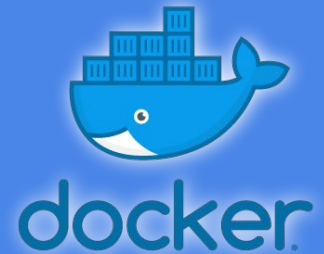
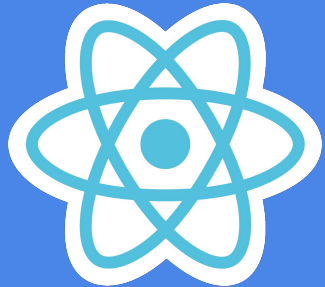
```
protractor.conf.js  
(Ausschnitt)
```

Live Demo

Docker / Container

Lokale/Isolierte Umgebung
(DB, Backend, Umsysteme)

CI Server



Docker Angular-CLI Images

- trion/ng-cli
 - Angular-CLI, node, npm, yarn
 - <https://hub.docker.com/r/trion/ng-cli/>
- trion/ng-cli-karma
 - Chrome Browser, xvfb
 - <https://hub.docker.com/r/trion/ng-cli-karma/>
- trion/ng-cli-e2e
 - Java, webdriver
 - <https://hub.docker.com/r/trion/ng-cli-e2e/>
- node.js, derzeit Version 10
- npm und yarn Package Manager
- Getestet mit Linux, macOS, Win
- Version folgt Angular-CLI Version
 - z.B. trion/ng-cli:**7.0.2**

testcontainers

Integration Docker mit JUnit

Testcontainers - für Java Entwickler

- **Docker Integration als JUnit Rule**
 - Beliebige Images nutzbar
- **Spezieller Support für einige Dienste**
 - Postgresql
 - Mysql
 - Kafka
- **Spezielle Anwendungsfälle**
 - Testwerkzeuge im Container, z.B. Webdriver



Beispiel für Postgresql Datenbank (und Spring Boot)

@ClassRule

```
public static PostgreSQLContainer postgres = new PostgreSQLContainer()
    .withDatabaseName("training")
    .withUsername("spring")
    .withPassword("spring");

public static class Initializer implements ApplicationContextInitializer <...> {
    @Override
    public void initialize(ConfigurableApplicationContext ctx){
        TestPropertyValues values = TestPropertyValues.of(
            "spring.datasource.url=" + postgres.getJdbcUrl()
        );
        values.applyTo(ctx);
    }
}
```

UI Tests mit Testcontainer

Spezieller Container mit Webbrowser: Firefox, Chrome

Aufzeichnung des Browsers als Video: Alle Tests / Nur fehlgeschlagene Tests

```
@Rule
public BrowserWebDriverContainer chrome =
    new BrowserWebDriverContainer()
        .withDesiredCapabilities(DesiredCapabilities.chrome())
        .withRecordingMode(RECORD_ALL, new File("target"));
```

Webdriver / Selenium API

```
@LocalServerPort
private int randomServerPort; //spring app port

@Test
public void simpleWebdriverTest() throws Exception {
    final RemoteWebDriver driver = chrome.getWebDriver();
    driver.get("http://localhost:" + randomServerPort);

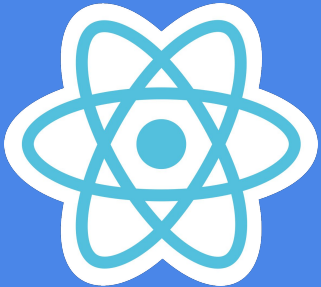
    final WebElement heading = driver.findElementByTagName("h1");
    assertThat(heading.getText(), is("Training Sample application"));
    TimeUnit.SECONDS.sleep(2); //longer video
}
```

Testcontainers demo

Automatisierung

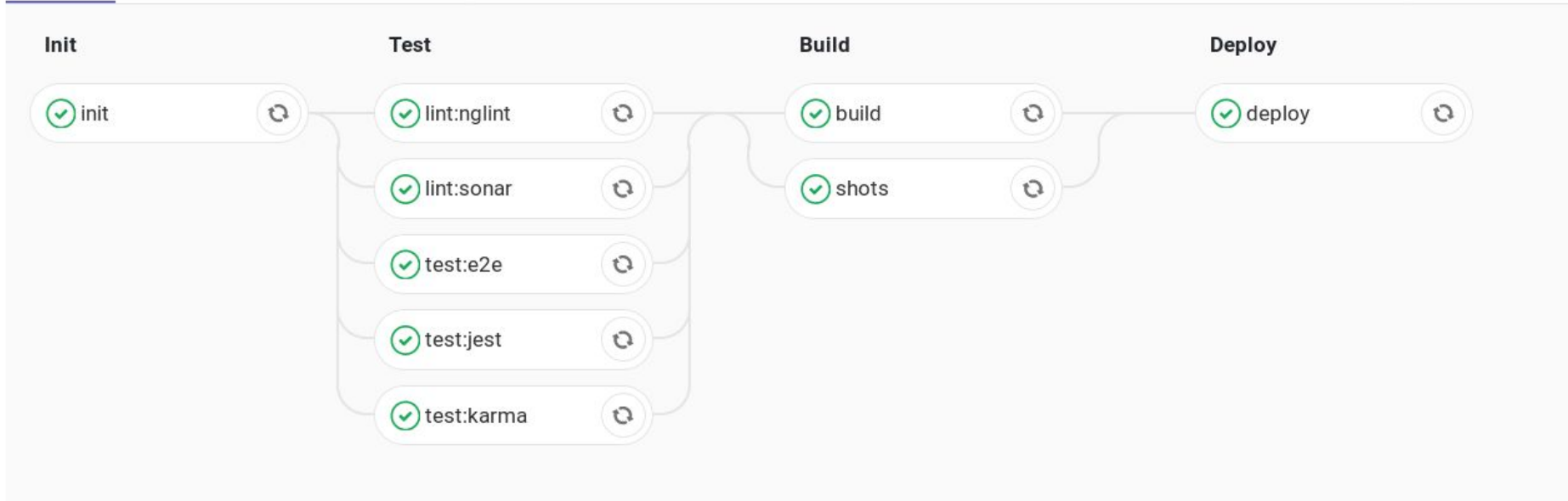
CI Server

Beispiel GitLab-CI




Build / Deployment Pipeline


Pipeline Jobs 9



GitLab CI: Test-Job

```
test:karma:  
  stage: test  
  image: trion/ng-cli-karma:${CLI_VERSION}  
  allow_failure: false  
  script:  
  - ng test --code-coverage --progress false --watch false  
  coverage: '/Lines \W+: (\d+\.\d+)%.*/'  
  artifacts:  
    paths:  
    - angular-quality/coverage/  
  tags:  
  - docker  
  - gce
```

 passed	#405 x86_64 docker	lint	🕒 01:21 📅 5 minutes ago	
 passed	#406 x86_64 docker	test:karma	🕒 01:27 📅 5 minutes ago	92.86%



GitLab CI: Sonar-Job

lint:sonar:

stage: test

image: trion/ng-cli:\${CLI_VERSION}

script:

- npm install -g sonarqube-scanner

- >

sonar-scanner

-Dsonar.projectKey=demo

-Dsonar.organization=everflux-github

-Dsonar.host.url=https://sonarcloud.io

-Dsonar.login=covfefe

-Dsonar.typescript.lcov.reportPaths=coverage/lcov/lcov.info

-Dsonar.sourceEncoding=UTF-8

-Dsonar.sources=src/app

-Dsonar.exclusions=**/node_modules/**, **/*.spec.ts

-Dsonar.tests=src/app

-Dsonar.test.inclusions=**/*.spec.ts

Danke.

Fragen?

Slides, Demo Repo



@kakulty

sitterberg.com

trion.de