ANGULAR TESTING IN VIGO NG

INHALT

- Tools
- Angular Testing
- Types of Tests
- VIGO NG Testing

TOOLS

KARMA

```
1 09 04 2019 21:03:33.731:INFO [karma]:
2 09 04 2019 21:03:33.732:INFO [launcher]:
3 09 04 2019 21:03:33.741:INFO [launcher]:
4 09 04 2019 21:03:59.049:WARN [karma]:
5 09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]:
6
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
1 09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
2 09 04 2019 21:03:33.732:INFO [launcher]:
3 09 04 2019 21:03:33.741:INFO [launcher]:
4 09 04 2019 21:03:59.049:WARN [karma]:
5 09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]:
6
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
1 09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
2 09 04 2019 21:03:33.732:INFO [launcher]: Launching browser ChromeCustom with unlimited concurrency
3 09 04 2019 21:03:33.741:INFO [launcher]:
4 09 04 2019 21:03:59.049:WARN [karma]:
5 09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]:
6
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
1 09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
2 09 04 2019 21:03:33.732:INFO [launcher]: Launching browser ChromeCustom with unlimited concurrency
3 09 04 2019 21:03:33.741:INFO [launcher]: Starting browser Chrome
4 09 04 2019 21:03:59.049:WARN [karma]:
5 09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]:
6
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
09 04 2019 21:03:33.732:INFO [launcher]: Launching browser ChromeCustom with unlimited concurrency
09 04 2019 21:03:33.741:INFO [launcher]: Starting browser Chrome
09 04 2019 21:03:59.049:WARN [karma]: No captured browser, open http://localhost:9876/
09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]:
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
1 09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
2 09 04 2019 21:03:33.732:INFO [launcher]: Launching browser ChromeCustom with unlimited concurrency
3 09 04 2019 21:03:33.741:INFO [launcher]: Starting browser Chrome
4 09 04 2019 21:03:59.049:WARN [karma]: No captured browser, open http://localhost:9876/
5 09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]: Connected on socket ... with id 0
6
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

KARMA

```
09 04 2019 21:03:33.731:INFO [karma]: Karma v1.7.1 server started at http://0.0.0.0:9876/
09 04 2019 21:03:33.732:INFO [launcher]: Launching browser ChromeCustom with unlimited concurrency
09 04 2019 21:03:33.741:INFO [launcher]: Starting browser Chrome
09 04 2019 21:03:59.049:WARN [karma]: No captured browser, open http://localhost:9876/
09 04 2019 21:03:59.500:INFO [HeadlessChrome 73.0.3683 (Linux 0.0.0)]: Connected on socket ... with id 0
```

Speaker notes

- 1 Zeile: Karma server
- 2 Zeile: Karma startet einen browser als client
- 3 Zeile: Browser startet
- 4 Zeile: Browser capturing, debugging in chrome etc.
- 5 Zeile: Socket connection für server -> browser crosstalk
- 6 Zeile: Tests running

- Test Runner
- Debugging
- Browser und Plattform unabhängig

JASMINE

```
describe("a suite", () => {
  it("should add 2 and 2 to 4", () => {
    expect(2 + 2).toBe(4);
  });

  it("should concatenate strings", () => {
    const a = 'someName';
    expect(`this is my name: ${a}`).toBe('this is my name: someName');
  });
});
```

Speaker notes

- Fast
- BDD for JavaScript
- Spying

JASMINE MATCHERS

```
1  // not.(other-matcher)
2  expect(something).not.toBe(true);
3
4  // equals
5  expect(something).toBe(true);
6
7  // equals float
8  expect(something).toBeCloseTo(expected, precision);
9  expect(number).toBeCloseTo(42.2, 3);
10
11 //contains
12 expect(array).toContain(anElement);
13
14 //error
15 expect(aFunction()).toThrow(anError);
```

TS-MOCKITO

```
1  // from ts-mockito docs:
2  // Creating mock
3  let mockedFoo:Foo = mock(Foo);
4
5  // stub method before execution
6  when(mockedFoo.getBar(3)).thenReturn('three');
7
8  // Getting instance
9  let foo:Foo = instance(mockedFoo);
10
11 // prints three
```

Speaker notes

- When to stub methods doesnt matter
- pass instance to logic
- use mock to stub and verify
- can spy on real objects
- for java developers its basically the same, except for the instance

ANGULAR TESTING

TESTBED & TESTING MODULE

```
let component: BannerComponent;
let fixture:   ComponentFixture<BannerComponent>;
let h1:        HTMLElement;

beforeEach(() => {
  TestBed.configureTestingModule({
    declarations: [BannerComponent],
    providers: [],
    imports: []
  });

  fixture = TestBed.createComponent(BannerComponent);
  component = fixture.componentInstance;
  h1 = fixture.nativeElement.querySelector('h1');
```

Speaker notes

- Angular Modules for Testing
- Bootstraps Angular

HTTP CLIENT TESTING MODULE

```
TestBed.configureTestingModule({
    //...
    imports: [HttpClientTestingModule]
});
```

```
const req = controller.expectOne(`myBasePath/gfs/${beilage.gfId}/beilagen`);

expect(req.request.method).toBe('POST');
const body: FormData = req.request.body;
expect(body.get('art')).toBe(beilage.art);
//...
req.flush('A', {headers: new HttpHeaders()});
```

Speaker notes

- Can be used to make sure requests are correctly sent to be.

TESTBED PROS/CONS

Positives	Negatives
• Dependency injection • Importing modules • Mocking with injection • Less code in tests • Tests module configuration	• Performance

TYPES OF TESTS

SHALLOW TESTS

- Shallow-render
- Template testing
- Component bootstrap testing
- NgOnInit template set-up testing

Speaker notes

- Shallow-render mit schema 'NO_ERRORS_SCHEMA'
- soll nicht die Controller Logik testen, sondern nur die template

ISOLATED TESTS

- Controller testing
- Isolated from other units
- Mock every dependency (except ZEntity related things)
- No Angular bootstrap

Speaker notes

- real unit-test
- no testbed, no module

SERVICE TESTS

- Isolated from other units
- Mock every dependency (except ZEntity related things)
- TestBed vs non-TestBed tests

Speaker notes

- semi-real unit-test for services
- no testbed, no module

INTEGRATION/E2E TESTS

- Protractor
- Page-object pattern
- Testing multiple components together, including the services

Speaker notes

- important for caching configuration issues, and wrong use of components, because shallow tests dont cover that.

VIGO NG TESTING