

# 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
```

# 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
```

# 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

# 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
```

# 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)]:
```

6



# 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
```

# 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 HeadlessChrome 73.0.3683 (Linux 0.0.0): Executed 81 of 424 SUCCESS (0 secs / 18.735 secs)
```

# JASMINE

```
1 describe("a suite", () => {
2   it("should add 2 and 2 to 4", () => {
3     expect(2 + 2).toBe(4);
4   });
5
6   it("should concatenate strings", () => {
7     const a = 'someName';
8     expect(`this is my name: ${a}`).toBe('this is my name: someName');
9   });
10 });
```

# 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
12 console.log(foo.getBar(3));
13
14 // verify called
15 verify(foo.getBar(3)).called();
16
```

# ANGULAR TESTING

# TESTBED & TESTING MODULE

```
1 let component: BannerComponent;
2 let fixture: ComponentFixture<BannerComponent>;
3 let h1: HTMLElement;
4
5 beforeEach(() => {
6   TestBed.configureTestingModule({
7     declarations: [BannerComponent],
8     providers: [],
9     imports: []
10  });
11
12  fixture = TestBed.createComponent(BannerComponent);
13  component = fixture.componentInstance;
14  h1 = fixture.nativeElement.querySelector('h1');
15 });
```

# HTTP CLIENT TESTING MODULE

```
1 TestBed.configureTestingModule({  
2     //...  
3     imports: [HttpClientTestingModule]  
4 });
```

```
1 const req = controller.expectOne(`myBasePath/gfs/${beilage.gfId}/beilagen`);  
2  
3 expect(req.request.method).toBe('POST');  
4 const body: FormData = req.request.body;  
5 expect(body.get('art')).toBe(beilage.art);  
6 //...  
7 req.flush('A', {headers: new HttpHeaders()});
```



# TESTBED PROS/CONS

## Positives

- Dependency injection
- Importing modules
- Mocking with injection
- Less code in tests
- Tests module configuration

## Negatives

- Performance

# TYPES OF TESTS

# SHALLOW TESTS

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

# ISOLATED TESTS

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

# SERVICE TESTS

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

# INTEGRATION/E2E TESTS

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

# VIGO NG TESTING