codecentric

Angular Testing

Schulung part 3

20.11.2018

Überblick

Angular testing

- **Karma**
  - Struktur
  - Konfiguration
- **Unit Testing**
  - Pros und Cons
  - Service testing
  - Component testing
- **e2e**
  - Pros und Cons
  - Konfiguration
  - Workflow testing

“No amount of testing can prove a software right, a single test can prove a software wrong.”

Amir Ghahrai

Karma

Karma

Struktur

- **testrunner basierend auf frameworks**
    - Z.B. Jasmine

```
describe('filter utils', () => {
  it('mannschaft match filter', () => {
     …
  });

  it('mannschaft match filter nicht', () => {
     const schalkeFilter = matchFilter('Schalke');
     expect(schalkeFilter({TeamName: 'Hsv', Points: 10, TeamIconUrl:
     ''})).toBeFalsy();

  });
});
```

Konfiguration

- Framework
- Files
- Browser
- Plugins
- Coverage

Unit Testing

Pros und Cons

- **Pro**
  - Schnelle Ausführung
  - Service Basiert
  - Komponenten Basiert
- **Cons**
  - Kein workflow Test
  - Kein Test von zusammenhängen

Service testing

- Logik möglichst isoliert testen
- Keine UI
- TestBed ermöglicht services zu injecten
- Http mock
- Observable testing

Service testing - setup

```
beforeEach(() => {
  TestBed.configureTestingModule({
    imports: [ HttpClientTestingModule],
    providers: [BundesligaDatenService]
  });
  httpTestingController = TestBed.get(HttpTestingController);
  service = TestBed.get(BundesligaDatenService);
});
```

Service testing - test

```
it('load data ', async(() => {
  service.loadTabelle(Liga.ErsteLiga);
  const req = httpTestingController.expectOne('https://www.openligadb.de/api/getbltable/bl1/2018');

  const teamData : TabelleMannschaft[] = [{TeamName: 'Bla2',Points:10,TeamIconUrl:''}];
  req.flush(teamData);

  service.tabelle$.subscribe((daten: TabelleMannschaft[]) => {
    expect(daten.length).toBe(1);
    expect(daten[0].TeamName).toBe('Bla2');
  });

}));
```

Component testing

- Component logik Testing
- TestBed ermöglich inject von Component
- Async testing

Component testing - setup

```
beforeEach(() => {
  TestBed.configureTestingModule({
    declarations: [ MannschaftFilterComponent ]
  });
  fixture = TestBed.createComponent(MannschaftFilterComponent);
  comp = fixture.componentInstance;
});
```

Component testing - test

```
it('typing', async(() => {
  comp.onFilterChange.subscribe((value)=>
    expect(value).toBe('Test Value')
  );
  comp.onKey({target:{value:'Test Value'}});
 }));
```

e2e

Pros und Cons

- **Pro**
  - Testing von zusammenhängen
  - Workflows
- **Cons**
  - langsam
  - kein isoliertes testing von Components / Services

Konfiguration

- timeouts
- ports
- files
- browser

Workflow testing - protractor

- protractor (cheatsheet)

```
import { browser, by, element } from 'protractor';

export class AppPage {
  navigateTo() {
    return browser.get('/');
  }

  getFilterInput() {
    return element(by.id('filterInput'));
  }

  findElementsByCss(css: string){
    return element.all(by.css(css));
  }

  wait(time:number){
    browser.sleep(time);
  }
}
```

Workflow testing - test

```
import { AppPage } from './app.po';

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

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

  it('should filter', () => {
    page.navigateTo();

    const all = page.findElementsByCss('.mannschaft');
    expect(all.count()).toEqual(18);

    page.getFilterInput().sendKeys('Schal');

    const filtered = page.findElementsByCss('.mannschaft');
    expect(filtered.count()).toEqual(1);

  });
});
```