



## Inhalt

- Überblick zu Jasmine
- Jasmine und Angular
- DEMO

SOFTWAREarchitekt.at



## Überblick zu Jasmine

### Anatomie eines Jasmine-Tests

```
describe("Object under test", function () {  
  
    beforeEach(function () { ... });  
    afterEach(function () { ... });  
  
    it("should do this", function () {  
        let r = add(1, 2);  
        expect(r).toBe(3);  
    });  
  
    it("should do that", function () {  
        let r = add(1, 11);  
        expect(r).toBe(12);  
    });  
  
});
```

**SOFTWARE**architekt.at

## Karma

- Testrunner für Unit-Tests
- Kommandozeile (und somit CI)
- Startet Browser
- Erzeugt Protokolle (z. B. im JUnit-Format)

SOFTWAREarchitekt.at

## Ausführen

- npm test
- ng test
- karma start
- Via WebStorm/ IntelliJ

SOFTWAREarchitekt.at

## Jasmine und Angular



## TestBed

```
beforeEach(async(() => {  
    TestBed.configureTestingModule({  
        imports: [HttpClientModule, ...],  
        declarations: [...],  
    })  
    .compileComponents();  
}));
```

## Unit-Test

```
it('should have no selected flight initially', () => {  
  let flightSearchFixture =  
    TestBed.createComponent(FlightSearchComponent);  
  
  [...]  
});
```

SOFTWAREarchitekt.at

## Unit-Test

```
it('should have no selected flight initially', () => {  
  let flightSearchFixture =  
    TestBed.createComponent(FlightSearchComponent);  
  
  let flightSearchComponent =  
    flightSearchFixture.componentInstance;  
  
  expect(flightSearchComponent.selectedFlight)  
    .toBeUndefined();  
});
```

SOFTWAREarchitekt.at

## Asynchrone Tests

```
it('should load flights', async(() => {  
    [...]  
}));
```

SOFTWAREarchitekt.at

## Asynchrone Tests

```
it('should load flights', (done: Function) => {  
    [...]  
    done();  
});
```

SOFTWAREarchitekt.at

## Services testen

```
it('should load flights', () => {  
  let flightService = TestBed.get(FlightService);  
  [...]  
});
```

SOFTWAREarchitekt.at

# DEMO

SOFTWAREarchitekt.at

## HttpClientTestingModule

```
beforeEach(async(() => {  
  TestBed.configureTestingModule({  
    imports:[FormsModule, HttpClientTestingModule],  
    declarations: [ FlightSearchComponent ]  
  })  
  .compileComponents();  
}));
```

**SOFTWARE**architekt.at

## HttpTestingController

```
it('...', () => {  
  component.from = 'Graz';  
  component.to = 'Hamburg';  
  component.search();  
  
  [...]  
});
```

**SOFTWARE**architekt.at



## HttpTestingController

```
it('...', () => {  
  component.from = 'Graz';  
  component.to = 'Hamburg';  
  component.search();  
  
  let httpTestingController: HttpTestingController =  
    TestBed.get(HttpTestingController);  
  
  [...]  
});
```

SOFTWAREarchitekt.at

## HttpTestingController

```
it('...', () => {  
  component.from = 'Graz';  
  component.to = 'Hamburg';  
  component.search();  
  
  let httpTestingController: HttpTestingController =  
    TestBed.get(HttpTestingController);  
  
  const req = httpTestingController.expectOne('http://www.angular.at/...');  
  
  [...]  
});
```

SOFTWAREarchitekt.at

## HttpTestingController

```
it('...', () => {
  component.from = 'Graz';
  component.to = 'Hamburg';
  component.search();

  let httpTestingController: HttpTestingController =
    TestBed.get(HttpTestingController);

  const req = httpTestingController.expectOne('http://www.angular.at/...');

  req.flush([{ id: 22, from: 'Graz', to: 'Hamburg', date: '' }]);

  [...]
});
```

SOFTWAREarchitekt.at

## HttpTestingController

```
it('...', () => {
  component.from = 'Graz';
  component.to = 'Hamburg';
  component.search();

  let httpTestingController: HttpTestingController =
    TestBed.get(HttpTestingController);

  const req = httpTestingController.expectOne('http://www.angular.at/...');

  req.flush([{ id: 22, from: 'Graz', to: 'Hamburg', date: '' }]);

  expect(component.flights.length).toBe(1);
});
```

SOFTWAREarchitekt.at

# DEMO

SOFTWAREarchitekt.at

## Globale Provider überschreiben

```
beforeEach(async(() => {  
    TestBed.configureTestingModule({  
        imports: [HttpModule, ...],  
        declarations: [],  
        providers: [  
            { provide: FlightService, useClass: FlightServiceMock },  
            { provide: BASE_URL, useValue: '' }  
        ]  
    }).compileComponents();  
}));
```

SOFTWAREarchitekt.at

## Provider für Komponente überschreiben

```
[...]
TestBed.overrideComponent(FlightSearchComponent, {
    set: {
        providers: [{
            provide: FlightService,
            useClass: FlightServiceMock
        }]
    }
})
.compileComponents();
[...]
```

SOFTWAREarchitekt.at

DEMO

SOFTWAREarchitekt.at



# E2E-Testing mit Protractor

## Protractor

- E2E-Test-Framework für AngularJS
- Interagiert mit Browser
- Simuliert Benutzereingaben
- Unterstützt Jasmine

## Erste Schritte

```
describe("FlugApp", function() {  
  
  beforeEach(function() {  
    browser.get('http://localhost:8080/');  
  });  
  
  it('should load page and read title', function() {  
  
    var expectedTitle = 'FlugApp';  
    expect(browser.getTitle()).toBe(expectedTitle);  
  
  });  
});
```

↑  
**Protractor löst Promises automatisch auf**

SOFTWAREarchitekt.at

## Weiteres Beispiel

```
var link = element(by.linkText("Buchen"));  
link.click();  
  
var von = element(by.name("von"));  
var nach = element(by.name("nach"));  
var suchen = element(by.css("button"));  
  
von.clear(); nach.clear();  
von.sendKeys("Graz");  
nach.sendKeys("Hamburg");  
suchen.click();  
  
var fluege = element.all(by.tagName("tr"));  
  
expect(fluege.count()).toBe(4);
```

**Locator** (points to `by.linkText`, `by.name`, `by.css`, and `by.tagName`)

**ElementFinder** (points to `element`)

**ElementArrayFinder** (points to `element.all`)

SOFTWAREarchitekt.at

DEMO

**SOFTWARE** *architekt.at*