die Mobiliar

Testen mit Protractor

Jan Kuonen

Inhaltsangabe

1. **Testautomatisieren: Sinn und Zweck**
2. **Technologischer Hintergrund**
3. **Getting Started**
4. **Architektur und Aufbau**
5. **Fazit**

Testautomatisierung

Sinn und Zweck?

ReguläresTesten

- Tester führt manuell Tests aus

Vorteile

- Tester ist schlau
- Tester hat Intuition

Nacheile

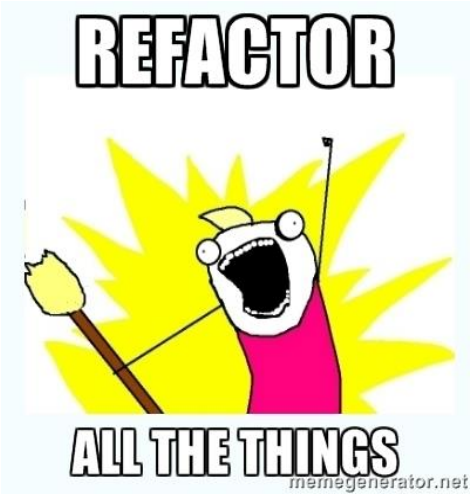
- Manuelles Ausführen
- Fehleranfällig bei Regression

Testautomatisierung

Problem

Was ist wenn sich was verändert?

- Markup
- Logik
- Datenpersistenz



Beispiel an Dokument Upload eDossier

- Veränderungen von aussen
- Neue Dokumente können hochgeladen werden
- Ein Eingabefeld existiert nicht mehr

REASON => wartbarer Aufbau…

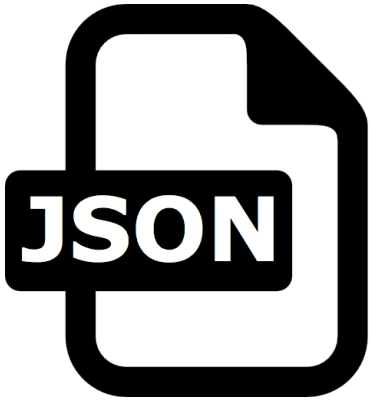
Testautomatisierung
Summary

Was wollen wir?

- Tests welche immer wieder ausgeführt werden
- Tests welche bei Veränderungen einfach anpassbar sind!
- Datenpersistenz

Technischer Hintergrund
Was verwenden wir?

- Protractor
- NodeJs
- CSS
- JSON



Technischer Hintergrund Protractor

Vorteile

- Interagiert selbstständig mit dem Angular Modell
- Open Source
- Einfach zu lernen

Nachteile

- Stabilität
- Nur Webapps
- Keine oder nur bedingte Interaktion mit OS Komponenten

Getting Started
oder wie bring ich das Ding zum laufen

```
jan@getting_started/$ npm install protractor --save-dev
jan@getting_started/$ npm install webdriver --save-dev
jan@getting_started/$ webdriver update --Chrome
jan@getting_started/$ webdriver update
jan@getting_started/$ protractor test.js
```

Architektur und Aufbau

Leichtes Leben dank Selektoren

```
<html>
    <div class=‘e2e-dossier-formular’’>
        <input class=‘’ e2e-dossier-edit’’/>
        <buton class=’’e2e-dossier-
commit-button’’ ></button>
    </div>
</html>
```

```
<html>
    <div class=‘e2e-dossier-formular’’>
        <table>
            <td>
            <tr>
                <input class=‘’e2e-dossier-
edit’’/>
            </tr>
            </td>
        </table>
        <buton class=’’e2e-dossier-
commit-button’’ >
        </button>
    </div>
</html>
```

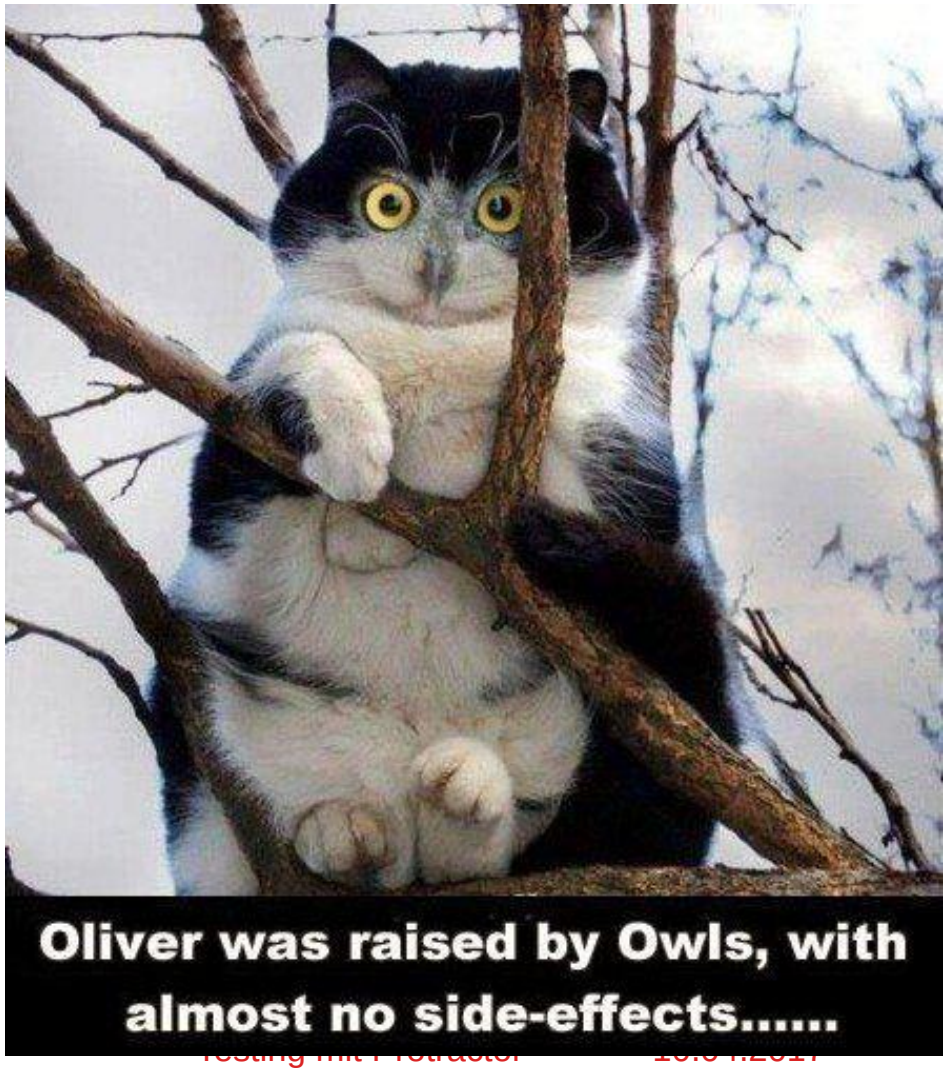
Architektur und Aufbau
Config over all

Side Effect

- Nach ungefähr 20 Tests

JSON CONFIG

- Herzstück des ganzen
- Testdaten und Ablauf in JSON Config definiert
- Redundanzen werden verringert



Architektur und Aufbau

UI-Suite	- Büdel der verschiedenen Tests - Description Log
tests	- Gekapselte Testfälle – Testschritte - In Methoden gekapselt
pom	- POM steht für Page Object Model - Zugriff auf entsprechenden Selektor
pomutils	- Spezielle sich wiederholende Aktionen

Fazit

- Methode hat sich bewährt
- Stabilität ist schwierig zu gewährleisten (Umsysteme)
- Gleiche Probleme => gleiche Lösungen

Fragen?

Demo Time