

Testbericht  
Nachrichtenkommunikation für das THW

na17b

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>2</b>
<b>2</b>	<b>Tests</b>	<b>2</b>
	Komponententests . . . . .	2
	GUI-Tests . . . . .	2
<b>3</b>	<b>Continuous Integration</b>	<b>3</b>

# 1 Allgemeines

Die Tests für dieses Softwareprojekt sind eng an das Qualitätssicherungskonzept geknüpft und sollen das Funktionieren des Codes gewährleisten. Im Projekt werden ausschließlich Komponententests durchgeführt, da die Kommunikation zwischen Komponenten nur indirekt über den Store erfolgt. Der Zugriff auf den Store wurde gemocked. Als Testframework wird Jest zusammen mit vue-test-utils verwendet.

## 2 Tests

### Komponententests

Komponententests befinden sich jeweils im gleichen Ordner wie die zu testende Komponente.

### GUI-Tests

Die Testspezifikationen befinden sich vom frontend-Verzeichnis aus gesehen in `test/unit/specs`. Dort liegt für jede Komponente eine eigene Datei, welche die geforderten Eigenschaften und Funktionen einer Komponente beschreibt. Es genügt, im Ordner frontend den Befehl `npm run unit` auszuführen; daraufhin werden alle Test-Suites automatisch abgearbeitet. Eine beispielhafte Ausgabe ist in folgender Abbildung zu sehen.

```
PASS src/test-examples/counter.spec.js
Counter
  ✓ renders the correct markup (11ms)
  ✓ has a button (12ms)
  ✓ button click should increment the count (10ms)

PASS test/unit/specs/sparql_queries.spec.js
  ✓ Dashboard query function works (6ms)
  ✓ FormQuery function works (1ms)
  ✓ InsertQuery function works (1ms)
  ✓ DeleteQuery function works (1ms)
  ✓ operationToInsertQuery function works
  ✓ Operations query function works

PASS test/unit/specs/sparql_response.spec.js
  ✓ sparql response parsed correctly (5ms)

PASS test/unit/specs/THWMenu.spec.js
THWMenu
  ✓ let's you choose the dashboard (22ms)
  ✓ let's you choose the ticket creation mask (3ms)
  ✓ let's you open the saved draft (3ms)

PASS test/unit/specs/THWLandingPage.spec.js
THWLandingPage
  ✓ binds to Hdz (35ms)
  ✓ binds to name (11ms)
  ✓ binds to Hdz (13ms)
  ✓ binds to operation (10ms)

Test Suites: 6 passed, 6 total
Tests:       25 passed, 25 total
Snapshots:  0 total
Time:        2.665s
Ran all test suites.
```

Abbildung 1: Ausgabe des Befehls `npm run unit`

Die Tests umfassen das Prüfen von Anwesenheit bestimmter Variablen und html-Elemente, sowie die Rückgabewerte einiger Funktionen und das Parsen von sparql-Rückgaben. Die Ergebnisse sind in folgender Tabelle aufgelistet. Die Tests für THWForm werden derzeit nicht

Komponente	Anzahl Tests	Bestanden
THWMenu	3	ja
THWLandingPage	4	ja
sparql_response	1	ja
sparql_queries	6	ja

Tabelle 1: Testergebnisse der momentanen Frontend-Komponenten

ausgeführt, weil es zu Komplikationen mit Jest kam, welche noch nicht behoben werden konnten.

### 3 Continous Integration

Zum zweiten Release wurde Continous Integration eingeführt, um das Einhalten der Vorgaben aus dem Dokumentationskonzept und Coding Standards automatisiert zu testen. Dazu wird GitLab CI verwendet. Die `gitlab-ci.yml` beinhaltet XML-Linting sowie JavaScript-Linting und -Testing. Um den Prozess zu beschleunigen wird auf `npm install` verzichtet, stattdessen werden nur für die Tests benötigte Pakete und deren Abhängigkeiten installiert. Anschließend erfolgen linting (`npm run lint`) und testing (`npm run unit`).

