

A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date. In the bottom left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

26.11.2014

GIBM Stundenplan

Blackbox Testplan

Dimitri Vranken

GEWERBLICH-INDUSTRIELLE BERUFSFACHSCHULE MUTTENZ

VORWORT

Ich habe mich bei diesem Projekt dafür entschieden das Blackbox-Testing nicht manuell sondern mit automatisierten UI Tests umzusetzen.

Der Grund dafür war es Erfahrungen mit automatisiertem UI Testing von Webseiten und JavaScript Unit Tests zu sammeln.

TESTS

Die Ergebnisse aller Tests werden im Ordner `/test/results/` protokolliert.

UNIT TESTS

Alle Unit Tests befinden sich in Ordner `/test/unit/`. Die Konfiguration befindet sich in `/build/tasks/qunit.js`. Es existieren neun Unit Tests mit insgesamt 45 Assertions¹.

Die Unit Tests können im Gegensatz zu den UI Tests ohne lokalen Webserver oder vorherigen Build des Source Codes ausgeführt werden.

SMOKE TESTS (UI TESTS)

Alle Unit Tests befinden sich im Ordner `/test/smoke/`. Die Konfiguration befindet sich in `/build/tasks/nightwatch.js`.

Es existiert ein Smoke Test mit insgesamt zwei Assertions.

Der Ablauf dieser Tests ist identisch mit dem bei der Ausführung der Acceptance Tests. Es wird jedoch keine Internetverbindung zur Fehlerfreien Ausführung benötigt.

ACCEPTANCE TESTS (UI TESTS)

Alle Unit Tests befinden sich im Ordner `/test/acceptance/`. Die Konfiguration befindet sich in `/build/tasks/nightwatch.js`.

Es existieren zwei Acceptance Tests mit insgesamt 45 Assertions.

Die Aufgabe dieser Tests kommt von allen am nächsten an den geforderten Blackbox Testplan. Der Hauptfokus der Tests liegt auf dem korrekten Anzeigen/ Ausblenden von Controls und dem Ausgeben der jeweils korrekten Daten (Berufe, Klassen und Stundenplan).

Beim Ausführen dieser Tests wird ein lokaler Webserver gestartet, ein frischer Build der Source Codes erstellt und die Website im Browser aufgerufen. Daraufhin wird eine festgelegte Abfolge von Benutzerinteraktionen mit der Weboberfläche ausgeführt, z.B. wird ein Vorgegebener Beruf und eine Klasse ausgewählt und überprüft ob die Angezeigten Stundendaten korrekt sind (Internetverbindung wird benötigt).

VERWENDETE TECHNOLOGIEN

- [nodejs](#): Voraussetzung für Grunt
- [gruntjs](#): Ausführung von JavaScript Tasks
- [grunt-nightwatch](#) (Grunt Task): Ausführen von automatisierten UI Tests
- [grunt-contrib-qunit](#) (Grunt Task): Ausführen von Unit Tests

¹ Eine Behauptung oder Überprüfung im Kontext eines Unit Tests