Good to know Workflow 2.0 Getting started



| THEMA | | ABTEILUNG (EN) | |
|-----------------------------------|----------------------|----------------|---------|
| Projektstart und wichtige Befehle | | Frontend | |
| THEOD | äarden museen a muse | , | HEROTON |

| AUTOR | ÄNDERUNGSDATUM | VERSION |
|-----------------|----------------|---------|
| Philip Morkisch | 07.04.2014 | 1.3 |

GOOD TO KNOW

In diesem Guide wird der generelle Start eines neuen Frontend-Projekts im Workflow 2.0, als auch die dazugehörigen Kommandozeilen-Befehle beschrieben.

Start eines neuen Projekts

- Alle Files/Ordner außer .git/ und die PDFs in den Ordner des neuen Projekts kopieren
- Die package.json öffnen
 - o Den Value "name" anpassen (sollte wie das Repo benannt werden)
 - o Den Value "system" anpassen in das gewünschte System
 - o Speichern
- Die Konsole öffnen
 - o *npm install* eingeben, um Dependencies zu installieren
 - o anschließend grunt project:init eingeben und los geht's
 - Die im Browser geöffnete URL kann von jedem Device geöffnet werden, das im selben WLAN angemeldet ist

Die Grunt commands

- grunt
 - o führt einen Build durch
- grunt project:init
 - o erstellt ein neues Projekt samt Struktur und Boilerplate, basierend auf dem definierten System in der package.json
 - o führt außerdem einen intialen Grunt-Build durch, startet den Socket-Server und öffnet die _modules.html im Default-Browser
- grunt project:sync
 - o führt einen Build nach jeder Änderung an CSS, JS, HTML, IMG Files durch, lädt alle geöffneten Browser neu und injected die geänderten Assets
- grunt code:compress
 - o Minified CSS und JS Files muss am Ende der Template-Entwicklung durchgeführt werden
- grunt code:validate
 - o validiert das JS und CSS und gibt Fehler/Änderungsvorschläge in einem Log (JS) bzw. in der Konsole aus (CSS).
 - Die Vorschläge von CSSLint sollten nach Möglichkeit umgesetzt werden
 - Die Errors im JSLint Log sollten behoben werden
- grunt images:compress
 - o Komprimiert SVGs und rekomprimiert JPG, PNG und GIF Files (sofern diese nicht für Web gepeichert wurden)
- grunt images:sprite
 - o Erstellt ein Sprite, sowie ein Sass-Partial mit Mixins auf Basis der Icons im icons-Ordner

Good to know Workflow 2.0 Getting started



- grunt watch

o führt einen Build nach Dateiänderungen, ähnlich wie **project:sync** durch, nur das kein Socket-Server gestartet und der Browser nicht neu geladen wird

grunt build:installModules

o installiert die in der bower.json vermerkten Dependecies, legt die Assets in die Zielordner und integriert etwaige Sass-Partials, HTML- und JS-Snippets in die main.scss, global.js bzw. _modules.html

- grunt build:insertAssets

o integriert etwaige Sass-Partials, HTML- und JS-Snippets in die main.scss, global.js bzw. modules.html

Wichtige Hinweise

- Sollte grunt project:sync keine Builds mehr ausführen, z.B. wenn man lange Zeit nicht mehr am Platz war, in der Konsole Strg + C und "j" eingeben, anschließend grunt project:sync erneut ausführen
- Die Kombination Windows + Webstorm Terminal + Sync macht Probleme, sprich, der Sync könnte nicht richtig funktionieren oder Builds werden nicht korrekt ausgeführt. Es sollte deshalb die native Windows Konsole verwendet werden
- Da Sync, wie der Name impliziert, syncen soll, also alle Browser/Tabs/Devices, die die von Browser-Sync erzeugte URL geöffnet haben, synchronisiert, ist das gleichzeitige Öffnen verschiedener Files, solange der Server läuft, nicht möglich.

Changelog

- V1.1
 - o Befehle geändert
 - o Wichtige Hinweise angepasst
- V1.2
 - o Befehle und Wichtige Hinweise erweitert
- v1.3
 - o Befehle aktualisiert
 - o für bessere Übersichtlichkeit gesorgt
 - o Typos korrigiert