

THEMA		ABTEILUNG (EN)	
Projektstart und wichtige Befehle		Frontend	
AUTOR	ÄNDERUNGSDATUM	VERSION	
Philip Morkisch	10.03.2014	1.2	
GOOD TO KNOW			
<p>In diesem Guide wird der generelle Start eines neuen Frontend-Projekts im Workflow 2.0, als auch die dazugehörigen wichtigsten Kommandozeilen-Befehle beschrieben.</p> <ul style="list-style-type: none"><li>▪ <b>Start eines neuen Projekts</b><ul style="list-style-type: none"><li>- Alle Files/Ordner außer <code>.git/</code> und die PDFs in den Ordner des neuen Projekts kopieren</li><li>- Die <code>package.json</code> öffnen<ul style="list-style-type: none"><li>o Den Value <b>"name"</b> anpassen (sollte wie das Repo benannt werden)</li><li>o Den Value <b>"system"</b> anpassen in das gewünschte System</li><li>o Speichern</li></ul></li><li>- Die Konsole öffnen<ul style="list-style-type: none"><li>o <b><code>npm install</code></b> eingeben</li><li>o anschließend <b><code>grunt project:init</code></b> eingeben</li><li>o einen initialen Build durch die Eingabe von <b><code>grunt</code></b> ausführen</li><li>o zum Schluss <b><code>grunt project:sync</code></b> eingeben und losentwickeln<ul style="list-style-type: none"><li>▪ Die durch <b><code>grunt project:sync</code></b> erzeugte URL kann von jedem Device geöffnet werden, das im selben Netz angemeldet ist</li></ul></li></ul></li></ul></li><li>▪ <b>Die wichtigsten Grunt commands</b><ul style="list-style-type: none"><li>- <b><code>grunt</code></b> - führt einen Build durch</li><li>- <b><code>grunt project:sync</code></b> - führt einen Build nach jeder Änderung an CSS, JS, HTML, IMG Files durch, lädt alle geöffneten Browser neu und injected die geänderten Assets</li><li>- <b><code>grunt project:init</code></b> - erstellt ein neues Projekt samt Struktur und Boilerplate, basierend auf dem definierten System in der <code>package.json</code></li><li>- <b><code>grunt code:compress</code></b> - Minified CSS und JS Files - muss am Ende der Template-Entwicklung durchgeführt werden</li><li>- <b><code>grunt images:compress</code></b> - Komprimiert SVGs und rekomprimiert JPG, PNG und GIF Files</li><li>- <b><code>grunt images:sprite</code></b> - Erstellt ein Sprite, sowie ein Sass-Partial mit Mixins auf Basis der Icons im icons-Ordner</li><li>- <b><code>grunt code:validate</code></b> - validiert das JS und CSS und gibt Fehler/Änderungsvorschläge in einem Log (JS) bzw. in der Konsole aus (CSS).<ul style="list-style-type: none"><li>o Die Vorschläge von CSSLint sollten nach Möglichkeit umgesetzt werden</li><li>o Die Errors im JSLint Log sollten behoben werden</li></ul></li><li>- <b><code>grunt watch</code></b> - führt einen Build nach Dateiänderungen, ähnlich wie <b><code>project:sync</code></b> durch, nur das kein Socket-Server gestartet und der Browser nicht neu geladen wird</li></ul></li><li>▪ <b>Wichtige Hinweise</b></li></ul>			

- 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 unter gleicher URL nicht möglich.
- Der Befehl grunt code:compress muss leider bis auf Weiteres manuell konfiguriert werden, wie genau das funktioniert, ist am Beispiel der main.js im Uglify-Task (Gruntfile.js) zu sehen oder bei Philip/Simon zu erfragen
- Gleiches gilt für den Concat-Task