|  |
| --- |
| QuakeWatch Austria |
| Vorbereitung der Entwicklungsumgebung |
|  |

|  |
| --- |
| Team Hamberger, Kocsis, Kelc  19.06.2016 |

Inhaltsverzeichnis

[1. Einleitung 2](#_Toc455131687)

[1.1. Verwendete OS Versionen. 2](#_Toc455131688)

[2. Frameworks 2](#_Toc455131689)

[2.1. Node.js 2](#_Toc455131690)

[2.2. npm 3](#_Toc455131691)

[2.3. Git 3](#_Toc455131692)

[2.4. Cordova 3](#_Toc455131693)

[2.5. Ionic 3](#_Toc455131694)

[2.6. Gulp 3](#_Toc455131695)

[2.7. Bower 4](#_Toc455131696)

[3. Setup des Projekts 4](#_Toc455131697)

[4. Testen des Projekts 4](#_Toc455131698)

[4.1. Im Browser 4](#_Toc455131699)

[4.2. Am Endgerät 5](#_Toc455131700)

[4.2.1. iOS 5](#_Toc455131701)

[4.2.2. Android 5](#_Toc455131702)

## Einleitung

Um an der „QuakeWatch Austria“- App weiterentwickeln zu können, muss eine entsprechende Umgebung vorhanden sein. Da sich diese aus mehreren Einzelkomponenten zusammensetzt, deren Installation sich nicht immer ganz einfach gestaltet, finden Sie im Folgenden eine Anleitung um ihr System entsprechend vorzubereiten.

### Verwendete OS Versionen.

Die folgende Anleitung wurde mit diesen Betriebssystemen ausführlich getestet. Sie verliert dadurch nicht ihre Gültigkeit auf anderen Systemen, allerdings können sich einzelne Schritte unterscheiden.

Ubuntu 14.04.4 LTS

Windows 7, 8/8.1

## Frameworks

### Node.js

Node.js ist das Rückgrat der App. Dieses JavaScript Framework ermöglicht die Entwicklung von skalierbaren Netzwerkanwendungen.

**Ubuntu**Die Installation unter Linux Systeme erfolgt über den Paketmanager.

apt-get install nodejs nodejs-legacy libc-ares2 libv8-3.14.5

Der Befehl installiert die folgenden drei Pakete:

* Nodejs - Das package für das NodeJS Framework selbst
* nodejs-legacy – Ein Workaround für Namespace Probleme auf Linux Systemen
* libc-ares2 - C-Bibliothek, benötigt von NodeJS um DNS-Anfrage und Namensauflösungen asynchron durchführen zu können.
* libv8-3.14.5 - High-performance JavaScript Engine

**Windows**

Download: <https://nodejs.org/en/download/current/>

Auf Windows Systeme erfolgt die Installation über einen graphischen Installer.

Dieser wird von der offiziellen Seite heruntergeladen. Zum Zeitpunkt an dem diese Anleitung verfasst wurde, ist die aktuellste Version v6.2.2.

Bei der Installation muss beachtet werden, die Option „Add to PATH“ auswählt. Diese erlaubt überall im System auf die installierten Komponenten mittels der Konsole zuzugreifen.

### npm

npm ist ein Packet- Manager für das NodeJS Framework.

Es ermöglicht, verwendeten Code einfach up-to-date zu halten, indem es Code-Snippets als Pakete betrachtet, die mit verschiedenen Metadaten versehen werden, um sie so schneller verwalten zu können.

**Ubuntu**

apt-get install npm

**Windows**

Für Windows Systeme sind hier keine weiteren Schritte erforderlich, da npm bereits im Node.js Setup enthalten ist.

### Git

Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien.

Sie wird hier nicht direkt genutzt, sondern ist wird unter anderem von Bower verwendet um entsprechend Pakete laden zu können.

**Ubuntu**

apt-get install git

**Windows**

Downloadlink: <https://git-scm.com/download/win>

Hier wird automatisch die passende Version herausgesucht. Die Installation erfolgt über eine graphische Oberfläche. Wichtig ist, dass auch hier wieder git zur PATH Variable hinzugefügt wird.

### Cordova

Apache Cordova ist ein open-source Framework für die Entwicklung auf mobilen Geräten.

Der Entwickler kann, wie bei einer Webapplikation HTML5, CSS3, and JavaScript für die cross-platform Entwicklung nutzen.

Die Installation erfolgt über npm. Das bedeutet, dass Sie unabhängig vom verwendeten Betriebssystem immer den gleichen Befehl nutzen können.

Dafür wird folgender Befehl ausgeführt:

npm install -g cordova

### Ionic

Ionic ist ein Framework, dass dazu dient mit Hilfe von HTML5, CSS und JavaScript Apps zu entwickeln, die sich auf den jeweiligen Geräten möglichst nativ verhalten.

npm install -g ionic

### Gulp

gulp ist Tool mit der selbstgestellten Aufgabe, zeitraubende Aufgabe während der Entwicklung zu automatisieren. Im Projekt wurde es mehrfach genutzt um repetitive Arbeiten mit einem einzigen Befehl zu erledigen.

npm install -g gulp

### Bower

Ist ein weiterer Paketmanager der verwendet wurde. Logischerweise hat dieser Zugriff auf teils ganz andere Pakete, weswegen wir ihn noch zusätzlich verwenden mussten.

npm install -g bower

## Setup des Projekts

Download: <https://github.com/pkocsis-tgm/QuakeWatch>

Am einfachsten geht dies, mit dem Download des Projekts, um es anschließend in den Anfangs erstellen Ordner zu entpacken.  
Für die tatsächliche Weiterentwicklung wird dringend ein Tool zu Versionierung/Verwaltung empfohlen!

Bis dato wurde dafür GitHub verwendet. (<https://github.com/>)

Dann werden folgende Befehle ausgeführt:

npm install – Zum Laden aller im package.json definierten Dependencies

npm update – Um zu überprüfen ob für die geladenen Module Updates zur Verfügung stehen

bower install – Um alle bower Dependencies zu laden.

## Testen des Projekts

Um die App testen zu können, gibt es im Wesentlichen zwei Wege:

### Im Browser

Der Erste ist das Ausführen im Browser ihrer Wahl. Diese Methode ist unkompliziert, und geht schnell, allerdings lassen sich damit nicht immer alle realen Bedingungen simulieren. (Rechenleistung eines Handys, (k)eine Internetverbindung, etc.)

Des Weiteren ist es notwendig einen Proxy hinzuzufügen, um die App im Browser testen zu können.

Dies hängt mit der Konfiguration des ZAMG REST Interfaces zusammen, die keinen direkten Zugriff mit JavaScript erlaubt.

Dafür wurde von uns ein gulp- Task geschrieben, der Proxy lässt sich also entsprechende mit

gulp add-proxy

hinzufügen, und mit diesem Befehl auch wieder entfernen:

gulp remove-proxy

Zum Ausführen im Browser nutzen wir folgenden Befehl (im Verzeichnis des Projekts): ionic serve

Anschließend öffnet ionic selbstständig den Standrad- Browser und leitet ihn auf die Adresse *http://localhost:8100/* um. Diese lässt sich selbstverständlich auch in anderen gängigen Browser öffnen.

### Am Endgerät

### iOS

Um die App unter iOS testen zu können, wird die Entwicklungsumgebung Xcode benötigt.

Download: <https://developer.apple.com/xcode/>

Anschließend sind folgende Schritte durchzuführen:

1. Ein Terminal im Projektordner öffnen.
2. Folgende Befehle eingeben:

ionic platform add ios

gulp remove-proxy

ionic build ios

3. xCode öffnen

4. Projekt öffnen (quakeWatch)

a. Falls das Projekt noch nicht vorhanden ist

i. Open another Project -> Projekt navigieren -> platforms -> ios ÖFFNEN

ii. Wenn ein Gerät zum ersten Mal angesteckt wird muss es initiiert werden -> Fix issues

5. Play Taste in xcode drücken (Run)

### 4.2.2. Android

Um die App auf Android Geräte übertragen zu können, wird zusätzlich noch das Android SDK benötigt.

Download: <https://developer.android.com/studio/index.html#downloads>

Hier steht zur Auswahl das Android Studio (IDE) oder nur die Commandline Tools zu installieren.

Für das entwickeln wird allerdings nur das CMD Tool benötigt, nicht auch die Programmierumgebung

Weiters muss das Android Gerät den USB- Debug Modus aktiviert haben. Wie man dies durchführt unterscheidet sich von Gerät zu Gerät.

Anschließend das Gerät dann an den PC hängen und folgende Befehle ausführen:

**Windows**

ionic platform add andorid

gulp remove-proxy

ionic run android

**Linux/OS X**

./runWithoutProxy.sh anroid

./runWithoutProxy.sh anroid

## Dokumentation

Unserer App liegt, neben der Kommentare im Source Code auch eine umfangreiche Dokumentation bei. Diese ist ähnlich wie zum Beispiel die Java API, also mit Header- Kommentaren im Code erstellt.

Daraus lässt sich dann mit folgenden Befehlen eine Website generieren, um die Dokumentation übersichtlich vor sich zu haben.

gulp generate-docs

gulp serve-docs

Danach kann die Dokumentation unter <http://localhost:8200> erreicht werden.