



QWC2 Dokumentation

Inhaltsverzeichnis

1	Voraussetzungen.....	3
2	Installation von QWC2.....	3
3	Konfiguration von QWC2.....	3
3.1	Die config.json Datei.....	4
3.2	Themenkonfiguration.....	5
3.2.1	Anpassen und Publizieren der QGIS Projekte.....	6
3.2.2	Themenkonfiguration themesConfig.json.....	6
3.2.3	Generieren von themes.json.....	9
3.3	Übersetzungen.....	9
4	Serverseitige Konfiguration.....	9
4.1	Cross-Origin Anfragen.....	9
4.2	Name der Dateien beim Drucken.....	10
4.3	Dienst zum Generieren und Auflösen von kompakten Permalinks.....	10
5	Fortgeschrittene Anpassungen vom QWC2.....	11
5.1	Kompilierung von QWC2.....	12
5.2	Suchdienste.....	12
5.3	Farbschema.....	13
5.4	Projektionen.....	13
5.5	Hilfe Dialog.....	14

1 Voraussetzungen

Eine lauffähige QGIS Server Instanz wird vorausgesetzt. Projekte, die im QWC2 als Themen angeboten werden, müssen als WMS publiziert sein.

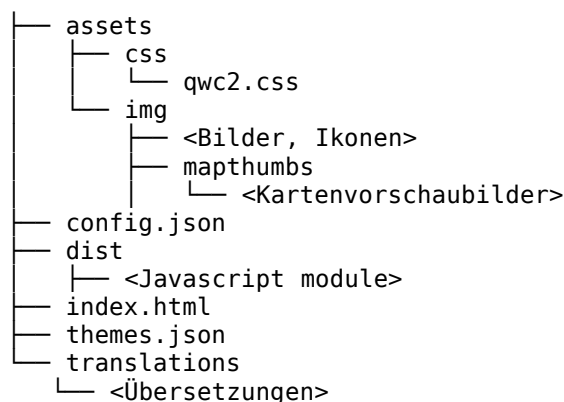
Desweiteren wird ein Webserver vorausgesetzt, welcher die QWC2 Applikation betreiben wird.

Nach Möglichkeit sollte der QGIS Server auf demselben Hostname und Port laufen wie die QWC2 Applikation, ansonsten müssen zusätzliche Massnahmen umgesetzt werden um Domänen-übergreifende Anfragen seitens QWC2 zu ermöglichen, siehe Kapitel 4.1.

2 Installation von QWC2

QWC2 ist eine Javascript Applikation. Auf produktiven Umgebungen wird Sie in kompilierter und optimierter Form („Applikationsbundle“) installiert. Die meisten Anpassungen lassen sich ohne neu-kompilieren der Applikation durchführen. Für das neu-kompilieren der Applikation muss eine Entwicklungsumgebung eingerichtet werden, dies ist in Kapitel 5 beschrieben.

Die kompilierte QWC2 Applikation hat folgende Struktur:



Um die QWC2 auf produktiven Umgebungen zu installieren, reicht es diese gesamte Struktur an einen vom Webserver zugänglichen Ort zu kopieren. Die **index.html** Datei ist der Eingangspunkt der Applikation. Falls QWC2 in einem Unterordner einer Website installiert wurde, müssen die **assetsPath** und **translationsPath** Einstellungen in der **config.json** Datei entsprechend angepasst werden.

3 Konfiguration von QWC2

Die Konfiguration von QWC2 erfolgt hauptsächlich über die folgenden zwei Dateien:

- **config.json**: Hier werden verschiedene Verhaltensaspekte von QWC2 konfiguriert, unter anderem welche Komponenten in der Desktop- bzw. Mobil-Version aktiv sind, welche Einträge in Menü und Werkzeugleiste erscheinen, wo sich Ressourcendateien befinden, usw.
- **themes.json**: Hier werden die Themen konfiguriert, die in QWC2 zur Auswahl

stehen, sowie die verfügbaren Hintergrundkarten.

Dazu kann QWC2 über folgende Dateien weiter personalisiert werden:

- `assets/css/qwc2.css`: Dieser ist ein globaler Stylesheet, und es können beliebige CSS Deklarationen hinzugefügt werden um die Darstellung von QWC2 anzupassen.
- `index.html`: Hier kann insbesondere der Titel angepasst werden, der im Browser angezeigt wird.

Im folgenden werden die Konfigurationsdateien genau beschrieben.

3.1 Die `config.json` Datei

Folgende Tabelle listet die Konfigurationsparametern in `config.json` auf:

<code>proxyUrl</code>	Die URL des Proxy-Dienstes den man verwenden kann um Cross-Origin abfragen zu erlauben, siehe Kapitel 4. Falls ein leerer String angegeben wird, wird kein Proxy-Dienst verwendet.
<code>qwc2serverUrl</code>	Die URL zum QWC2 Server, der Verwendet wird um die kompakte Permalinks der Applikation zu generieren und auflösen. Falls ein leerer String angegeben wird, werden keine kompakte Permalinks für die Share Komponente generiert und diese verwendet stattdessen die vollständige Permalinks. Siehe Kapitel 4.3.
<code>translationsPath</code>	Der Pfad vom Root-Verzeichnis der Website zum Order, in dem die Applikationsübersetzungen liegen.
<code>assetsPath</code>	Der Pfad vom Root-Verzeichnis der Website zum Ordner, in dem die Ressourcendateien liegen.
<code>urlPositionFormat</code>	Wie die aktuelle Kartenansicht in der URL kodiert wird: <ul style="list-style-type: none">• <code>extent</code>: Die Kartenansicht wird über den Ausschnitt kodiert.• <code>centerAndZoom</code>: Die Kartenansicht wird über Zentrum und Masstab kodiert.
<code>urlPositionCrs</code>	Der EPSG-Code des Koordinatensystems, in dem die Ausschnitt bzw. Zentrum Koordinaten in der URL angegeben werden. Falls ein leerer String angegeben wird, wird das Kartenkoordinatensystem verwendet. Es können nur Koordinatensysteme verwendet werden, die der Applikation bekannt sind, siehe Kapitel 5.4.
<code>plugins</code>	Die Applikation ist modular und besteht aus vielen Plugins, die individuell aktiviert oder deaktiviert werden können, insbesondere separat für mobile und desktop. Um ein Plugin zu aktivieren, muss der entsprechender Eintrag in dieser Liste vorhanden sein. Dieser Eintrag hat folgende Struktur: <pre>{ „name“: <Name des Plugins>, „cfg“: { <Pluginspezifische Konfiguration> } }</pre>

Relevante Parameter für die Pluginspezifische Konfiguration:

Map	projection	Der EPSG-Code des Kartenkoordinatensystems. Es können nur Koordinatensysteme verwendet werden, die der Applikation bekannt sind, s. Kapitel 5.4.	
	maxExtent	Der Maximale Ausschnitt, der von der gewählten Kartenprojektion unterstützt wird. Kann von http://www.spatialreference.org/ref/epsg/ abgelesen werden.	
TopBar	menuItems	Konfiguriert die Einträge, die im Applikationsmenü erscheinen.	
	toolbarItems	Konfiguriert die Einträge, die in der Werkzeugleiste erscheinen.	
	searchOptions	minScale	Minimaler Massstab fürs Zoomen auf Suchresultate.
		showProvider Selection	Ob ein Suchdienstselektionsmenü angezeigt werden soll. Falls nicht werden beim Suchen alle konfigurierten Suchdienste verwendet.
BottomBar	viewertitleUrl	Die URL zur Homepage des Betreibers der Applikation. Das Textlabel dazu kann in den Übersetzungen unter viewertitle_label angepasst werden.	
	termsUrl	Die URL zu den Nutzungsbedingungen der Applikation. Das Textlabel dazu kann in den Übersetzungen unter terms_label angepasst werden.	
ZoomIn, ZoomOut, LocateButton, LayersButton, BackgroundSwitcher	position	Die Position des jeweiligen Buttons, wobei die Positionen von unten nach oben nummeriert sind (d.h. 0 ist ganz unten).	
LayerTree	showLegendIcons	Ob die Legendenkonsolen im Ebenenbaum angezeigt werden sollen.	
Identify	params	Toleranzen für die Identify abfragen für Punkt (FI_POINT_TOLERANCE), Linie (FI_LINE_TOLERANCE) und Polygon (FI_POLYGON_TOLERANCE).	

Measure	showMeasureMode Switcher	Ob die Schaltflächen zum Umstellen des Messmodus angezeigt werden sollen. Deaktivieren ist nur sinnvoll falls in menuItems oder toolbarItems separate Einträge für die jeweiligen Modi definiert sind.
---------	--------------------------	--

Weitere Plugin-Einstellungen sollten nicht verändert werden, da sie das korrekte Verhalten der Produktspezifischen Komponenten gewährleisten.

3.2 Themenkonfiguration

Jedes Thema entspricht einem QGIS Projekt, welches via QGIS Server publiziert wird.

Die Schritte für die Themenkonfiguration sind folgende:

1. Anpassen und Publizieren der QGIS Projekte.
2. Einträge für Themen und Hintergrundkarten in themesConfig.json erstellen.
3. Generieren von themes.json mit Hilfe von themesConfig.py.

3.2.1 Anpassen und Publizieren der QGIS Projekte

Neben dem eigentlichen Karteninhalt werden weitere Eigenschaften eines Thema direkt im entsprechenden QGIS Projekt definiert. Folgende Tabelle gibt eine Übersicht:

Abfragbare Ebenen	Einstellungen → Projekteigenschaften → Layer abfragen	Steuert für welche Ebenen QWC2 Objektinformationen abfragen kann.
Geometrie in der Objektantwort beifügen	Einstellungen → Projekteigenschaften → WMS Capabilities	Dies wird sehr empfohlen, ansonsten können die abgefragte Objekte in QWC2 nicht hervorgehoben werden.
Maptip-Anzeigetext	Layereigenschaften → Anzeigen	Dieser wird in QWC2 beim Anhalten mit der Mouse über einem Objekt angezeigt, falls die Maptips aktiv sind (QWC2 Menu → Ebenen & Legende → Maptips anzeigen). Falls der Maptip-Anzeigetext einem Feld entspricht, wird er dazu für die Benennung der Resultate im Abfragedialog verwendet.
Layermetadaten	Layereigenschaften → Metadaten	Die Attributionen (falls vorhanden) der sichtbaren Ebenen werden unten rechts im QWC2 Kartenfenster angezeigt. Titel, Zusammenfassung, Schlüsselwörter und Metadaten-URL werden, soweit spezifiziert, im QWC2 Layer-Metadatendialog angezeigt.

Minimaler und maximaler Masstab	Layerereigenschaften → Allgemein → Massstabsabhängige Sichtbarkeit	Wie in QGIS steuern diese auch in QWC2 den Massstabsbereich innerhalb welchem eine Ebene sichtbar ist.
Sichtbarkeit der Ebenen	Layerfenster	Definiert den Anfangszustand der Ebenen beim laden des Themas.
Rendering-Reihenfolge der Ebenen	Layerreihenfolgefenster	Definiert die Reihenfolge in der im QWC2 die Ebenen gezeichnet werden.
Schlüsselwörter	Einstellungen → Projekteigenschaften → OWS-Server → Diensteigenschaften → Schlüsselwortliste	Die Suchfunktion in der QWC2 Themenauswahl berücksichtigt diese, um die Themen zu filtern.
Thementitel	Einstellungen → Projekteigenschaften → OWS-Server → Diensteigenschaften → Titel	Dieser wird in der QWC2 Themenauswahl angezeigt.
Druckvorlagen		Gültige Druckvorlagen (d.h. mit einem Kartenelement) werden im QWC2 Druckdialog zur Auswahl angeboten.

Unter Einstellungen → Projekteigenschaften muss schliesslich das Projekt veröffentlicht werden, zwar als WMS und WFS (WFS wird für die Abfrage von Objekten innerhalb eines Polygonbereichs verwendet).

3.2.2 Themenkonfiguration themesConfig.json

Das minimale Skelett der themesConfig.json Datei sieht wie folgt aus:

```
{
  "themes": {
    "items": [
      { <ThemaDefinition> },
      { <ThemaDefinition> },
      ...
    ], "groups": [
      {
        "title": <Titel untergruppe>,
        "items": [...],
        "groups": [...]
      },
      ...
    ]
  },
  "backgroundLayers": [
    { <HintegrundDefinition> },
    { <HintegrundDefinition> },
    ...
  ],
  "defaultScales": [<Massstabszahlen>]
}
```

Die Themen sind hierarchisch in Gruppen organisiert: jede Gruppe hat einen Titel (mit Ausnahme der obersten Gruppe), eine Liste von Themen und eine optionale Liste von Untergruppen. Entsprechend dieser Hierarchie werden die Themen in der QWC2 Themenauswahl angeordnet.

Die Struktur der Thema-Definition ist folgende:

```
{
  "url": "http://qgisserverhost/wms/theme",
  "title": "ThemeTitle",
  "thumbnail": "thumbnail.png",
  "attribution": "ThemeCopyright",
  "attributionUrl": "http://www.company.com/about",
  "default": true,
  "scales": [8000, 4000, 2000, 1000, 500],
  "printScales": [8000, 4000, 2000, 1000, 500],
  "printLabelForSearchResult": "labelid",
  "watermark": {
    "text": "Watermark Text",
    "texpadding": "1",
    "fontsize": "14",
    "fontfamily": "sans",
    "fontcolor": "#0000FF",
    "backgroundcolor": "#FFFFFF",
    "framecolor": "#000000",
    "framewidth": 1
  }
  "extent": [10, 20, 15, 25],
  "tiled": false,
  "format": "image/png;mode=8bit",
  "backgroundLayers": [
    {
      "name": "Background1",
      "printLayer": "PrintLayer",
      "visibility": true
    },
    {
      ...
    }
  ],
  "searchProviders": ["provider1", "provider2", ...],
  "additionalMouseCrs": ["EPSG:1234", "EPSG:5678", ...]
}
```

url	Erforderlich	Basis-URL zum QGIS-Server WMS Dienst.
title	Optional	Personalisierter Thementitel, überschreibt Titel aus Projekt.
thumbnail	Optional	Dateiname des personalisierten Vorschaubildes, das unter assets/img/mapthumbs abgelegt werden muss. Standardmässig wird das Bild automatisch via WMS GetMap Anfrage generiert.
attribution	Optional	Attribution des Thema, wird unten rechts im Kartenbereich angezeigt.
attributionUrl	Optional	URL zum Attributionstext.

default	Optional	Definiert, ob das Thema das Standardthema ist, welches beim starten von QWC2 geladen wird, falls keins in der URL spezifiziert ist.									
scales	Optional	Liste von erlaubten Zoom-Massstäbe für dieses Thema. Falls nicht definiert wird die globale defaultScales Liste verwendet.									
printScales	Optional	Verfügbare Massstäbe beim Drucken. Falls nicht definiert, kann beim Drucken ein beliebiger Massstab eingegeben werden.									
printLabelForSearchResult	Optional	Die ID von einem ComposerLabel in dem die Beschriftung des ausgewählten Suchresultats beim Drucken geschrieben werden soll. Falls nicht definiert, wird die Beschriftung in der Karte neben dem selektierten Objekt geschrieben.									
watermark	Optional	Konfiguration des Quellenzeichens der beim Raster-Export dem Bild hinzugefügt wird.									
extent	Optional	Benutzerdefinierte Maximalausdehnung des Thema in WGS84 Gradkoordinaten. Falls nicht definiert wird sie vom WMS-Dienst ermittelt.									
tilled	Optional	Ob ein tiled-WMS verwendet werden soll. Standardmässig false.									
format	Optional	Das Bildformat welches vom WMS-Dienst angefordert wird. Standardmässig image/png.									
backgroundLayers	Optional	<div> <div>Liste von Hintergrundebenen für das Thema:</div> <table> <tr> <td>name</td><td>Erforderlich</td><td>Name der Hintergrundebene gemäss deren Definition (s. weiter unten).</td></tr> <tr> <td>printLayer</td><td>Optional</td><td>Name der im QGIS Projekt enthaltene Ebene, die beim Drucken verwendet werden soll (s. weiter unten).</td></tr> <tr> <td>visibility</td><td>Optional</td><td>Die initiale Sichtbarkeit dieser Hintergrundebene beim laden des Thema.</td></tr> </table> </div>	name	Erforderlich	Name der Hintergrundebene gemäss deren Definition (s. weiter unten).	printLayer	Optional	Name der im QGIS Projekt enthaltene Ebene, die beim Drucken verwendet werden soll (s. weiter unten).	visibility	Optional	Die initiale Sichtbarkeit dieser Hintergrundebene beim laden des Thema.
name	Erforderlich	Name der Hintergrundebene gemäss deren Definition (s. weiter unten).									
printLayer	Optional	Name der im QGIS Projekt enthaltene Ebene, die beim Drucken verwendet werden soll (s. weiter unten).									
visibility	Optional	Die initiale Sichtbarkeit dieser Hintergrundebene beim laden des Thema.									
searchProviders	Optional	Liste von Suchdienstnamen, die für das Thema aktiv sind. S. Kapitel 5.2.									
additionalMouseCrs	Optional	Zusätzliche Koordinatensysteme für die Mousekoordinatenanzeige. Die angegebenen EPSG Codes müssen dem QWC2 bekannt sein, s. Kapitel 5.4. Standardmässig werden WGS84 (EPSG:4326) und die Kartenprojektion angeboten.									

Die Struktur der Hintergrundebenen-Definition hängt vom Typ der Hintergrundebene ab. Gemeinsam ist folgende Struktur:

```
{
  "name": "Background1",
  "title": "Human readable title",
```

```

    "type": "layertype",
    "thumbnail": "thumbnail.png",
    [...]
}

```

name	Erforderlich	Name der Hintergrundebene. Dieser Name ist in backgroundLayers in der Thema-Definition zu verwenden.
title	Erforderlich	Titel der Hintergrundebene, wird im Hintergrundauswahlmenü angezeigt.
type	Erforderlich	Der Hintergrundebentyp. Z.b. wms, wmts, osm, ...
thumbnail	Optional	Dateiname des personalisierten Vorschaubildes, das in assets/img/mapthumbs abgelegt werden muss. Standardmässig wird default.jpg verwendet.

Einige Beispiele von Hintergrundebenen sind unter <https://github.com/qgis/qwc2-demo-app/blob/master/themesConfig.json> zu sehen. Typischerweise werden entweder WMS oder WMTS verwendet.

Beim Drucken werden die Hintergrundebene ausgelassen. Man kann allerdings für jede Hintergrundebene eine im QGIS Projekt eingebundene Rasterebene als passende Druckebene angeben. Somit kann ein schneller WMTS für die Darstellung in der Applikation verwendet werden und beim Drucken hingegen ein passender WMS dessen Darstellung optimal für Massstab und Auflösung ist.

3.2.3 Generieren von themes.json

Das Python-Skript themesConfig.py welches im qwc2/scripts Verzeichnis der Applikation vorliegt liest die themesConfig.json Datei ein und vervollständigt diese mit den Eigenschaften aus dem WMS-Dienst. Die Ausgabe wird in themes.json geschrieben. Diese Datei muss im Applikationsbundle hinterlegt werden (siehe Struktur in Kapitel 2).

3.3 Übersetzungen

Die Übersetzungen befinden sich im translations Ordner und sind nach dem Muster data.<locale> benannt. Diese sind einfache JSON Dateien und können mit einem Texteditor bearbeitet werden.

4 Serverseitige Konfiguration

4.1 Cross-Origin Anfragen

Falls der QGIS Server nicht auf demselben Host und Port wie QWC2 betrieben wird, muss sichergestellt werden dass die Cross-Origin Anfragen seitens QWC2 erlaubt sind. Dazu gibt es zwei Möglichkeiten:

1. Der Webserver, der QGIS Server betreibt, wird so konfigurieren, dass Access-Control-Allow-Origin Headers den Antworten des QGIS Servers hinzugefügt

werden.

2. Ein zusätzlicher Proxy-Dienst wird eingerichtet, und QWC2 wird so konfiguriert, dass sämtliche Anfragen durch diesen Proxy-Dienst gehen. Dazu muss die `proxyUrl` Einstellung in `config.json` entsprechend gesetzt werden, siehe Kapitel 3.1. Eine Beispielimplementation eines Proxy-Servers findet man auf <https://github.com/sourcepole/qwc2-server/blob/master/qwc2demo.py>. Die `proxyUrl` Einstellung für diesen Beispiel-Server würde dann so aussehen:

```
http://<hostname>/proxy?url=
```

4.2 Name der Dateien beim Drucken

Der QGIS Server liefert beim Drucken die Ausgabedateien, ohne einen Dateiname zu setzen. Ohne zusätzliche Massnahmen benutzt den Browser für den Download dieser Datei daher bloss den letzten Teil der WMS URL als Dateiname. Dies ist unschön, insbesondere weil die Dateiendung fehlt. Auch hier gibt es zwei Möglichkeiten, dies zu korrigieren:

1. Der Webserver der QGIS Server betreibt wird so konfigurieren, dass Content-Disposition Headers den Antworten des QGIS Servers hinzugefügt werden, für Apache könnte eine entsprechende Regel wie folgt aussehen:

```
SetEnvIf Request_URI "^/wms.*/(.+) $" project_name=$1
```

```
Header always setifempty Content-Disposition "attachment; filename=%  
{project_name}.pdf" "expr=%{CONTENT_TYPE} = 'application/pdf'"
```

2. Ein Proxy-Dienst wird eingerichtet, der die Content-Disposition Headers den Antworten hinzufügt. Falls im QWC2 `config.json` die `proxyUrl` Einstellung gesetzt ist, fügt QWC2 automatisch einen sinnvollen Dateiname im `filename` Query-Parameter der Druckanfrage hinzu. Dieser kann dann vom Proxy-Dienst für den Content-Disposition Header verwendet werden. Ein Beispiel ist auf <https://github.com/sourcepole/qwc2-server/blob/master/qwc2demo.py> zu sehen.

4.3 Dienst zum Generieren und Auflösen von kompakten Permalinks

Die QWC2 Permalinks können schnell relativ lang werden, vor allem wenn das Thema viele Ebenen besitzt, z.B.

```
http://<hostname>/?  
bl=mapnik&st=Bahn&e=949844%3B6003595%3B951876%3B6004324&t=test&l=a%5B29%5D%2Cc  
%5B26%5D%2Cb%5B20%5D
```

QWC2 unterstützt deshalb auch kompakte Permalinks, womit sämtliche Query-Parametern der vollständigen URL verkürzt kodiert werden, z.B.

```
http://<hostname>/?k=fa2b20925
```

Dazu muss allerdings auf Serverseite ein Dienst betrieben werden, der diese kompakte Permalinks generiert und auflöst. Dieser Dienst muss die folgende Anfragen unterstützen:

createpermalink?url=<url>	<p>url: Die vollständige QWC2 URL, für die der kompakte Permalink generiert werden soll.</p> <p>Ausgabe: Ein JSON-Dokument mit Inhalt {permalink: <permalink_url>}</p>
resolvepermalink?key=<key>	<p>key: Wert des k Parameters des kompakten Permalinks (fa2b20925 im obigen Beispiel).</p> <p>Ausgabe: Ein JSON Dokument, in dem sämtliche Key-Value Paare des vollständigen Permalinks stehen: {c: <...>, s: <...>, l: <...>, ...}</p>

Eine Beispielimplementation ist auf <https://github.com/sourcepole/qwc2-server/blob/master/qwc2demo.py> zu sehen. Für den Produktiven Einsatz sollte der Permalink-Dienst die Keys der generierten kompakten Permalinks in einer Datenbank speichern.

Sodass QWC2 den Permalink-Dienst verwendet, muss die qwc2serverUrl Einstellung im config.json entsprechend gesetzt werden.

5 Fortgeschrittene Anpassungen vom QWC2

Einige Aspekte von QWC2 lassen sich nur im Quellcode der Applikation anpassen, unter anderem:

- Änderungen an der Suchdienstlogik
- Anpassungen am Farbschema
- Hinzufügen und entfernen von eingebauten Projektionen
- Anpassung des Hilfe-Dialogs

All diese Änderungen erfordern ein neukompilieren von QWC2 ausgehend vom Quellcode. Der Quellcodeordner hat folgende Struktur:

```

— assets/           # Ressourcen
— dist/            # Zwischenspeicher, wird beim Kompilieren angelegt
— js/              # Produktspezifischer Applikationsquellcode
— prod/            # Ausgabeordner des kompilierten Applikationsbundle
— qwc2/            # Gemeinsame QWC2 Applikationskomponenten
— translations/    # Übersetzungen
— config.json      # Haupt Konfigurationsdatei
— index.html       # Eingangspunkt der Applikation
— LICENSE          # Lizenz
— package.json     # Konfiguration des Quellcodepackets
— README.md        # Entwicklerhilfe
— styleConfig.js   # Farbschema-Konfiguration
— themesConfig.json # Quelldatei für die Themenkonfiguration
— themes.json      # Generierte Themenkonfiguration
— webpack.config.js # Regeln fürs kompilieren des Applikationsbundle
— yarn.lock        # Yarn Metadaten

```

Die Ordner `assets`, `translations` sowie `config.json`, `index.html` und `themes.json` werden unverändert in den Applikationsbundle übernommen.

5.1 Kompilierung von QWC2

Um QWC2 zu kompilieren müssen `npm` und `yarn` installiert sein:

- `npm`: <https://nodejs.org/en/download/>
- `yarn`: <https://yarnpkg.com/en/docs/install>

Einmalig müssen dann sämtliche Abhängigkeiten von QWC2 installiert werden. Dazu muss in einer Node.js Kommandozeile (Start → Programme → Node.js → Node.js command prompt) ins QWC2 Quellcodeverzeichnis gewechselt werden und folgenden Befehl ausgeführt werden:

```
yarn install
```

Um QWC2 zu kompilieren muss dann in der Node.js Kommandozeile, wiederum im QWC2 Quellcodeverzeichnis, folgenden Befehl ausgeführt werden:

```
yarn run prod
```

Somit wird ein deploybares Applikationsbundle im `prod` Ordner erstellt. Dieser kann wie in Kapitel 2 installiert werden. Die Applikation kann für Entwicklungszwecke auch direkt vom Quellcodeordner mit dem Befehl

```
yarn start
```

gestartet werden. Standardmässig läuft diese dann auf <http://127.0.0.1:8081>.

Die Generierung der Themenkonfigurationsdatei `themes.json` erfolgt automatisch beim kompilieren. Sie kann aber auch separat generiert werden via

```
yarn run themesconfig
```

oder mit dem Python-Skript `themesConfig.py`, siehe Kapitel 3.2.3.

5.2 Suchdienste

Die Suchdienste sind in `js/SearchProviders.js` implementiert. Die verfügbaren Dienste werden ganz unten in der Datei exportiert, im Format

```
"<providerid>": {
  label: "<title>",
  fields: [{
    id: "<fieldid>",
    label: "<fielddtitel>",
    wildcard: [true/false]
  }, {
    ...
  },
  onSearch: <searchfunction>
},
"<id>": {
  ...
}
```

```
}
```

Der `providerid` Schlüssel identifiziert den Suchdienst und wird in `themesConfig.json` verwendet, um die jeweilige Suche für ein Thema zu aktivieren.

Falls ein Suchdienst mehrere Eingabefelder anzeigen soll, müssen diese unter `fields` angegeben werden. Dabei gibt `wildcard` an, ob vor und nach dem Eingabetext des Feldes Wildcard-Zeichen hinzugefügt werden sollen. Dies ist z.B. für Ortsnamen sinnvoll, sodass unvollständige Eingaben Resultate liefern, für Strassennummern hingegen ist dies weniger sinnvoll.

Die unter `onSearch` angegebene Funktion ist für die eigentliche Suche verantwortlich. Die API dafür ist im Kommentar im Oberen Bereich der Datei beschrieben.

5.3 Farbschema

Die Datei `styleConfig.js` enthält Variablen die das Farbschema der Applikation steuern. Die Werte der jeweiligen Farbvariablen müssen in gültiger CSS-Syntax angegeben werden.

5.4 Projektionen

Im oberen Bereich von `js/appConfig.js` können die Projektionen angegeben werden, die die Applikation unterstützen soll. Dazu muss eine Zeile im Format

```
Proj4js.defs("EPSG:XXXX", "<proj4 string>");
```

hinzugefügt werden, z.B.:

```
Proj4js.defs("EPSG:25832", "+proj=utm +zone=32 +ellps=GRS80  
+towgs84=0,0,0,0,0,0,0 +units=m +no_defs");
```

Die `proj4`-Strings können von QGIS abgelesen werden (Einstellungen → Projekteigenschaften → KBS).

5.5 Hilfe Dialog

Der Hilfe-Dialog kann via QWC2 Menu → Hilfe aufgerufen werden. Der Inhalt dazu wird von der Datei `js/Help.jsx` gesteuert, zwar durch die Funktion `renderHelp`, die im einfachsten Fall einfach einen statischen HTML Inhalt zurückgibt.