



# QWC2 Documentation

## Inhaltsverzeichnis

1	Prerequisites.....	3
2	Installation of QWC2.....	3
3	Configuration of QWC2.....	3
3.1	The config.json File.....	4
3.2	Themes configuration.....	6
3.2.1	Configuration and publication of QGIS projects.....	6
3.2.2	Themenkonfiguration themesConfig.json.....	8
3.2.3	Generieren von themes.json.....	11
3.3	Übersetzungen.....	11
4	Serverseitige Konfiguration.....	12
4.1	Cross-Origin Anfragen.....	12
4.2	Name der Dateien beim Drucken.....	12
4.3	Dienst zum Generieren und Auflösen von kompakten Permalinks.....	13
5	Fortgeschrittene Anpassungen vom QWC2.....	13
5.1	Kompilierung von QWC2.....	14
5.2	Suchdienste.....	15
5.3	Farbschema.....	15
5.4	Projektionen.....	15
5.5	Hilfe Dialog.....	16
5.6	Vorlage für den Legendendruck.....	16

# 1 Prerequisites

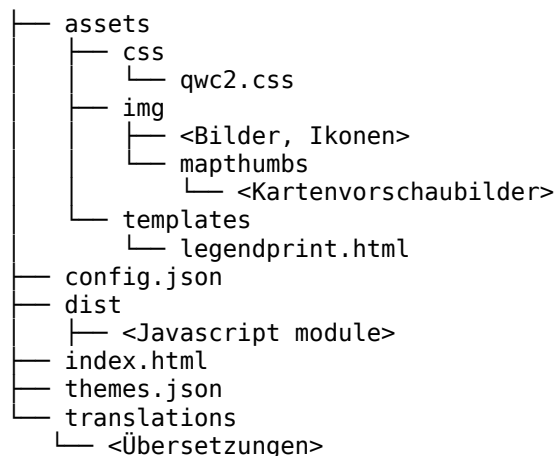
An already running QGIS server instance is a prerequisite. Projects, that are published through QWC2 must be published as WMS. In addition, a running Webserver is required for the deployment of the QWC2 applications.

Ideally, QGIS server runs on the same host name and port as the web server that publishes the QWC2 application, otherwise, additional actions have to be taken for enabling cross domain queries, see chapter 4.1.

# 2 Installation of QWC2

QWC2 is a Javascript application. In production environments, it will be installed in compiled and optimized form (application bundle). Many configuration options (such as additional themes), can be done without recompiling the application. For the compilation of the QWC2 application, a development environment has to be installed and configured. Chapter 5 explains how to do this.

The compiled QWC2 application has the following structure:



To deploy QWC2 on a production system it is enough to copy this whole structure in „prod“ to a path that is reachable on the web server. The `index.html` file is the entry point. If QWC2 is deployed in a subfolder of the website and not directly in the root, you have to change `assetsPath` and `translationsPath` in the file `config.json`.

# 3 Configuration of QWC2

The configuration happens primarily in the following two files:

- `config.json`: Here you configure various aspects about the functionality of QWC2 – which components are enabled, separate for the desktop and mobile version, which entries are included in the menu and toolbar, and where various resources can be found.
- `themes.json` (`themesConfig.json` as the source file): Here you configure the various projects that should be published in QWC2 as well as the background

maps. Themes are organized in groups and items. An item in a group corresponds to a .qgs project file published with QGIS Server. Individual items (themes) can also have different search configuration options.

In addition, QWC2 can be further personalized in the following files:

- `assets/css/qwc2.css`: This is a global style sheet and one can add additional CSS declarations that influence the look of QWC2.
- `index.html`: Here you can configure the title that is displayed in the browser window bar.

Following, the options of the above mentioned configuration files are explained:

## 3.1 The config.json File

Configuration parameters of `config.json`:

<code>proxyUrl</code>	The URL of a Proxy service that can be used to work with more than one QGIS server. Such a proxy service allows cross-origin queries, see chapter 4. If you don't want to use a proxy just use an empty string here.
<code>qwc2serverUrl</code>	The URL of a small component called QWC2 server, that is used to encode and decode compact permalink strings. If you leave this option as an empty string, the permalink component cannot create compact string and displays the full string instead. See also chapter 4.3.
<code>translationsPath</code>	The path of the root directory as viewed by the web server, where the translations can be found. It has to be changed if QWC2 isn't deployed directly to the root of the web server.
<code>assetsPath</code>	The path of the root directory as viewed by the web server, where the assets (images, CSS, etc.) can be found. It has to be changed if QWC2 isn't deployed directly to the root of the web server.
<code>urlPositionFormat</code>	Define how the current map view is encoded in the URL, either by <ul style="list-style-type: none"> <li>• <code>extent</code>: defined by map extent</li> <li>• <code>centerAndZoom</code>: defined by center and scale</li> </ul>
<code>urlPositionCrs</code>	The EPSG-Code of the coordinate system, that is used for encoding the map extent or center coordinates. If an empty string is specified, no coordinate system is used during encoding. Only coordinate systems that are known to QWC2 can be used. See chapter 5.4.
<code>preserveExtentOnThemeSwitch</code>	Controls if the map extent should be preserved on a theme switch (change to a different .qgs project) – if possible. Off by default.
<code>preserveBackgroundOnThemeSwitch</code>	Controls if the background maps should be preserved on a theme switch (change to a different .qgs project) – if possible. Off by default.

plugins	<p>The QWC2 application is modular and consists of numerous plugins, that can be individually activated or deactivated, in particular separate for the desktop and mobile environment. To enable a plugin, the respective entry has to be available:</p> <pre>{   „name“: &lt;name of the plugin&gt;,   „cfg“: { &lt;plugin-specific configuration&gt; } }</pre>
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Relevant parameters for plugin-specific configuration::

TopBar	menuItems	Configures the entries that show up in the applications menu.	
	toolbarItems	Configures the entries that show up in the toolbar on the top.	
	searchOptions	minScale	Minimal scale for the zoom-in on search results.
		showProvider Selection	Configures if a search provider selection should be displayed. If set to false, all configured search providers will be used.
	appMenuClearsTask	Configures if the opening of the application menu should close all running/open tasks (e.g. open windows and side bars). This is set to false by default.	
BottomBar	viewertitleUrl	The URL leading to the home page of the QWC2 publisher. The corresponding URL label/title can be configured in the translation files in the viewertitle_label option.	
	termsUrl	The URL leading to the terms of use of the QWC2 application, e.g. containing uptime regulations, copyright, impressum, etc. The corresponding URL label/title can be configured in the translation files in the terms_label option.	
ZoomIn, ZoomOut, LocateButton, LayersButton, BackgroundSwitcher	position	The order of the position of the respective buttons. The numbers are ranging from the bottom positions to the top (0 based).	
LayerTree	showLegendIcons	Defines if legend icons should be displayed in the layer tree.	
	showRootEntry	Defines if the root-entry of the layer tree should be shown or hidden.	

	showQueryableIcon	Defines if a small (i (info)) icon should be displayed next to the entry in a layer tree, indicating if the layer is queryable or not.
	allowMapTips	Defines if the option to show map tips should be enabled at the bottom of the layer tree component. If disabled, map tips cannot be shown.
	groupTogglesSublayers	Defines if a toggle of the checkbox of a group in the layer tree triggers toggling of all sub layers or not.
	layerInfoWindowSize	The initial size of the layer-info (attribute info) window.  Default: { "width": 480, "height": 400 }
Identify	params	Tolerances for the identify tools. Should be separately defined for desktop and mobile. Mobile typically needs higher tolerances. The tolerances are separately defined for points (FI_POINT_TOLERANCE), lines (FI_LINE_TOLERANCE) and polygons (FI_POLYGON_TOLERANCE).
Measure	showMeasureModeSwitcher	Defines if the switch buttons for the changes in measurement modes should be shown. Deactivating this is only useful if there are entries for separate measurement modes in the menuItems oder toolbarItems.

Other plugin-settings shouldn't be changed, otherwise it can't be guaranteed that the plugin-components still work as expected.

## 3.2 Themes configuration

Each theme (item) corresponds to one QGIS project (.qgs project files) that is published through QGIS server.

The steps for the theme configuration are the following:

1. Create a new QGIS project and do the necessary OWS server configuration, both on the project level and on the layer level.
2. Create entries for theme configuration, background maps and search providers in themesConfig.json.
3. Generation of the file themes.json based on the configuration of themesConfig.json by using the script themesConfig.py.

### 3.2.1 Configuration and publication of QGIS projects

The following settings effect how QGIS Server (OWS Server) behaves:

Queryable layers	Project → Project Properties → Identify layers	Defines which layers can be queried in both QGIS Desktop and QGIS Server / Web Client
Add geometry to the feature info response	Project → Project Properties → WMS Capabilities	A very recommended setting, otherwise objects cannot be  Dies wird sehr empfohlen, ansonsten können die abgefragte Objekte in QWC2 nicht hervorgehoben werden.
Maptip-Display	Layer properties → Display	This will be used by QWC2 as a map tip if map tips are enabled through QWC2 menu → Layers & Legend → Maptips. The tool tips are displayed if the mouse stays over a feature a fraction of a second. The map tip content can be a field or an expression.
Layer Metadata	Layer properties → Metadata	The data in the attribution section of enabled layers is displayed on the bottom right.  Title, Abstract, Keywords and Metadata-URLs are displayed in the layer metadata dialog.
Minimal and maximal scale  Minimaler und maximaler Masstab	Layer properties → General → Scale dependent visibility	Same as in QGIS desktop - this defines at what scale levels a layer is visible
Initial visibility state of map layers	Layer panel	The last saved state of visible layers in the QGIS project defines the initial visibility in QWC2
Rendering order of map layers	Layer order panel (if activated), otherwise the order in the layer panel	Defines the rendering order of the map layers like in QGIS desktop
Keywords	Project → Project properties → OWS-Server → Service capabilities → Keywords	Used by the filter function in QWC2 for filtering themes by keywords or names.
Theme title	Project → Project properties → OWS-Server → Service capabilities → Title	This title will be displayed above the theme thumbnails in the theme choice.
Print composer layouts		Valid print composers (with at least a map frame) that are not suppressed in Project → Project properties → OWS-Server → Exclude composers will be offered as templates in the printint section of QWC2

IMPORTANT: Under Project → Project properties → OWS-Server you have to enable the Checkbox left of „Service Capabilities“ so that QGIS server accepts this project for publication. You have to enable WMS and WFS, as WFS is used for querying features inside a polygon.

### 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>],
  "defaultPrintResolutions": [150, 300, 600],
  "defaultPrintGrid": [<siehe printGrid nachfolgende Beschreibung>]
}
```

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],
  "printResolutions": [150, 300, 600],
  "printGrid": [
    { "s": 10000, x: 1000, y: 1000 },
    { "s": 1000, x: 100, y: 100 },
    ...
  ],
  "printLabelForSearchResult": "labelid",
  "extent": [10, 20, 15, 25],
  "tiled": false,
}
```



```

"format": "image/png;mode=8bit",
"backgroundLayers": [
  {
    "name": "Background1",
    "printLayer": "PrintLayer",
    "visibility": true
  },
  {
    ...
  }
],
"searchProviders": ["provider1", "provider2", ...],
"mapCrs": "EPSG:3857",
"additionalMouseCrs": ["EPSG:1234", "EPSG:5678", ...]
"watermark": {
  "text": "Watermark Text",
  "texpadding": "1",
  "fontsize": "14",
  "fontfamily": "sans",
  "fontcolor": "#0000FF",
  "backgroundcolor": "#FFFFFF",
  "framecolor": "#000000",
  "framewidth": 1
},
"collapseLayerGroupsBelowLevel": <level>,
"skipEmptyFeatureAttributes": false
}

```

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 sowohl printScales sowie die themenunabhängigen defaultScales nicht definiert sind, kann beim Drucken ein beliebiger Massstab eingegeben werden.
printResolutions	Optional	Verfügbare Auflösungen beim Drucken. Falls sowohl printResolutions sowie die themenunabhängigen defaultResolutions nicht definiert sind, kann beim Drucken ein beliebiger Massstab eingegeben werden.

printGrid	Optional	Liste von masstabsabhängige Gitterintervalle für das Gitter in der Druckausgabe. s bedeutet jeweils der Maximalmasstab bis zu welchem die angegeben x und y Intervalle verwendet werden. Somit bedeutet das obige Beispiel, dass {x: 100, y: 100} für den Masstabsbereich $1000 \leq \text{Masstab} < 10000$ verwendet wird. Falls sowohl printGrid sowie das themenunabhängigen defaultPrintGrid nicht definiert sind, wird in der Druckausgabe kein Gitter angezeigt.									
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.									
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	<p>Liste von Hintergrundebenen für das Thema:</p> <table border="1"> <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>	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.									
mapCrs	Optional	Die zu verwendende Kartenprojektion. Falls nicht angegeben wird EPSG:3857 verwendet. Es können nur Koordinatensysteme verwendet werden, die der Applikation bekannt sind, s. Kapitel 5.4.									
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.									
watermark	Optional	Konfiguration des Quellenzeichens der beim Raster-Export dem Bild hinzugefügt wird.									
collapseLayerGroupsBelowLevel	Optional	Hierarchiestufe ab der der Ebenenbaum anfänglich zusammengeklappt angezeigt werden soll. Falls nicht angegeben, wird er komplett aufgeklappt angezeigt.									

skipEmptyFeatureAttributes	Optional	Ob leere Attribute in im Abfrageresultat nicht angezeigt werden sollen. Standardmässig werden alle Attribute angezeigt.
----------------------------	----------	-------------------------------------------------------------------------------------------------------------------------

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 Hintergrundauswahlménü 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.

Um eine neue Sprache hinzuzufügen, muss diese in translations/tsconfig.js und unter supportedLocales in js/appConfig.js eingetragen 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, der sicherstellt, dass sämtliche Antworten mit dem Access-Control-Allow-Origin versehen sind, 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 Beispielimplementierung 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:

<code>createpermalink?url=&lt;url&gt;</code>	<p>url: Die vollständige QWC2 URL, für die der kompakten Permalink generiert werden soll.</p> <p>Ausgabe: Ein JSON-Dokument mit Inhalt {permalink: &lt;permalink_url&gt;}</p>
<code>resolvepermalink?key=&lt;key&gt;</code>	<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: &lt;...&gt;, s: &lt;...&gt;, l: &lt;...&gt;, ...}</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: "<fieldtitel>",
    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` angegeben 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.

## 5.6 Vorlage für den Legendendruck

In der Titelleiste des Ebenenbaums befindet sich eine Funktion, um eine Zusammenstellung der Legenden aller sichtbaren Ebenen auszudrucken. Die Vorlage befindet sich in `assets/templates/legendprint.html` und kann beliebig angepasst werden, solange sie ein Kontainerelement mit `id="legendcontainer"` beinhaltet.