OpenSchoolMaps: Veroeffentlichung eines QGIS-Projekts als Webkarte Loesung

OpenSchoolMaps.ch — Freie Lernmaterialien zu freien Geodaten und Karten

Ein Arbeitsblatt für Selbstlerner, Studierende und Schülerinnen und Schüler der Sekundarstufe II

Einleitung

QGIS ist eine freie und quelloffene Geoinformationssystem-Software (GIS) zum Betrachten, Bearbeiten, Erfassen, Analysieren, Konvertieren und Darstellen räumlicher Daten. Um ein QGIS-Projekt mit Personen zu teilen, ohne dass diese QGIS ebenfalls installiert haben müssen, kann es als Webkarte veröffentlicht werden. In diesem Tutorial wird Schritt für Schritt aufgezeigt, wie eine Karte mit einem GPX-Track und verschiedenen Punktmarkierungen und Beschriftungen aufgebaut und anschliessend als Webseite exportiert werden kann. Dazu wird das QGIS-Plugin **qgis2web** verwendet.

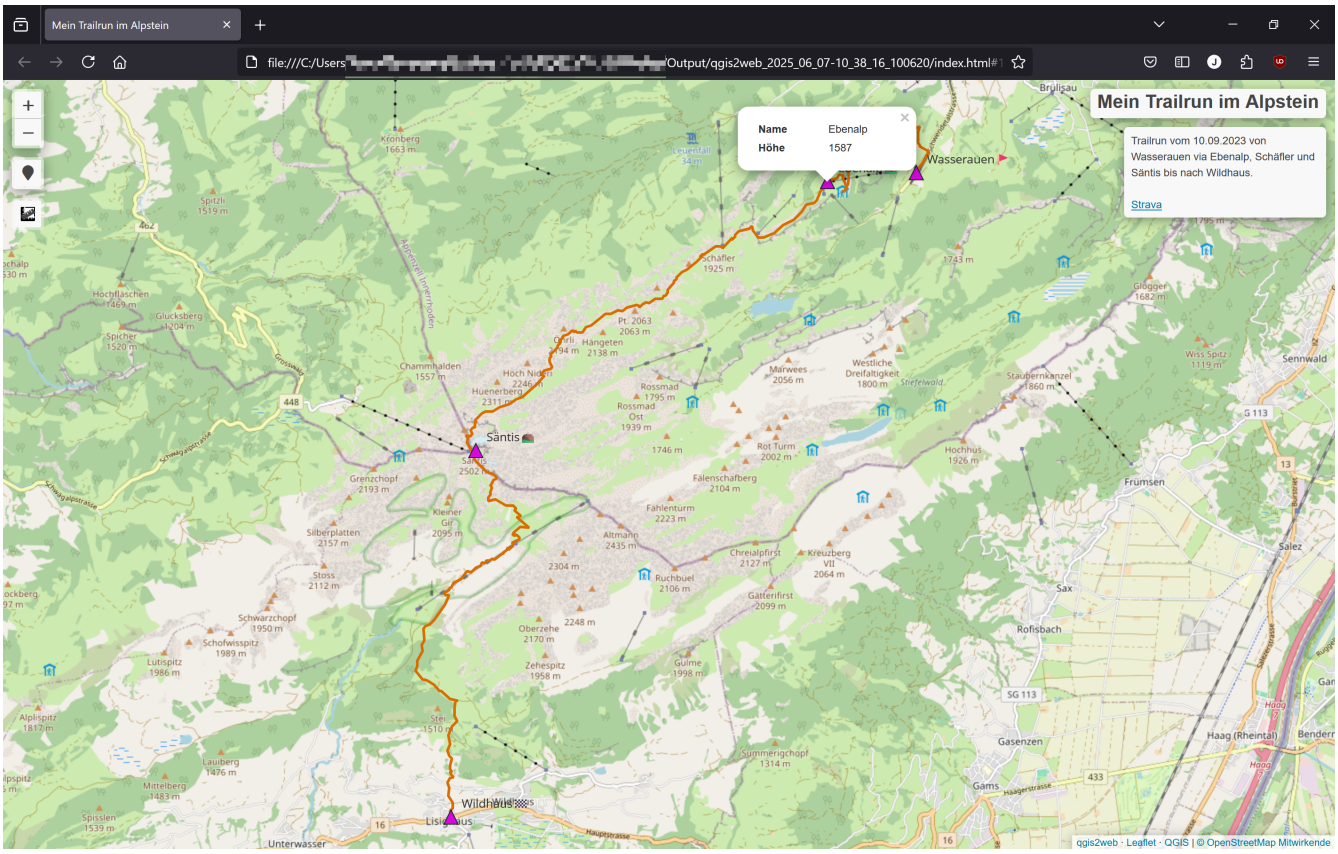

Abbildung 1. Webmap, wie sie nach der Durchführung dieses Tutorials aussehen könnte. Hintergrundkarte von OpenStreetMap.

Das wird im Arbeitsblatt gelernt:

- Einbinden und Darstellen eines GPX-Tracks und eigener Punktdaten in QGIS
- Gestalten einer übersichtlichen Karte mit Beschriftungen
- Export der Karte als interaktive Webkarte mit dem Plugin qgis2web
- Veröffentlichen der Karte als Webseite im lokalen Netzwerk oder auf GitHub Pages

Zeitaufwand: Dieses Tutorial dauert ca. 60 Minuten.

Software-/Hardware-Voraussetzungen: Als Software wird QGIS 3 vorausgesetzt. Es kann entweder die neueste herunterladbare Version oder der neueste Long Term Release (LTR) von QGIS verwendet werden. QGIS läuft auf den gängigen Betriebssystemen Windows, MacOS und Linux.

Vorbereitungen: QGIS installiert. Weitere Instruktionen siehe Abschnitt Vorbereitungen.

Eingangskompetenzen: Es werden Grundkenntnisse im Umgang mit einem Betriebssystem sowie Grundkenntnisse in QGIS 3 vorausgesetzt. Beides kann mit den frei zugänglichen OpenSchoolMaps-Tutorials nachgeholt werden.

Vorbereitungen

Installation des QGIS Plugins qgis2web: Dieses Tutorial verwendet das QGIS Plugin qgis2web. Dieses kann mit den folgenden Schritten installiert werden:

1. In der Menüleiste **Erweiterungen › Erweiterungen verwalten und installieren...** klicken
2. Im Suchfeld `qgis2web` eingeben und das gefundene Plugin anwählen
3. Unten rechts die Schaltfläche **[Erweiterung installieren]** anwählen, und bei erfolgreicher Installation das Fenster schliessen.

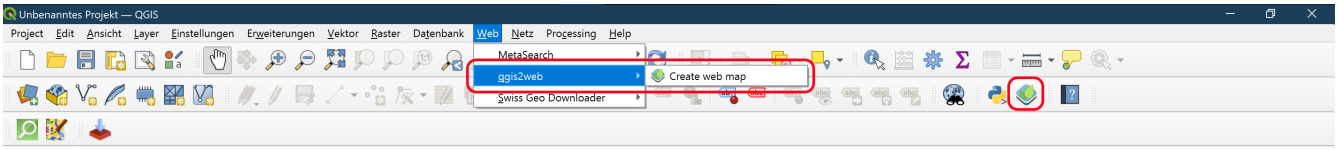

Abbildung 2. Das installierte Plugin qgis2web in der Menüleiste.

Aktualisierung der Vorschau-Komponente (Qt WebEngine)

Das Plugin qgis2web bietet ein integriertes Vorschaufenster, um die Karte direkt in QGIS zu prüfen (siehe Abbildung 2). Hierfür nutzt QGIS eine interne Browser-Komponente (Qt WebEngine). Je nach QGIS-Installation und Version kann es sein, dass diese Komponente veraltet ist oder fehlt. Das Plugin weist in diesem Fall darauf hin oder die Vorschau bleibt leer.

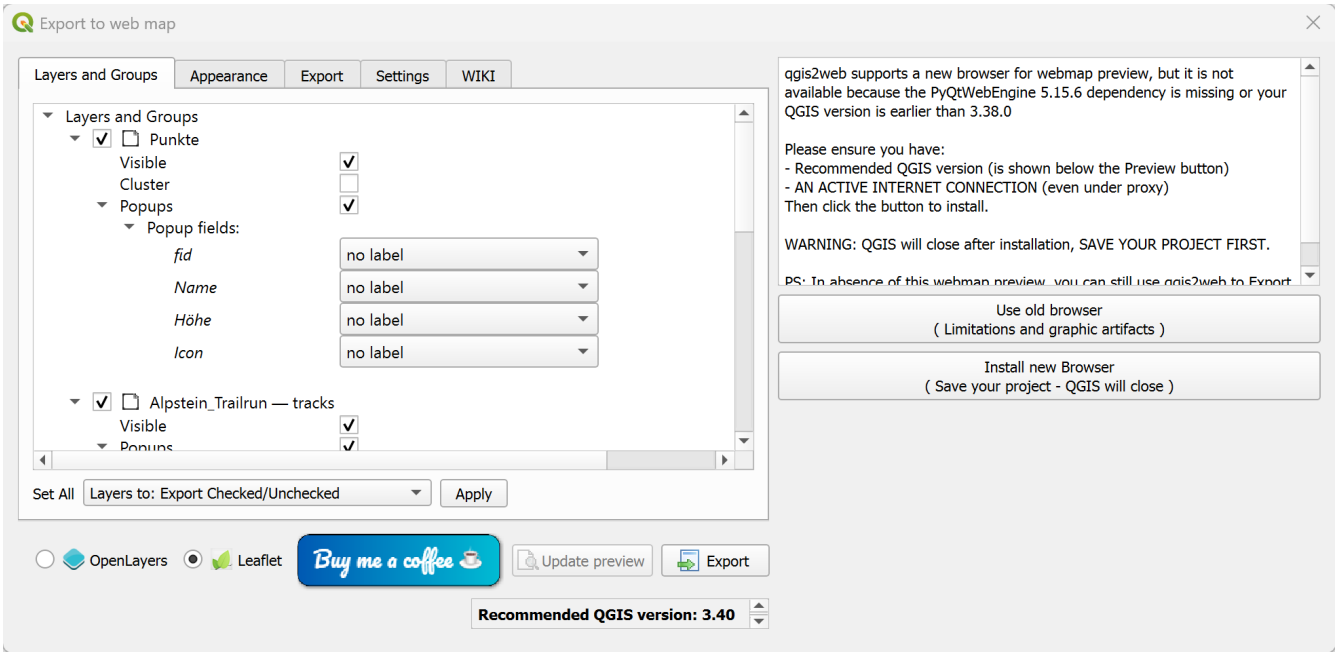

Abbildung 3. qgis2web Exportfenster mit Hinweis zur Aktualisierung der Preview-Komponente.

Dieser Schritt ist nur notwendig, wenn die Vorschau nicht korrekt angezeigt wird oder der Dialog zur Aktualisierung erscheint.

Vorgehen bei Bedarf (Update via Plugin):

1. Speichere dein QGIS-Projekt (QGIS wird durch die Aktualisierung geschlossen).
2. Klicke im Plugin-Fenster auf **[Install new Browser]** (bzw. Update). Warte bis das Update installiert wurde und QGIS geschlossen ist.
3. Öffne QGIS erneut und starte das Plugin. Prüfe mit **[Update preview]**, ob die Karte erscheint.

Falls das automatische Update nicht funktioniert (z.B. bei OSGeo4W-Installation), muss die Komponente manuell nachinstalliert werden:

1. Starte das Programm **OSGeo4W Shell** (Windows). Tippe `setup` und dann `Enter`.
2. Wähle **Fortgeschrittene Installation**, dann klick **[Weiter >]**.
3. Wähle **Aus dem Internet installieren**, dann klick **[Weiter >]**.
4. Bis zur Seite **Pakete wählen** alle Einstellungen bei den Standardeinstellungen belassen und mit **[Weiter >]** bestätigen.
5. Oben links den Suchtext `python3-pyqtwebengine` eingeben, dann beim gefundenen Paket das Feld **[Skip]** anklicken, bis eine Versionsnummer erscheint.

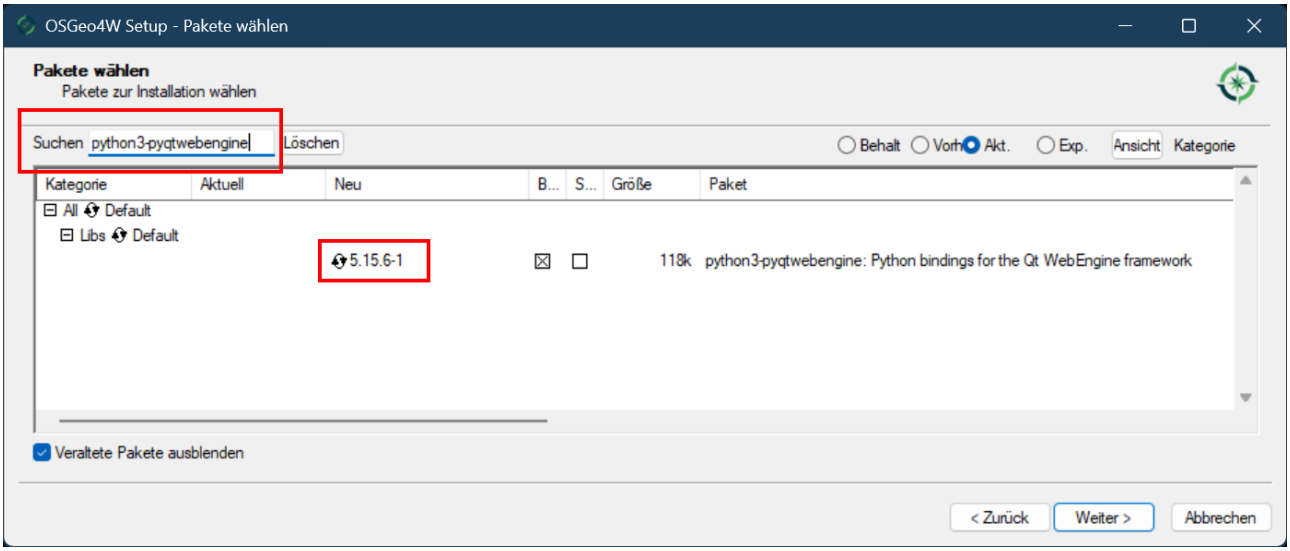

Abbildung 4. Installation von Python Qt WebEngine mittels OSGeo4W.

6. Die Installation mit **[Weiter >]** bestätigen, dann mit **[Fertig stellen]** beenden.

Veröffentlichung eines GPX-Tracks mit Annotationen

In diesem Beispiel werden ein GPX-Track sowie mehrere von Hand erfasste Punktobjekte auf der Karte dargestellt. Die meisten Schritte können aber gleich oder ähnlich auch auf andere Daten angewendet werden. Sollen statt Wanderhighlights zum Beispiel alle Sushi-Restaurants im Kanton Zürich dargestellt werden, kann nach dem Importieren der Daten genau gleich vorgegangen werden.

Daten

Überlege dir, welche Daten du für dieses Tutorial verwenden möchtest. Du kannst die im Tutorial verwendeten Beispieldaten herunterladen, einen eigenen GPX-Track verwenden, oder völlig andere Daten und Datentypen verwenden.

Karte erstellen

Starte QGIS und öffne ein neues Projekt. Wähle unten rechts das Koordinatenreferenzsystem **EPSG:3857**. Füge dann eine Hintergrundkarte (z.B. XYZ-Tiles von OpenStreetMap) hinzu.

GPX-Track importieren

Füge jetzt dein GPX-File zum Projekt hinzu. Dafür in der Menüleiste **Layer › Layer hinzufügen › Neuen GPX-Layer hinzufügen...** anwählen, und im neuen Fenster mit **[...]** die richtige Datei auswählen. Beim Objekttyp **Spuren** anwählen, und den Layer mit **[Hinzufügen]** erstellen.

Alternativ kannst du die Datei auch per Drag-and-Drop ins QGIS ziehen, **tracks** anwählen und mit **[Layer hinzufügen]** bestätigen.

Falls du mehrere Tracks gleichzeitig importieren, oder deinen Track glätten und vereinfachen

möchtest, findest du mehr Informationen im Tutorial Eine Karte deiner Reise mit QGIS erstellen.

Punktdaten erstellen

Um Punktdaten im GeoJSON-Format zu erstellen, verwenden wir die freie Webapplikation geojson.io.

1. Klicke auf das Punktsymbol (Marker) rechts auf der Karte.
2. Setze einen Punkt an der gewünschten Stelle auf die Karte.
3. Klicke den Punkt an, um dessen Eigenschaften (Properties) zu bearbeiten.
4. In der Tabelle rechts kannst du neue Spalten hinzufügen, z.B. `Name`, `Höhe` oder `Icon`.

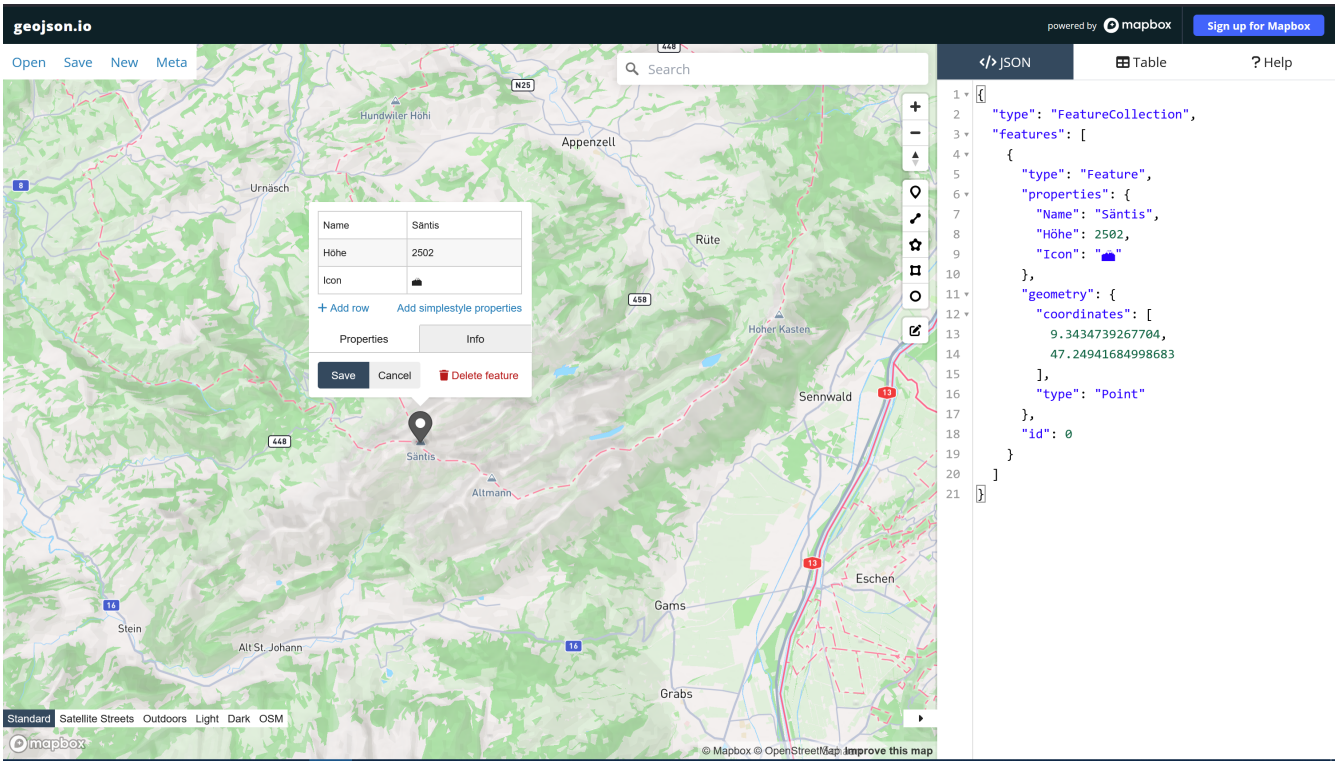

Abbildung 5. Eine Punktmarkierung mit den Eigenschaften "Name", "Höhe" und "Icon", erstellt in der Webapplikation geojson.io.

Emojis als Icons einfügen: Um Symbole wie 🚩, 🏔️ oder 📷 in die Tabelle einzufügen, kannst du die Emoji-Tastatur deines Betriebssystems nutzen:

- **Windows:** Drücke die Tasten `Windows` + `.` (Punkt).
- **Mac:** Drücke die Tasten `Cmd` + `Ctrl` + `Leertaste`.

Wenn jetzt weitere Punkte hinzugefügt werden, ist es einfacher, die Eigenschaften direkt rechts in der Tabelle zu ergänzen.

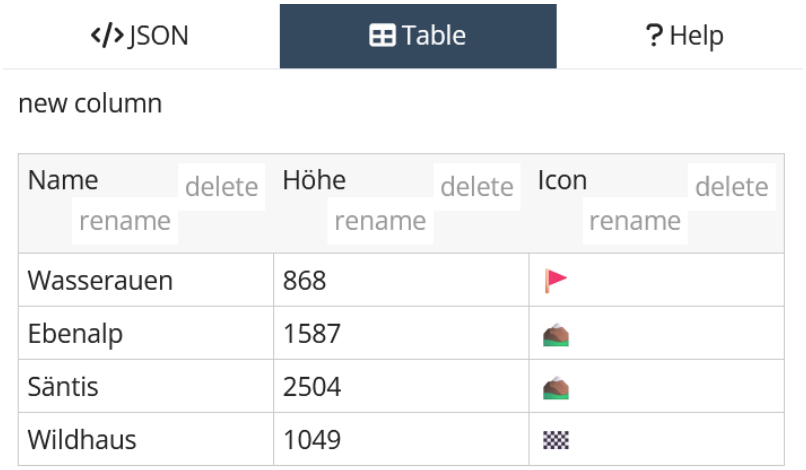

Abbildung 6. Mehrere Punkte direkt in der Tabelle eingetragen.

Wenn alle Punkte erfasst worden sind, können die Punkte oben links mit **[Save]** → **[GeoJSON]** unter dem Namen `Punkte.geojson` abgespeichert werden.

Um die erstellten Punkte in QGIS zu importieren, kann die GeoJSON-Datei per Drag-and-Drop in das Projekt gezogen werden. Alternativ kann wieder über **Layer › Layer hinzufügen › Vektor Layer hinzufügen...** ein Importfenster geöffnet werden, und die Datei über **[...]** ausgewählt werden. Danach alle Standardeinstellungen belassen und mit **[Hinzufügen]** bestätigen.

Karte gestalten

Aktuell sieht die Karte noch nicht sehr ansprechend aus. Die Farben des GPX-Tracks und der Punkte sind von QGIS zufällig gewählt worden und je nach Farbe schlecht sichtbar. Das kannst du jetzt ändern.

Nicht alle Einstellungen, die in QGIS gemacht werden können, werden beim Exportieren mittels qgis2web übernommen. Bevor du also zu viel Zeit damit verlierst, einen Effekt perfekt zu konfigurieren, erstelle zuerst einen Test-Export mit qgis2web (siehe Abschnitt Karte exportieren), um zu prüfen ob es überhaupt möglich ist. Wirf dazu unbedingt auch einen Blick in den Anhang Best Practices für qgis2web!

Titel und Beschreibung

Deinem QGIS-Projekt kann ein Titel vergeben werden, der anschliessend im Browser als Tab-Name angezeigt wird. Der Titel und eine Beschreibung können zusätzlich auch auf der exportierten Karte dargestellt werden. Um den Titel hinzuzufügen, klicke oben links auf **Projekt › Eigenschaften** und füge im Abschnitt **Allgemein** deinen Titel unter **Projekttitel** hinzu. Die Beschreibung kann im Abschnitt **Metadaten** unter **Zusammenfassung** hinzugefügt werden. Dabei kann auch HTML für die Formatierung verwendet werden.

GPX-Track gestalten

Um den GPX-Track zu gestalten, kann der entsprechende Layer angewählt und das Gestaltungsfenster geöffnet werden (Pinselsymbol oberhalb der Layerliste). Dann kann die Farbe,

Linienstärke, Deckkraft, Effekte, usw. eingestellt werden. Experimentiere mit verschiedenen Einstellungen, um eine Darstellung zu finden, die dir gefällt.

Punkte gestalten

Um die Punkte zu gestalten, kann ebenfalls das Layergestaltungsfenster geöffnet werden. Sollen alle Punkte gleich dargestellt werden, kann gleich vorgegangen werden wie beim Track – es kann Farbe, Grösse, Umrandung, Effekt, Symbol und vieles mehr eingestellt werden.

Zusätzlich zum Design der Punkte können sie auch mit einem Label beschriftet werden. Dazu im Layergestaltungsfenster vom Tab **Symbolisierung** in den Tab **Beschriftung** wechseln. Dort von **Keine Beschriftung** auf **Einzelne Beschriftungen** umstellen. Beim Wert kann dann angegeben werden, was angezeigt werden soll, also zum Beispiel `Name` um den Namen des Punkts anzuzeigen, `Name || Icon` um neben dem Namen auch das Emoji anzuzeigen, oder `Name || ' - ' || Höhe || 'm.ü.M.'` um die Höhe mit Masseinheit neben dem Namen zu haben. Zusätzlich können die Schriftart, -grösse und -farbe, sowie viele weitere Einstellungen gewählt werden.

Karte exportieren

Um die Karte zu exportieren, kann qgis2web gestartet werden (siehe Abb. 2). Im geöffneten Fenster können verschiedene Einstellungen angepasst werden. Nachdem alle Einstellungen gemacht wurden, kann die Webmap unten rechts mit **[Export]** erstellt werden.

OpenLayers vs. Leaflet

OpenLayers und Leaflet sind JavaScript-Bibliotheken, die es erlauben, interaktive Karten im Browser anzuzeigen. Beim Exportieren aus QGIS kann gewählt werden, welche dieser Bibliotheken verwendet werden soll. Grundsätzlich ist keine der beiden Bibliotheken besser als die andere, aber sie unterstützen zum Teil leicht andere Funktionen. Wenn also eine Konfiguration nicht korrekt dargestellt wird (zum Beispiel ein Puffer um den Labeltext in Leaflet), prüfe, ob das bei der anderen Bibliothek funktioniert. Wenn beide Bibliotheken funktionieren, kann selbst entschieden werden. In diesem Tutorial wird Leaflet verwendet, da es etwas populärer ist als OpenLayers.

Einstellungen

Die Einstellungen im Fenster sind in drei Abschnitte unterteilt. Zusätzlich gibt es einen Abschnitt **Settings**, der die Einstellungen für das Vorschaufenster definiert, und einen Abschnitt **WIKI**, der ausführliche Beschreibungen des qgis2web Plugins auf Englisch enthält.

Layers and Groups

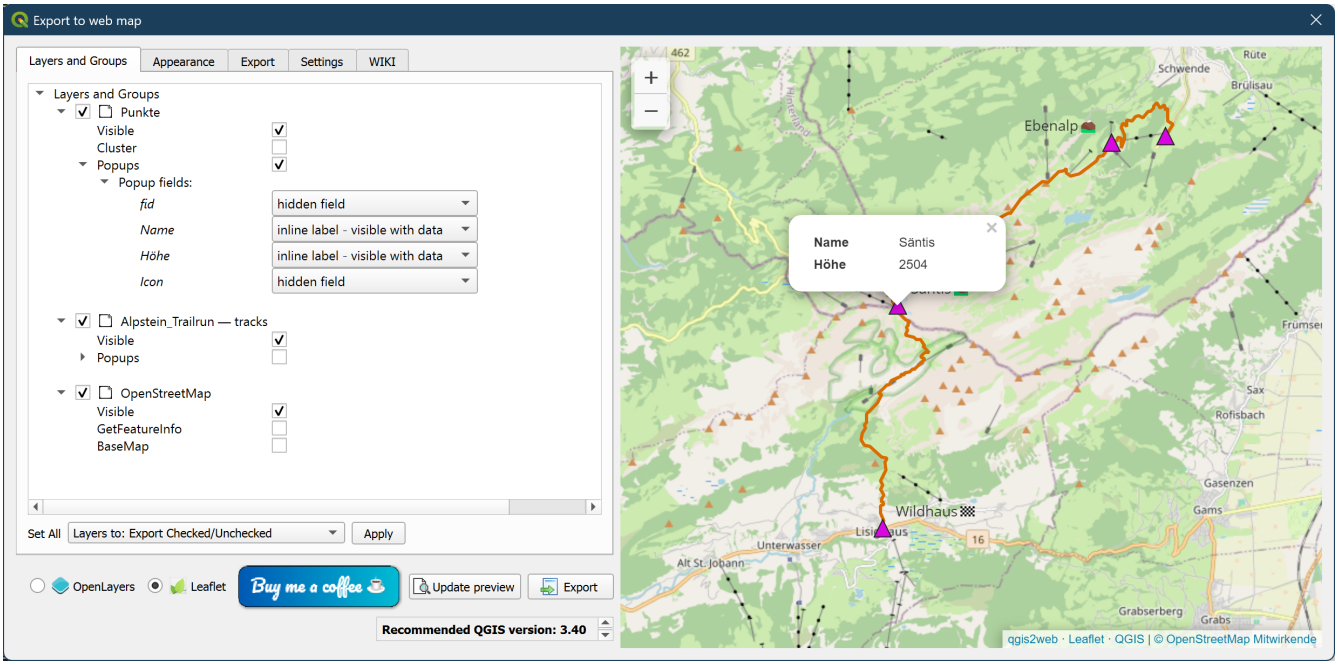

Abbildung 7. Einstellungen im Tab "Layers and Groups" des Exportfensters von qgis2web.

In diesem Abschnitt kann eingestellt werden, welche Daten aus den Layers angezeigt werden. In unserem Beispiel sollen alle Layer sichtbar (**Visible**) sein. Die Punkte sollen zusätzlich ein Popup haben (siehe Abbildung 7). Dabei wollen wir die ID und das Icon nicht anzeigen und setzen die Felder auf **hidden field**. Für den Namen und die Höhe wollen wir davor ein Label und wählen **inline label - visible with data**. Wenn die Daten leer sind, wird kein Label angezeigt. Experimentiere mit verschiedenen Einstellungen und prüfe das jeweilige Resultat mit **[Update preview]**.

Appearance

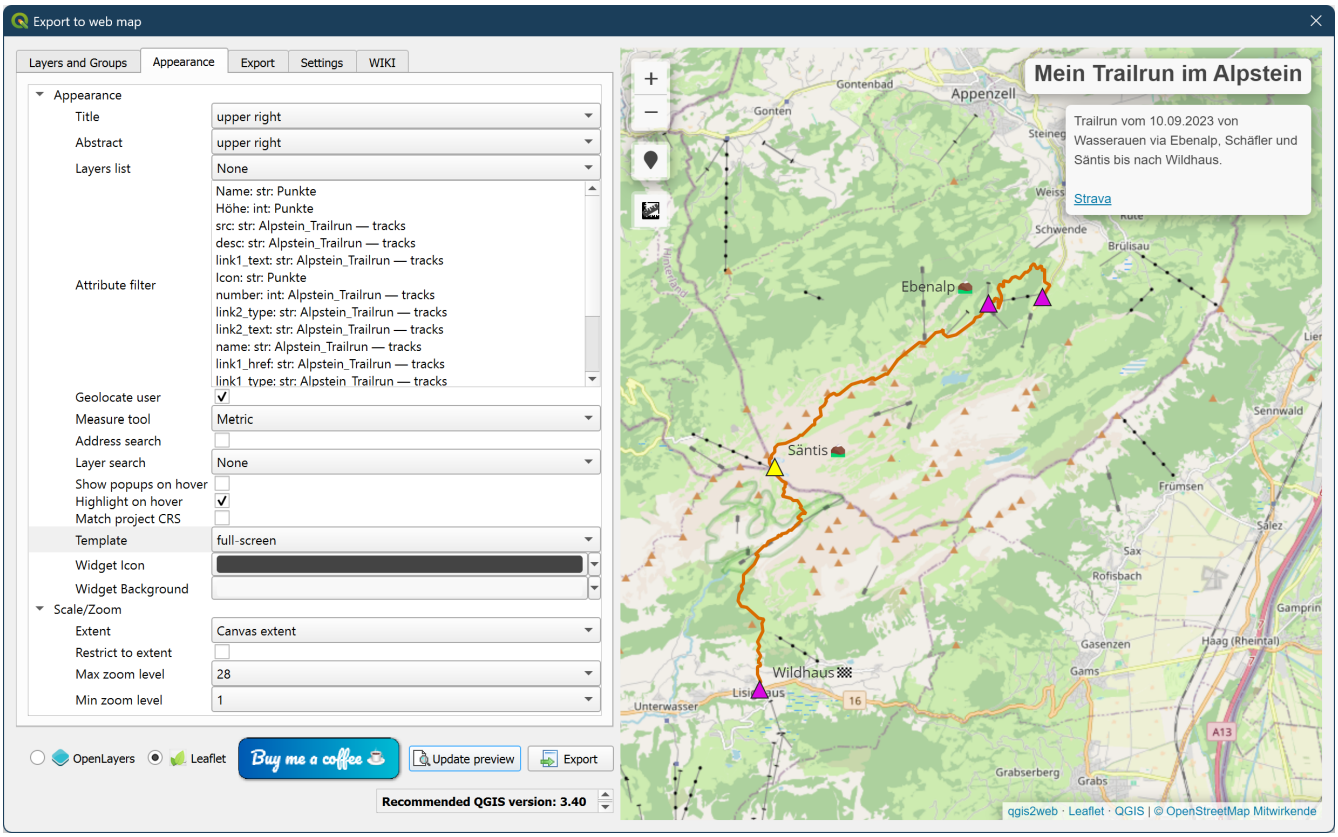

Abbildung 8. Einstellungen im Tab "Appearance" des Exportfensters von qgis2web.

In diesem Abschnitt kann eingestellt werden, welche Funktionen auf der Webmap dargestellt werden sollen. Es kann beispielsweise der Titel und die Beschreibung (**Abstract**) angezeigt werden, oder ein Messwerkzeug (**[Measure tool]** → **Metric**) eingeblendet werden. Mit **Geolocate user** kann der Benutzer seine eigene, aktuelle Position auf der Karte anzeigen (dies funktioniert erst in der exportierten Karte). Auch hier kannst du mit verschiedenen Einstellungen experimentieren, bis du mit deiner Webmap zufrieden bist.

Export

In diesem Abschnitt kann der Ordner für den Export festgelegt werden. Dafür neben **Export to folder** auf **[...]** klicken und das gewünschte Verzeichnis auswählen. Die restlichen Einstellungen können so belassen werden.

Resultat prüfen

Klicke auf **[Export]**. Deine generierte Webmap sollte sich im Browser (Chrome, Edge, Firefox, Safari, etc.) öffnen. Untersuche deine Karte, ob alles so aussieht und funktioniert, wie du dir gedacht hast. Wenn nicht, ändere die Einstellungen des QGIS-Projekts oder des qgis2web Plugins und exportiere deine Karte erneut.

Integration in eigene Webseite

Wenn du Änderungen an der Darstellung der Webmap machen möchtest, kannst du die generierten Dateien von Hand bearbeiten. So kannst du beispielsweise einen Link zu einer anderen Webseite einfügen. Das Problem bei dieser Herangehensweise ist, dass bei einer Änderung im QGIS Projekt die gesamten Webseiten-Daten erneut generiert werden. Die manuellen Änderungen müssen dann jedes Mal erneut gemacht werden.

Stattdessen kann die neu erstellte Karte auch als `iframe` in eine separate Webseite eingebunden werden. Benenne dazu die generierte Datei `index.html` in `map.html` um. Erstelle anschliessend eine neue Datei `index.html` im selben Ordner.

Es ist eine nützliche Konvention, dass die Hauptseite `index.html` genannt wird. Wenn du mit dem Browser eine Webseite öffnest, wird standardmässig `index.html` geöffnet. Beispiel: openschoolmaps.ch und openschoolmaps.ch/index.html sind die gleiche Webseite.

Füge der neuen Datei `index.html` folgenden Inhalt hinzu:

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Meine Karte</title>
</head>
<body>
    <header>
```

```
        <h1>Meine Karte</h1>
    </header>
    <main>
        <iframe src="map.html" width="100%" style="height:80vh;border:none;"></iframe>
    </main>
    <footer>
        <a href="https://www.example.com">Meine Webseite</a>
    </footer>
</body>
</html>
```

Mit etwas HTML und CSS kannst du jetzt deine eigene Seite entwerfen und deine Karte einbinden. Tutorials dazu findest du im Internet, beispielsweise SELFHTML oder W3Schools (Englisch).

Hosting

Du hast jetzt eine Webmap, die du im Browser anschauen kannst. Aber um die Karte jemand anderem zur Verfügung zu stellen, wollen wir nicht nur eine Zip-Datei mit den Daten versenden, sondern die Webseite von anderen Geräten zugänglich machen. Dazu muss die Webseite auf einem Server gehostet werden.

Hosting im lokalen Netzwerk

Um die Webseite in deinem lokalen Netzwerk (z.B. im WLAN bei dir zu Hause) zu hosten, kannst du deinen eigenen Computer verwenden. Dafür gibt es viele verschiedene Möglichkeiten. Für die hier beschriebene Variante muss Python installiert sein.

1. Öffne eine Kommandozeile.
2. Wechsle in das Verzeichnis mit deiner Webseite (`cd C:\Users...`).
3. Führe den Befehl `ipconfig` (Linux `ip a`) aus, um deine IP-Adresse zu finden (beginnt in den meisten Fällen mit `192.168.`)
4. Starte den Server mit `python -m http.server`.
5. Gib in deinem Browser `localhost:8000` ein, um zu prüfen, ob der Webserver korrekt gestartet hat.
6. Gib in deinem Browser deine gefundene IP-Adresse gefolgt von `:8000` (also bspw. `192.168.1.13:8000`) ein, um zu prüfen, dass die IP-Adresse korrekt ist.
7. Wenn das funktioniert, kannst du versuchen, von einem anderen Computer oder Smartphone im gleichen Netzwerk auf die Webseite zuzugreifen. Wenn das nicht klappt, musst du deine Firewall-Einstellungen prüfen.

Hosting im Web

Bevor du deine Webseite im Internet freigibst, stelle sicher, dass du keine persönlichen oder privaten Daten von dir selbst oder anderen Personen ungewollt freigibst. Mit einem GPX-Track könntest du beispielsweise aus Versehen deine Adresse preisgeben.

Zusätzlich musst du sicherstellen, dass keine Urheberrechte verletzt werden. Dazu ein Beispiel: In

diesem Tutorial wird OpenStreetMap als Hintergrundkarte verwendet. OpenStreetMap bietet offene Kartendaten an, verlangt aber, dass bei der Verwendung klar auf ihre Copyright-Seite verlinkt wird. Um diesen Link auf deiner Webseite einzufügen, kannst du die Layereigenschaften des OpenStreetMap-Layers mit Rechtsklick auf den Layer → **Eigenschaften › QGIS Server** öffnen. Füge dann im Abschnitt **Quellenangabe** den Titel "© OpenStreetMap Mitwirkende" sowie den URL https://www.openstreetmap.org/copyright hinzu. Das Plugin qgis2web verwendet diese Quellenangabe automatisch, um den Link in der unteren rechten Ecke anzuzeigen (siehe bspw. Abb. 8).

Für andere offene Daten muss separat geprüft werden, ob und wie diese verwendet werden dürfen. Eigene Daten (selbst geschossene Bilder, selbst aufgezeichnete GPX-Tracks) dürfen immer verwendet werden.

Mit Qgis2OnlineMap gibt es eine unkomplizierte Möglichkeit, einen qgis2web-Export im Internet zu veröffentlichen. Dazu muss lediglich der qgis2web-Exportordner oder eine ZIP-Datei davon hochgeladen werden. Anschliessend erhält man einen Share-Link zum Teilen der Karte, wie dieses Beispiel zeigt. In der kostenlosen Version können aktuell bis zu fünf Karten pro Nutzer gespeichert werden, jeweils mit maximal 50 MB. Die Karten können zudem passwortgeschützt werden und lassen sich in externe Websites einbetten. Für die Kontoeröffnung ist keine Kreditkarte erforderlich.

GitHub Pages

Um deine Webseite im Internet zu hosten, wollen wir nicht den eigenen PC verwenden. Das ist zwar möglich, aber aus Sicherheits- und Komplexitätsgründen nicht empfohlen. Stattdessen verwenden wir einen Online-Service. Es gibt viele verschiedene Möglichkeiten, je nach Anforderungen, z.B. Netlify, Cloudflare Pages oder Infomaniak (Schweizer Dienst, kostenpflichtig).

Für dieses Tutorial wird GitHub Pages verwendet. Dafür brauchst du einen GitHub Account.

1. Erstelle ein neues Repository und gib ihm einen sinnvollen, einfachen Namen (z.B. "Meine-Karte"). Bestätige mit dem **[Create repository]** Button.
2. Lade alle Dateien deiner Webseite mit dem Upload Fenster hoch.

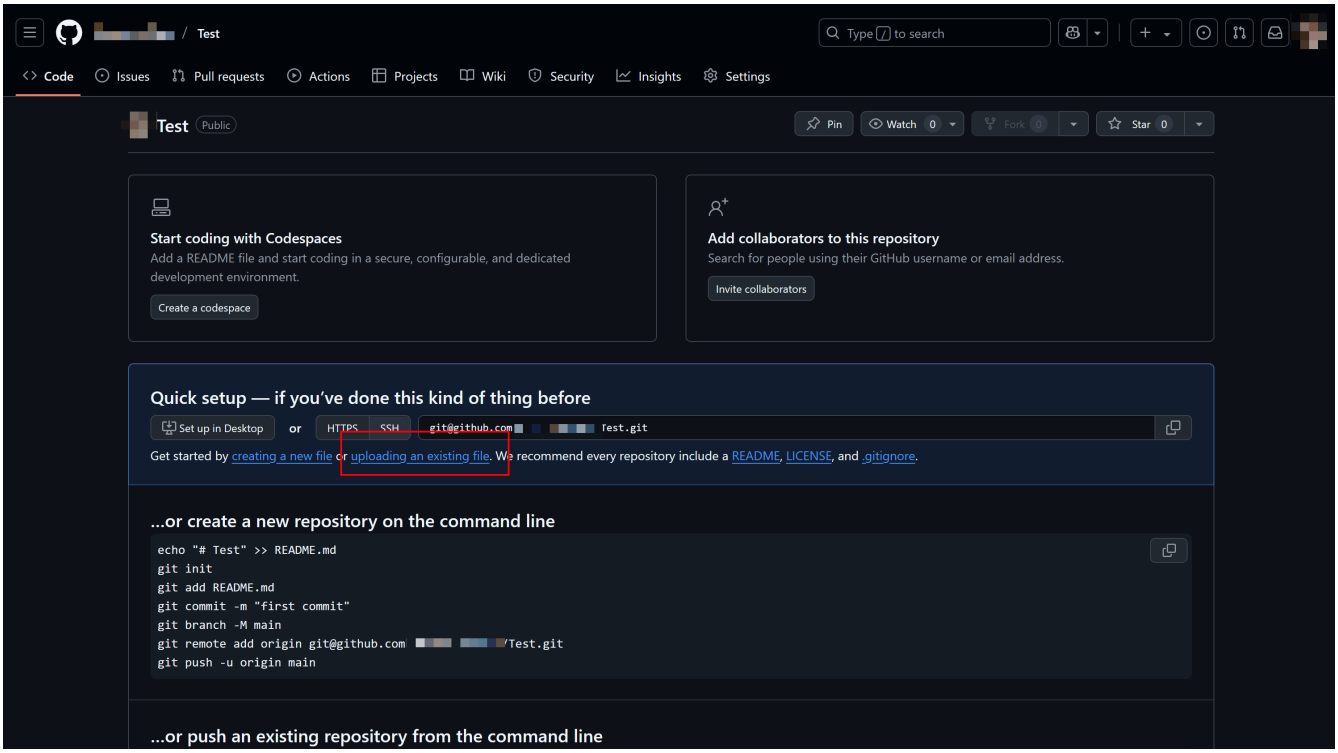

Abbildung 9. Neu erstelltes GitHub Repository mit Upload-Link.

1. Bestätige mit **[Commit changes]**, die Commit-Nachricht kann leer gelassen werden.
2. Gehe zu den Repository-Einstellungen und wähle auf der linken Seite **[Pages]**. Ändere dort den Branch von **None** zu **main**. Bestätige mit dem **[Save]**-Button.

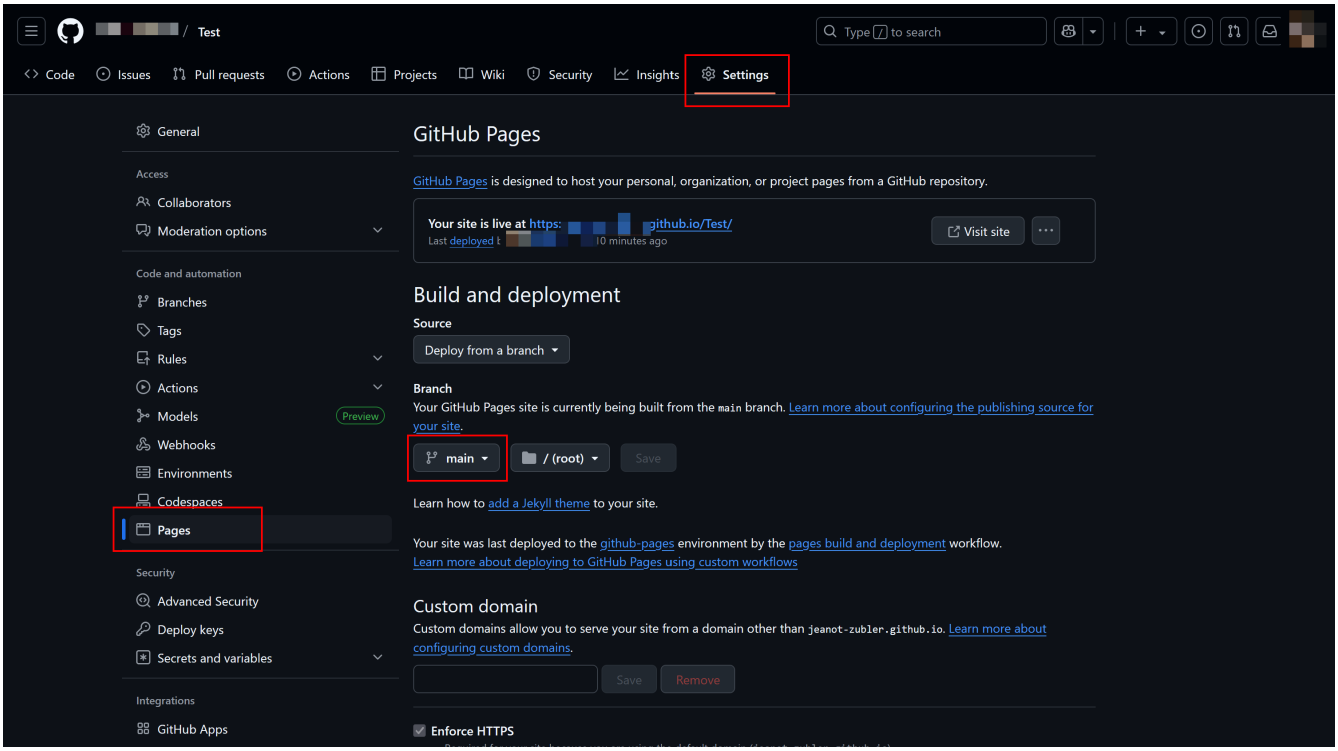

Abbildung 10. Einstellungsfenster für GitHub Pages.

1. Füge einen Link zu deiner neuen Webseite zum Repository hinzu, indem du bei **About** auf das Zahnrad klickst, und dann **Use your GitHub Pages website** anwählst. Mit **[Save changes]** bestätigen.

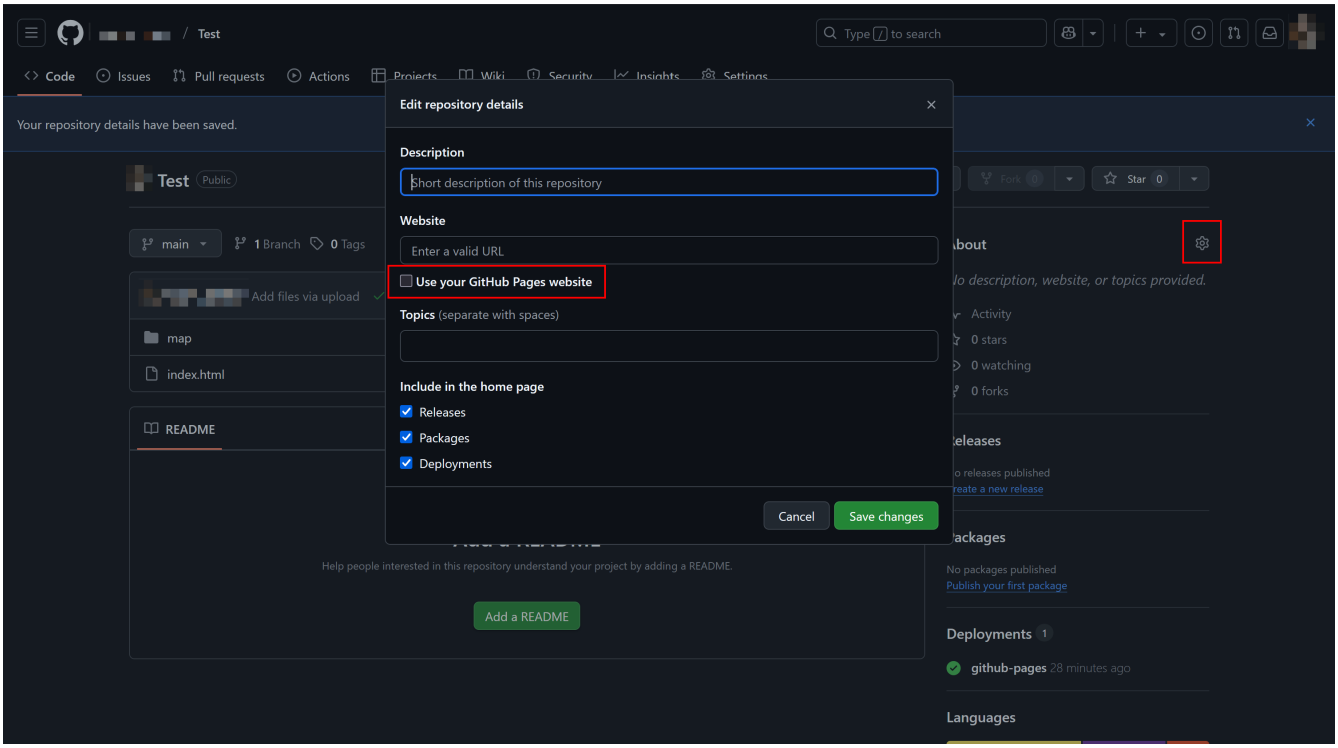

Abbildung 11. Webseitenlink zum Repository hinzufügen.

1. Jetzt siehst du unter About einen Link auf deine neue Webseite (nach dem ersten Mal aktivieren kann es einige Minuten dauern, bis alles aktiv ist).

Wenn du in Zukunft eine Änderung an deiner Seite machen möchtest, kannst du die aktualisierten Dateien wieder in GitHub hochladen, und in kürzester Zeit ist auch deine Webseite wieder aktuell.

Übungen

Aufgabe 1

Untersuche die Daten im Ordner `Output\data`. Was beinhalten die Dateien? Wie unterscheidet sich die Punkte-Datei von `Punkte.geojson`?

Die Dateien enthalten grundsätzlich dieselben Daten, die zuvor QGIS hinzugefügt wurden, also einen GPX-Track und eine Gruppe von Punkten.

Der Unterschied zwischen Punkte.geojson und der generierten Datei ist relativ gering. Es gibt zwei Hauptunterschiede:

1. Punkte.geojson ist im JSON-Format, während die Ausgabedatei (im Beispiel `Output\data\Punkte_2.js`) eine JavaScript Codedatei ist. Dabei wird ein JavaScript Objekt mit beinahe identischen Daten erstellt, und einer Variablen (hier `json_Punkte_2`) zugewiesen. Diese kann dann vom restlichen Code geladen und angezeigt werden.
2. Punkte_2.js hat zwei zusätzliche Eigenschaften, `crs` ist das verwendete Koordinatenreferenzsystem und `name` gibt der Punktegruppe einen Namen.

Aufgabe 2

Kannst du einem Punkt auf der Karte ein Bild anhängen?

Das Wiki von qgis2web enthält viele nützliche Hinweise, unter anderem auch wie ein Bild angehängt werden kann.

Wie würdest du das Popup-Feld konfigurieren, wenn nicht alle Punkte ein Bild haben?

Die Anleitung findet sich im Wiki unter Set Layers im Unterabschnitt **Media (Image or Video)**. Für die Konfiguration des Popup-Felds bietet sich **header label - visible with data** oder **no label** an. So wird kein unnötiger Text angezeigt, wenn es kein Bild hat, und das Bild kann den ganzen Platz füllen. Das kann zum Beispiel so aussehen:

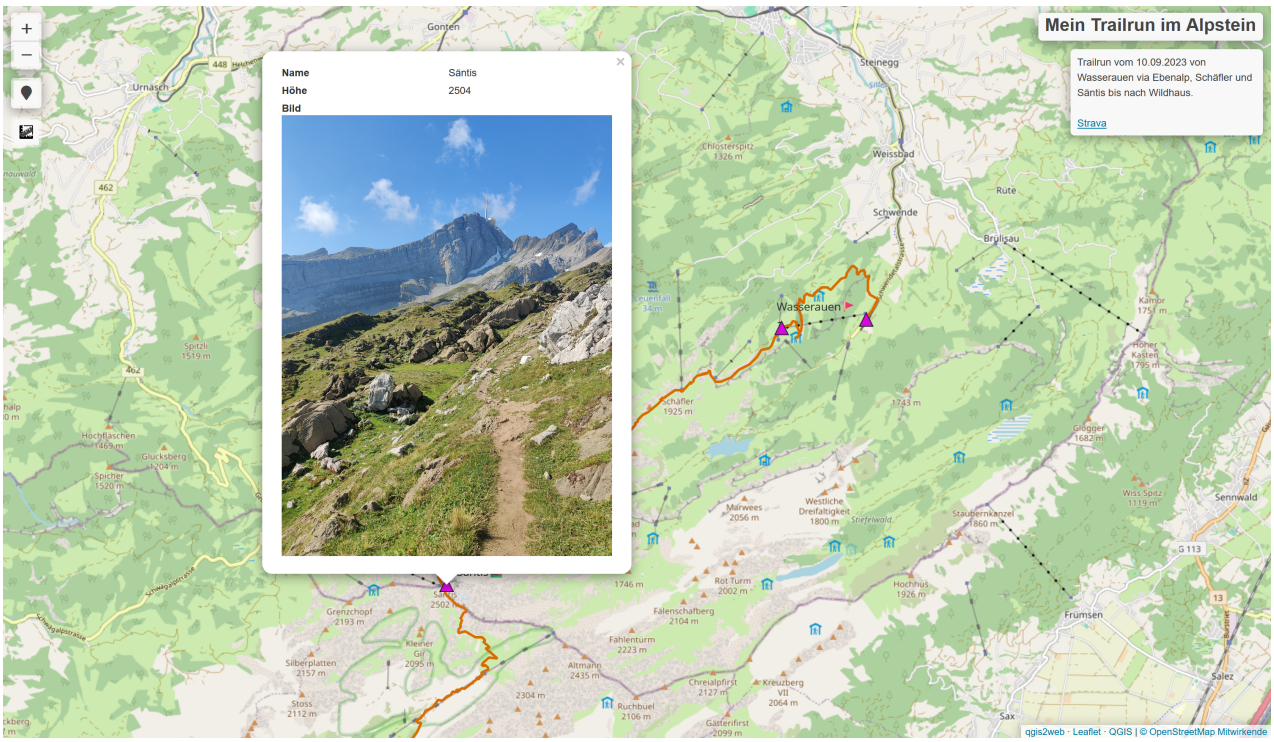

Abbildung 12. Webkarte mit einem Bild, das im Popup eines Punktes angezeigt wird. Bildquelle: eigene

Weiterführende Quellen

Dieses Unterrichtsmaterial ist Teil von OpenSchoolMaps, insbesondere des Themas **Weitere Arbeitsblätter zu QGIS 3 und Geoinformationssysteme (GIS)**. Falls mehr Informationen benötigt werden, sollte das die erste Anlaufstelle sein.

Weiterführende Quellen sind:

- Das Wiki von qgis2web für eine allgemeine Anleitung
- Das GitHub Repository von qgis2web für Fragen oder um zu prüfen, ob andere das gleiche Problem haben
- GIS Stack Exchange für Fragen, die vielleicht auch andere schon hatten

Beitragende

Für die Erarbeitung dieses Tutorials danken wir:

- Jeanot Zubler
- Ole Marin
- Tihan Morrol
- Stefan Keller

Best Practices für qgis2web

QGIS Desktop und Webkartentechnologien wie Leaflet oder OpenLayers sind sehr unterschiedliche Technologien. Es ist kaum möglich, eine 1:1-Kopie aller Funktionen eines Desktops ins Web zu bringen. Manuelle Nachbearbeitungen am exportierten Code können daher für komplexe Anforderungen notwendig sein.

Benutzerdefinierte Namen (Dateinamen, Layernamen, Feldnamen)

- **Sonderzeichen in benutzerdefinierten Namen:** In Layernamen werden einfache (') und doppelte Anführungszeichen (") zwar technisch unterstützt, sollten aber vermieden werden. In **Aliasen** (Feldbezeichnungen) sind sie nicht unterstützt.
- Für Bilder, Audio- oder Videodateien in Popups werden ausschliesslich absolute Pfade unterstützt.

Koordinatensysteme

- Basiskarten (XYZ-Tiles) sind normalerweise nur in EPSG:3857 (Web Mercator) verfügbar.
- **Projektion:** Standardmässig wird die Webkarte in EPSG:3857 exportiert. Mit der Option *Match project CRS* (Tab *Appearance*) kann stattdessen das Projekt-KRS beibehalten werden.

Symbole und Stile

- **Schriftarten (Labels):** Das Plugin exportiert nur die Information, welche Schriftart verwendet werden soll, aber nicht die Schriftartdatei selbst. Der Browser des Endnutzers muss die Schriftart lokal installiert haben oder sie muss manuell eingebunden werden.

Tipp zum Einbinden

Kopiere den folgenden Link in den `<head>` der `index.html` und passe in der CSS-Datei des Exports den Schriftnamen auf `'Roboto', sans-serif;` an:

```
<link href="https://fonts.googleapis.com/css2?family=Roboto"
```

```
rel="stylesheet">
```

- **Expressions (Ausdrücke):** Einfache Expressions (z.B. Verkettung von Feldwerten) werden unterstützt. Komplexe QGIS-Funktionen (wie Aggregatfunktionen) sowie datendefinierte Übersteuerungen für Grösse, Farbe oder Rotation werden beim Export oft ignoriert oder fehlerhaft umgesetzt.
- **Regelbasierte Symbologie:** Regelbasierte Symbologie wird grundsätzlich exportiert, ist aber fehleranfällig und sollte immer mit einem Test-Export geprüft werden. Regelbasiertes Labeling wird nicht unterstützt.
- **Fortgeschrittene Renderer:** Diagramme werden nicht unterstützt. Spezielle Renderer wie Geometry Generator, 2.5D-Symbologie werden nicht korrekt in den Web-Code übernommen und zum Teil auf einfache Symbole zurückgesetzt. Heatmaps sind nur eingeschränkt und engineabhängig unterstützt.
- **Symbolebenen (Symbol Levels):** In OpenLayers werden Symbol Levels unterstützt; in Leaflet nicht zuverlässig.

Tipps zur Vorbereitung und Optimierung

- **Geometrie-Vereinfachung:** qgis2web bietet im Tab *Export* eine eingebaute *Precision*-Einstellung (Skala 1–15; 1 = stärkste Vereinfachung). Für Polygone ist der Wert 5 ein guter Ausgangspunkt. Alternativ oder zusätzlich kann das QGIS-Werkzeug *Vereinfachen* zur Vorverarbeitung genutzt werden. Ein leichter Detailverlust ist im Web meist unsichtbar, verbessert aber die Ladezeit deutlich.
- **Attribut-Vorbereitung (Pre-Calculating):** Statt Farben oder Rotationen im Web-Export berechnen zu lassen, erstelle mit dem Feldrechner feste Spalten (z.B. `web_color`, `rotation`). Wähle im Plugin dann diese Felder direkt zur Steuerung aus. Das spart JavaScript-Rechenleistung beim Endnutzer.
- **Manuelle Feature-Sortierung:** Da Web-Karten Symbolebenen oft ignorieren, sortiere den Layer dauerhaft, bevor du ihn exportierst. Features, die am Ende der Attributtabelle stehen, werden im Web meist “obenauf” gezeichnet (Z-Order). In der Verarbeitungswerkzeugkiste das Werkzeug *Dauerhafte Sortierung* (oder *Sort*) verwenden.
- **Raster-Fallback für komplexe Symbole:** Wenn ein Stil nicht reproduzierbar ist (z.B. sehr spezifische Texturen oder künstlerische Linien), exportiere diesen Layer in QGIS als Rasterbild (GeoTIFF) und binde ihn als Raster-Layer ein. Das behält die Optik vollständig bei.
- **Datenmenge begrenzen:** Sehr grosse Vektordatensätze sind für statische qgis2web-Exporte ungeeignet. Das OpenSchoolMaps-Arbeitsblatt “Darstellen von grossen Punktmengen als Punkt-Cluster und als Heatmap” beschreibt einige Lösungen dazu.

Noch Fragen? Wende dich an uns oder an die QGIS-Community!