**Maplibre project Käfertaler Wald**

# Projektauftrag „Audioguide Käfertaler Wald“

Bearbeiter Praktikant Herr Faller; Betreuung Geodatenkompetenzzentrum

Zeitkalkulation: 14 bis 21 Arbeitstage

## Beschreibung des Ist-Standes

Die untere Forstbehörde für den Käfertaler Wald hat einen Audioguide auf der kostenfreien Plattform Digiwalk entwickelt. Diese Anwendung wird Ende 2024 eingestellt.

„Im Moment funktioniert der Audioguide so, dass wir 11 Informations-Stationen eines Rundwanderwegs über Koordinaten bei Digiwalk verankert haben. Diese sind über die App mit einem Aktivierungsradius verknüpft, so dass in diesem Radius die Station aktiviert wird und die Audiodatei zuhören ist sowie Bilder angesehen werden können. Im Wald vor Ort gibt es keine Markierungen, der Rundweg ist aber als Walk bei Digiwalk hinterlegt und kann so gefunden werden.

## Ziel

Entwicklung eines Clients auf Grundlage der zentralen Auskunft Oberfläche Geoportal der Stadt Mannheim mit den entsprechenden Funktionalitäten. Ablösung der externen Anwendung.

## Beschreibung des Projektes

Erstellen sie eine Dokumentation zu dem gesamten Projekt

### Inhalt:

* Beschreibung der Teilschritte
* Verwendete Links
* Verwendete Software Komponenten
* Workflow
* Ansprechpartner
* Probleme

### Abschnitt 1

Arbeiten sie sich in die Anwendung der bestehenden externen Oberfläche ein

### Abschnitt 2

Erarbeiten sie ein Konzept für die Erstellung einer neuen grafischen Oberfläche

Der Auskunft Client setzt auf der Open Source Software MapLibre: <https://maplibre.org/> auf.

**Dokumentation des Projekts Audioguide Käfertaler Wald**

Für einen besseren Überblick zum Projekt „Käfertaler Wald Audioguide“ ist mit folgendem Link mein GitHub-Repository zu finden.

<https://github.com/leonfallair/Kaefertaler-Wald-audio-guide>

**Überblick**

***index.html*** *(java script Funktionen mit enthalten)*

***styles.css***

sind relevante codefiles für das Darstellen der Webmap (mit Maplibre als open-source library) mit zugehörigen Funktionen (*index.html*) und styles (*styles.css*). Alle bedeutsamen Zeilen wurden für die Lesbarkeit und zum weiteren Verständnis in den files kommentiert.

**Kartenhintergrund**

Als Kartenhintergrund wurde die Streets-Karte von **Openstreetsmap** über **Maptiler** (dient als API) verwendet, da in dieser die gewünschte Route von Frau Müller-Beblavy verfügbar ist. Google Maps und Basemap DE stellen den gewünschten Weg nicht dar. Die route.js ist die Route, welche auf OSM beruht. Route\_basemap.js hat noch die Route, welche auch digiwalk verwendet hat. Beide haben **EPSG:3857** als Koordinatensystem, aber dennoch stimmen die Station-Koordinaten zwischen OSM und Basemap De nicht perfekt überein. Daher sind in ***locations.js*** und ***locations\_basemap.js*** Koordinatenabweichungen vorhanden.

**Offline-Verfügbarkeit**

Durch eingeschränkte Netz-Verfügbarkeit im Käfertaler Wald ist eine Offline-Nutzung für die Verwendung des Audioguides nötig. Eine Möglichkeit ist es über einen **Service worker** (***sw.js***) zu machen (<https://web.dev/learn/pwa/serviceworkers?continue=https%3A%2F%2Fweb.dev%2Flearn%2Fpwa&hl=de#article-https://web.dev/learn/pwa/service-workers&hl=de>).

Dieser ermöglicht eine Cachenutzung im Browser. In dem Cache können Daten zwischengespeichert werden und werden somit offline verfügbar gemacht. Der Cache ist allerdings an Speicher beschränkt und für größere Datenmengen weniger geeignet. In diesem Fall wird er verwendet um z.B die Hintergrundkarte (<https://unpkg.com/maplibre-gl/dist/maplibre-gl.js>) offline verfügbar zu machen.

Zusätzlich ist eine **Progressive web App** (PWA) eingerichtet. Damit kann die Webanwendung zum Homebildschirm hinzugefügt werden und im Prinzip als „App“ agieren. Dies ist über die Datei *manifest.json* definiert (<https://web.dev/learn/pwa/web-app-manifest?hl=de>).

Hier noch ein Link wie die PWA auf verschiedenen Devices hinzugefügt wird (<https://support.google.com/chrome/answer/9658361?hl=de&co=GENIE.Platform%3DDesktop&oco=1>).

Die Audios und Bilder werden über **jszip** und **cdnjs** verfügbar gemacht (<https://stuk.github.io/jszip/>). Der Benutzer wird über ein popup aufgefordert die Dateien zu downloaden. Ist der Download fertig, sind die Daten lokal auf dem Handy gespeichert und werden automatisch bei Offline-Nutzung der Webkarte verwendet. Damit wird die Offline-Nutzung mit einer Mischung aus Online Cache und lokal gespeicherten Dateien ermöglicht.

<https://cdnjs.cloudflare.com/ajax/libs/jszip/3.7.1/jszip.min.js>

<https://cdnjs.cloudflare.com/ajax/libs/FileSaver.js/2.0.5/FileSaver.min.js>

**Einbettung in zukünftiges Geoportal Mannheim über NETGis**

Der Audioguide soll über das Mannheimer Geoportal angeboten werden. Dafür muss der relevante code in das vorhandene System eingebettet werden. Folgende files sind hiefür nötig und wurden entsprechend angepasst.

***Index.html*** (webkartenanwendung mit allen Inhalten)

***Filtered\_content.html*** *(gefiltert von index.html ohne maptiler und maplibre verbindung)*

***Styles.css***

***sw.js*** *(service worker für Cache und Offline-Nutzung)*

***manifest.json*** *(falls PWA ermöglicht werden soll)*

**Referenzen**

<https://www.digiwalk.de/walks/auf-gehts-in-den-kaefertaler-wald-1>

bisheriger externer Audioguide

<https://netgis.de/webgis-3/>

Dienstleister für neues Geoportal

Basemap DE

<https://basemap.de/web-vektor/>

**Verwendete Links**

<https://docs.maptiler.com/sdk-js/examples/control-style-switcher/?utm_medium=social&utm_source=youtube&utm_campaign=2022-11%20|%20cloud%20|%20maplibre&utm_content=maplibre4&utm_term=desklink>

Beispielcode für mapswitching über maptiler

<https://maplibre.org/maplibre-gl-js/docs/examples/set-popup/>

attach a popup to a marker instance

<https://maplibre.org/maplibre-gl-js/docs/examples/popup-on-click/>

display a popup on click

<https://maplibre.org/maplibre-gl-js/docs/examples/locate-user/>

locate the user

<https://www.youtube.com/watch?v=iFVlbEb5P5s&list=PL8ApLBFiTZlwpTfOGF6LGJc61IVekKa3s>

Web mapping tutorial

<https://web.dev/learn/pwa>

Einführung zum Erstellen einer PWA und service worker

<https://stuk.github.io/jszip/>

<https://cdnjs.cloudflare.com/>

zum Erstellen eines dynamischen ZIP Ordners, dessen Daten in Webanwendung verwendet werden

**Verwendete Software Komponenten**

Maplibre

Maptiler

Service worker

Progressive Web App (PWA)

Vs code als code editor

QGIS um linestring für Route zu zeichnen und genaue Bestimmung der Koordinaten

Cdnjs

jszip

**Ansprechpartner**

Frau Müller-Beblavy 5362 Durchwahl