

Interaktive historische Karten über Altona

Dokumentation

Das Kartenportal ist erreichbar unter

- <https://altona.ws/karten>
- <https://pod-o-mart.github.io/bzag-kartenwerkstatt>

Mit den Einzelkarten

- <https://altona.ws/karte> – Hauptkarte
- <https://altona.ws/karte/quarree> – Spezialkarte Hertie-Quarree / Mercado
- <https://altona.ws/karte/quarree/friedhof> – Spezialkarte jüdischer Friedhof Ottensen

1 Software-Grundlagen

- 1.1 **Programmiersprachen:** Ein wenig *HTML/CSS* und viel *JavaScript*
- 1.2 **Kartengrundlage:** JavaScript-Bibliothek **Leaflet** <https://leafletjs.com>. Zusätzlich installierte Module <https://leafletjs.com/plugins.html>:
 - 1.2.1 *Leaflet.HtmlLegend* – Einbindung einer Legende pro Karte mit Steuerung der Transparenz
 - 1.2.2 *Leaflet.Control.Layers.Tree* – Baumasicht im Kartenauswahlmenü
 - 1.2.3 *Leaflet.zoominfo* – Anzeige des Zoom-Levels
 - 1.2.4 *Leaflet Ajax* – GeoJSON-Dateien einbinden
 - 1.2.5 *Leaflet.Spin* – Ladeanzeige (funktioniert noch nicht ganz)
 - 1.2.6 *leaflet-hash* – Zoomlevel und Koordinaten in Adressleiste anzeigen und damit verlinkbar machen
- 1.3 Weitere JavaScript-Bibliotheken: *Bootstrap*, *JQuery* (Grafik- und Anzeigefunktionen)
- 1.4 Die Karte ist zur Veröffentlichung im Internet konzipiert, kann aber, wenn lokal auf dem Computer gespeichert, auch ohne Internetverbindung verwendet werden, da JavaScript grundsätzlich immer browserseitig ausgeführt wird, siehe Begriffsdefinitionen
- 1.5 Die Kartenanzeige wird innerhalb der Quelltext-Datei <https://altona.ws/karte/index.html> durch eigens zu programmierende *JavaScript*-

Anweisungen definiert. Es wird festgelegt, welche Funktionen die Karte haben soll und aus welchen Quellen die verschiedenen Karten geladen werden sollen. Siehe Tutorials <https://leafletjs.com/examples.html>

1.6 Hilfsprogramme

1.6.1 Folgende Programme haben die Möglichkeit, WMS einzubinden, eigene Karten zu georeferenzieren, georeferenziertes Datenmaterial anzuzeigen und zu bearbeiten

1.6.2 **QGIS** <https://qgis.org/de> – Sehr komplexes *geographisches Informationssystem* (GIS) mit der Möglichkeit, eigene, auf Daten basierte Karten zu erstellen und zwischen den verschiedenen Koordinatenreferenzsystemen zu konvertieren (Koordinaten können auf verschiedene Weise dargestellt werden, winkeltreu, flächentreu, Mercator usw.). Nützlich, da *LeafletJS*, *GoogleMaps*, *Openstreetmap* usw. nur das Koordinatenreferenzsystemen *Mercator* verstehen, die Koordinaten insbesondere des *Geoportals HH* aber ein anderes Referenzsystem nutzt. Siehe → *Datenaufbereitung*

1.6.3 **JOSM** <https://josm.openstreetmap.de> – Auf *Openstreetmap* spezialisiertes GIS. Hier können Kartenbearbeitungen direkt auf *Openstreetmap* hochgeladen werden, vor Allem aber, können Datensätze für eigene Weiterverwendung heruntergeladen werden. Siehe → *Datenaufbereitung*

1.6.4 **ogr2ogr** – ein Kommandozeilenprogramm zur Umformatierung von Geodaten (insbesondere zu *geoJSON*) und Koordinatenreferenzsystemen (insbes. *Mercator*)

1.6.5 Beliebiger **Texteditor** mit REGEX-Funktion

2 Funktionen

2.1 **Kartenanzeige** – Die einzelnen Karten können unabhängig vom Format als *Basemaps* (Grundkarten) oder *Layers* (Ebenen/Folien) definiert werden

2.1.1 **Externe Quellen** (serverseitig)

2.1.1.1 Format: *Tiles* / Kacheln. Muster:
<https://tile.openstreetmap.org/{z}/{x}/{y}.png>

2.1.1.2 Format: *WMS*.
Muster: https://geodienste.hamburg.de/HH_WMS_HamburgDE?REQUEST=GetCapabilities&SERVICE=WMS

2.1.1.2.1

2.1.2 **Interne Quellen** (lokal abgespeichert), Format: Grafikdatei
Muster: <https://altona.ws/karte/quarree/images/alternativvorschlag1.png>

2.2 **Datenanzeige** – bei altona.ws lokal als *GeoJSON* abgespeichert, aber auch serverseitig möglich, wenn man denn überhaupt eine gute Quelle findet.
Muster: https://altona.ws/karte/geojson/infotafeln_stadtteilarchiv.geojson

2.3 Mischformen

2.3.1 Grenzkarten – *Kartenanzeige* aus *GeoJSON*-Dateien (*intern*)

2.3.2 Denkmalschutz (Punkte) – *WMS*-Karte (*extern*) zusammen mit *GeoJSON*-Datei (*intern*)

2.4 **Adressbuch** – Suchmaske zur Abfrage der Adressbücher 1782-1976 (Stabi)

3 Karten- und Datenquellen

3.1 Karten

3.1.1 WMS

3.1.1.1 **Geoportal HH** <https://geoportal-hamburg.de/geo-online> A) → einzelne Karte → Legende und Information → *WMS*-Adresse B) → einzelne Karte → Legende und Information → weitere Metadaten → Verweise

3.1.1.2 Bibliotheken, Online-Archive

3.1.1.2.1 Recherche | Stabi Hamburg

https://digitalisate.sub.uni-hamburg.de/karten.html?tx_dlf%5Bcollection%5D=137

3.1.1.2.2 Deutsche Digitale Bibliothek <https://www.deutsche-digitale-bibliothek.de>

3.1.1.3 <https://wiki.openstreetmap.org/wiki/WMS>

3.1.1.4 Siehe → *Weitere (Karten und/oder Daten)*

3.1.2 Tiles

3.1.2.1 https://wiki.openstreetmap.org/wiki/Tile_servers

3.1.2.2 <https://wiki.openstreetmap.org/wiki/Tiles#Servers>

3.1.2.3 <https://leaflet-extras.github.io/leaflet-providers/preview>

3.1.2.4 <https://mc.bbbike.org/mc>

3.2 Datenmaterial

3.2.1 In den allermeisten Fällen müssen Daten zur Verwendung in unserem Kartenportal nachbearbeitet werden. Sie können nur verwendet werden, wenn wir die Daten ins *GeoJSON*-Format gebracht haben. Siehe → *Datenaufbereitung*

3.2.2 **Geoportal HH** <https://geoportal-hamburg.de/geo-online> **A)** → einzelne Karte → *Legende und Information* → *Datensatz herunterladen* **B)** → einzelne Karte → *Legende und Information* → *weitere Metadaten* → *Verweise*

3.2.3 Openstreetmap

3.2.3.1 Alle georeferenzierten Daten wie Info-Punkte, Linien und Polygone können nach bestimmten Vorgaben an einem Stück heruntergeladen werden. Hierfür verwenden wir das bereits genannte Programm JOSM. *Beispiele:* Stolpersteine, OSM historische Objekte.

3.2.4 Wikipedia

3.2.4.1 Georeferenzierte Daten wie Artikel, Bilder etc. können nach Kategorie (und somit thematisch) an einem Stück heruntergeladen werden. *Beispiel:* Wikipedia-Infolayer.

3.2.4.2 https://de.wikipedia.org/wiki/Wikipedia:WikiProjekt_Georeferenzierung/Hauptseite/Datenextraktion

3.2.4.3 <https://kmlexport.toolforge.org>

3.2.4.4 <https://de.wikipedia.org/wiki/GeoNames>

3.2.5 **KML** aus vielen Quellen, Beispiel Infotafeln Stadtteilarchiv https://www.google.com/maps/d/viewer?mid=1JqzT6lX-I3UqjbLodU8bIFm_ZBw&ll=53.55154181082571%2C9.940161300000007&z=15. GoogleMaps-Karten können als KML-Datei heruntergeladen werden.

3.3 Weitere (Karten und/oder Daten)

3.3.1 Geodatenzentrum des Bundesamtes für Kartographie und Geodäsie <https://gdz.bkg.bund.de>

3.3.2 Bundesamt für Kartographie und Geodäsie (BKG) <https://geodatenkatalog.de>

3.3.3 DigitalerAtlasNord <https://danord.gdi-sh.de>

3.3.4 GEO-LEO <https://geo-leo.de>

3.3.5 <https://geoportal.de>

4 Karten- und Datenaufbereitung

4.1 Karten

4.1.1 Hosting bei MapWarper

4.1.1.1 Die zu erstellende Karte sollte bevorzugt bei *MapWarper* abgelegt werden, da einfacher und für alle zur Mitarbeit zugänglich

4.1.1.2 Gewünschte Karte recherchieren und herunterladen Siehe → *Karten- und Datenquellen*

4.1.2 Gewöhnlich wird die Karte als PDF bereitgestellt. Da wir aber eine Grafikdatei benötigen, muss das PDF in einem Grafikprogramm (*Photoshop, GIMP* etc.) geöffnet werden. Hier kann die Karte zunächst zurechtgeschnitten und eine passende Auflösung gewählt werden (nicht zu niedrig, kartenrelevante Vergrößerung mitbedenken).

4.1.2.1 Gelegentlich ist die Karte Teil eines Buches als PDF. In diesem Falle sollte man mit einem Hilfsprogramm (z.B. PDFshuffler, PDFsam) die Seite mit der bestimmten Karte einzeln abgespeichert und erst dann weiterverarbeitet werden.

4.1.3 Als Grafikdatei exportieren (am Besten JPG oder PNG)

4.1.4 Grafikdatei auf *MapWarper* <https://mapwarper.net> hochladen (zuvor Konto erstellen). Dort die Karte georeferenzieren. Unter Karte → *Export* die WMS-Adresse notieren und dem Kartenportal-Administrator zwecks Einbindung mitteilen

4.1.5 Hosting auf der eigenen Webspaced

4.1.5.1 Anleitung unvollständig, wird zur nächsten Version überarbeitet

4.1.5.2 Die grafisch vorbereitete Karte muss in QGIS georeferenziert werden.

4.1.5.3 QGIS: Grafik georeferenzieren und als geoTIFF abspeichern

4.1.5.4 - Koordinaten des geoTIFFs notieren:

Raster → *Sonstiges* → *Informationen...* → *Datei wählen* → *OK* → *Raster Info*
rauskopieren. ACHTUNG Länge/Breite umgekehrt

4.1.5.5 - geoTIFF zu PNG formatieren und dieses in Karte einbinden (links oben / rechts unten):

4.1.5.6 `var imageUrl = 'images/1_planung-austiff.png';`

4.1.5.7 `var imageBounds = [[53.5538036, 9.9305205], [53.5518156, 9.9338432]];`

4.1.5.8 `L.imageOverlay(imageUrl, imageBounds).addTo(map);`

4.1.5.9 <https://jsfiddle.net/user2314737/Lfzrqvet/>

4.1.5.10 https://www.tutorialspoint.com/leafletjs/leafletjs_overlay.htm

4.1.5.11 In QGIS georeferenziertes geoTIFF nach JOSM exportieren: In JOSM Hintergrund > (Mehr... >) Bild importieren

4.2 Daten

4.2.1 Geoportal HH – mithilfe von ogr2ogr und QGIS

4.2.1.1 Kompliziert – Beschreibung folgt hier in der nächsten Version

4.2.1.2 Datei entzippen, entzippte *XML*-Datei muss per *ogr2ogr* in die enthaltenen verschiedene Rubriken gesplittet werden. Die hieraus entstandenen *GML*-Dateien können dann mit *ogr2ogr* ins *GeoJSON*-Format exportieren. Danach muss mit *QGIS* noch das Koordinatenreferenzsystem konvertiert werden. Erst dann können die *GeoJSON*-Dateien in *Leaflet* eingebunden werden. Eine genaue Anleitung folgt hier in der nächsten Version.

4.2.2 Openstreetmap – mithilfe von JOSM

4.2.2.1 *Bearbeiten* → *Einstellungen* → *Expertenmodus* aktivieren

4.2.2.2 *Datei* → *Daten herunterladen* → *Reiter von Overpass-API herunterladen*, siehe https://wiki.openstreetmap.org/wiki/Overpass_turbo/Wizard

4.2.2.3 *Reiter Gebiete um bestimmte Orte* → *Server: Nominatim*

4.2.2.4 *Ortsamen eingeben*, z.B. „Altona“ und „Altona, Hamburg, Deutschland“ auswählen

4.2.2.5 *Abfrage-Wizard: Suchbegriff ein bestimmter Tag*, siehe <https://taginfo.openstreetmap.org>. Habe für diese Karte, jeweils nacheinander, die Abfrage für die *Tags*

4.2.2.5.1 `historic=*`

4.2.2.5.2 `tourism=museum`

4.2.2.5.3 `heritage=*`

4.2.2.6 *Erstelle Anfrage und führe diese aus*

4.2.2.7 Eine neue Datenebene mit den dem *Tag* entsprechenden Punkten, Linien wurde erstellt. Diese kann noch nachbearbeitet werden, Punkte entfernt werden usw.

4.2.2.8 Diese Ebene könnte nun schon in *GeoJSON* konvertiert werden. Allerdings müssen 3 *GeoJSON*-Dateien erstellt werden, jew. eine für Punkte, Linien und Polygone. Das muss sein, weil *Leaflet* leider nur *GeoJSONs* auslesen will, in denen Punkte, Linien und Polygone nicht zusammen vorkommen, ist leider so. In der Karte können die drei Dateien dann wieder kombiniert werden.

4.2.3 Wikipedia

4.2.3.1

4.2.4 Die von *Geoportal HH*, *Openstreetmap*, *GoogleMaps*, *Wikipedia* usw. gewonnenen Daten liegen uns nach der Quellenrecherche in verschiedensten Formaten vor. Wir müssen diese in *GeoJSON*-Dateien umformatieren, damit wir sie im Kartenportal verwenden können.

4.2.5 **GML** – ggf. muss eine KMZ- oder KML-Datei heruntergeladen und diese zunächst nach umformatiert werden (und danach erst nach GeoJSON). GeoPortal HH und Google Maps

4.2.6 **XML** – in unserem Falle vom GeoPortal HH

5 Kollaboratives Arbeiten

5.1 **Ohne Programmierkenntnisse – Möglichkeiten:** Kartenrecherche, grafische Aufbereitung. Auf *MapWarper*: Hochladen und Georeferenzierung, Verbesserung der Georeferenzierung bestehender Karten. Schließlich Weitergabe der WMS-Daten an den Administrator des Kartenportals zwecks Einbindung

5.2 **Mit Programmierkenntnissen – Möglichkeiten:**

5.2.1 Im Kartenportal: Karteneinbindung, Verbesserung/Veränderung der Software und damit des Kartenportals über **GitHub** <https://github.com/pod-o-mart/bzag-kartenwerkstatt>

5.2.2 Eigene Karte erstellen. Benötigt wir eigener Webspace (Internet-Adresse mit Administratorzugriff). Entweder anhand von Leaflet von Grund auf neu erstellen, oder das Kartenportal von **GitHub** <https://github.com/pod-o-mart/bzag-kartenwerkstatt> *forken* (absplaten) und für eigene Zwecke modifizieren

6 Einbindung der Karten und Daten

6.1 Die letztliche Einbindung kann nur mit Programmierkenntnissen bewerkstelligt werden. Der geübte Programmierer wird eine fertig vorbereitete, neue Karte in wenigen Minuten einbinden können.

6.2 **WMS**

6.2.1 Quelladresse feststellen, beispielsweise https://geodienste.hamburg.de/HH_WMS_HamburgDE?REQUEST=GetCapabilities&SERVICE=WMS

6.2.2 Mit der Adresse bei *QGIS* oder *JOSM* die Karte laden

6.2.3 Dort feststellen, welche Ebenen der Karte verwendet werden sollen

6.2.4 Quelladresse und Ebenennummern in eigenes Kartensystem als Grundkarte oder als Layer mithilfe von *Leaflet* einbinden

6.3 **Tiles**

6.3.1 Quelladresse feststellen, beispielsweise <https://tile.openstreetmap.org/{z}/{x}/{y}.png>

6.3.2 Quelladresse in eigenes Kartensystem als Grundkarte oder als Layer mithilfe von *Leaflet* einbinden

6.4 **Selbsterstellte Karte**

6.4.1 Aufbereitete Karte bei *MapWarper* <https://mapwarper.net> hochladen, georeferenzieren und unter *Karte* → *Export* die WMS-Adresse feststellen

6.4.2 Quelladresse in eigenes Kartensystem als Grundkarte oder als Layer mithilfe von *Leaflet* einbinden

6.5 Daten

6.5.1 *GeoJSON*-Datei auf Homepage des eigenen Kartensystems hochladen

6.5.2 Quelladresse auf der Homepage in eigenes Kartensystem als Layer mithilfe von *Leaflet* einbinden

6.6 Mischformen

6.6.1 Karte kombiniert mit *GeoJSON*-Daten wie z.B. Marker (anklickbare Punkte mit näheren Informationen)

6.6.1.1 Quelladressen der Karte und der *GeoJSON*-Datei zusammen in *Leaflet* als *einen* Layer definieren (beide werden bei Aktivierung des Layers im Kartenportal als eine Karte angezeigt)

6.6.2 Karte aus *GeoJSON*-Datei: Möglich, da in *GeoJSON* georeferenzierte Punkte, Linien und Polygone definiert werden können. Diese können dann grafisch von *Leaflet* aufbereitet werden, dass sie als Karte erscheinen

Begriffsdefinitionen

lokal, intern auf dem eigenen Computer oder der eigenen Homepage.

↑

extern von anderen Internetquellen.

browserseitig Inhalte werden erst vom Browser auf dem eigenen Computer verarbeitet.

↑

serverseitig Inhalte werden von der jew. Homepage bereits fertig verarbeitet abgeliefert.

JavaScript Programmiersprache, die erst im Browser gestartet wird, kurz **JS**. Mit ihr können dem Browser beispielsweise Anweisungen gegeben werden, welche Inhalte in welcher Form nachgeladen werden sollen (Karten ...), was ein Mausklick auslösen soll, usw. Ist in jedem Browser installiert, so gut wie alle modernen Webseiten nutzen *JS*.

JavaScript-Bibliothek In der Karte deklarierte, vorgefertigte Datei, die *JS* um spezielle Funktionen erweitert, in unserem Falle sind das Kartenfunktionen.

JavaScript-Modul Eine Unter- oder Zusatzbibliothek. Auch: **Plugin**.

GeoJSON	Ein von Kartenprogrammen wie <i>QGIS</i> oder <i>JOSM</i> genutztes Dateiformat, mit dem Punkte, Flächen, georeferenzierte Informationen uvm. abgespeichert werden können
Tiles	Tiles (Kacheln) bezeichnet eine Schnittstelle zum Abrufen von Landkarten. Wird von vielen Diensten angeboten, z.B. Gmaps oder Openstreetmap.
WMS	Web Map Service (WMS) ist eine Schnittstelle zum Abrufen von <i>Auszügen</i> aus Landkarten. Hier können einzelne Ebenen einer Karte ein- oder ausgeschaltet werden, oft wird eine Legende geboten, eine Karte kann sehr individuell ausgestaltet werden. Wird besonders von staatlichen Stellen verwendet.