

GIS Client für Tombejo:

Neben einer handelsüblichen Tombejo-Installation braucht der funktionierende GIS Client folgendes:

Benötigt wird:

Einen Geoserver (z.b. 2.6.2), am besten Standalone

Tombejo und Geoserver (<http://geoserver.org/>) für unix und windows.

Eine Datenbasis, bevorzugt POSTGRES, bevorzugt 9.4.0.1. mit POSTGIS plugin. Es gibt eine Zip Datei die die Installationsfiles für Windows enthält, und zwar unter ocw2.de/home/frauch/to_install.zip

Installation:

Beim einrichten des Geoservers muss darauf geachtet werden, dass das JDK Version 7 oder größer ist

Beim einrichten von POSTGRES wird im Installationsprozess das POSTGIS Plugin optional hinzugefügt. Hier muss darauf geachtet werden, dass die Bitrate mit der von Postgres übereinstimmt (normalerweise sollten es 64 bit sein).

Einrichtung der Datenbank:

Postgres Admin wird mit Pgadmin3 gestartet. Zuerst muss eine Datenbank angelegt werden (Name: Ahaus_neu), bei der unter extensions POSTGIS ausgewählt wird. Klappt das nicht hat das vermutlich was mit 32/64 bit zu tun.

Werden die Daten aus Shapefiles im importiert, so müssen jetzt alle shapefiles mit dem Plugin „Postgis Shapefiles and DBF Loader“ geladen werden. Wichtig ist (momentan noch), dass die Option „Preserve case of column names“ gesetzt wird, sonst kommt der gisclient damit nicht zurecht, da diese vom Geoserver incl. case übernommen werden.

Einrichtung des Geoservers

Erstelle Arbeitsbereich:

Name: Ahaus_neu

Namensraum URI: Ahaus_neu

Neue Datenquelle:

Name der Datenquelle: Ahaus_neu

Beschreibung: Ahaus neu

Database: Ahaus_neu

user: postgres

passwd: postgres

Nun alle layer publizieren mit Koordinatensystem EPSG:31466. Die Namen der Layer müssen mit Großbuchstaben beginnen, Titel auch.

Gruppenlayer hinzufügen: Ahaus_neu_gesamt

Dabei die Stile: Rasen-grass Gebäude-rot Weg_wassergebunden-blau Offentl_gruen-green, alles andere-polygon.

Die Stile müssen im Moment noch aus ocw2.de geladen werden.

Apache Proxy einrichten

Damit es nicht zu Cross-Domain-Konflikten kommt, muss der Apache direkt auf den Geoserver weiterleiten. Dafür muss in die apache2.conf folgendes eingetragen werden:

```
# AJP13 Proxy
<IfModule mod_proxy.c>
<IfModule mod_proxy_ajp.c>
Include „conf/extra/httpd-ajp.conf“
</IfModule>
</IfModule>
```

```
ProxyPass /geoserver http://localhost:8080/geoserver
ProxyPassReverse /geoserver http://localhost:8080/geoserver
```

Außerdem müssen folgende Zeilen auskommentiert werden:

```
LoadModule proxy_module modules/mod_proxy.so
```

```
LoadModule proxy_connect_module modules/mod_proxy_connect.so
```

```
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Aufbereitung der Daten

```
UPDATE "Graeber" SET "Geo_ID" = NULL WHERE "Geo_ID" = 0
UPDATE "Graeber" SET "fzeig" = NULL WHERE "fzeig" = 0
```

So lange das hier ausführen, bis es nichts mehr gibt:

```
DELETE FROM "Graeber" WHERE "gid" IN (
SELECT
MAX("Graeber"."gid")
```

```
FROM
public."Graeber"
WHERE "Graeber"."Geo_ID" IS NOT NULL
GROUP BY "Graeber"."Geo_ID",fzeig HAVING COUNT(*)>1
)
```

Anschließend das hier

```
CREATE UNIQUE INDEX "Graeber_Geo_ID" ON "Graeber" USING
btree("Geo_ID","fzeig")
```

Konvertierung von Map-Info Daten nach Shapefile:

Im Terminal:

```
ogr2ogr -f 'ESRI Shapefile' *Zieldatei* *Quelldatei*
```

Linux

Um den Geoserver mit sh bin/startup.sh oder sudo -b sh /usr/share/geoserver-2.8.3/bin/startup.sh zu starten muss vorher das eingegeben werden

```
sudo bash -c "echo JAVA_HOME=*java-pfad*>> /etc/environment"
sudo bash -c "echo GEOSERVER_HOME=/*geoserver-pfad*>> /etc/environment"
```

Postgis

CREATE EXTENSION postgis;

kindex unique machen in layer graeber.


ALTER TABLE graeber ADD CONSTRAINT unique_kindex UNIQUE (kindex);

Graeber layer:

idinteger

NOT NULL

| | | | | | | | | | |
|-------------------------------------|-------------------------------|--------|-------|------------|--------|-------|------------|------|--|
| nextval('graeber_id_seq'::regclass) | | Browse | Alter | Privileges | Drop | | | | |
| geom | geometry(MultiPolygon,900915) | | | | Browse | Alter | Privileges | Drop | |
| gname | character varying(80) | | | | Browse | Alter | Privileges | Drop | |
| grabart | character varying(80) | | | | Browse | Alter | Privileges | Drop | |
| nutzende | character varying(80) | | | | Browse | Alter | Privileges | Drop | |
| gmzustand | character varying(80) | | | | Browse | Alter | Privileges | Drop | |
| pfzustand | character varying(80) | | | | Browse | Alter | Privileges | Drop | |
| abtl | character varying(80) | | | | Browse | Alter | Privileges | Drop | |

| | | | | | | | | |
|---------------------------|-----------------------|--|---------------|---|------------------------|-----------------------|----------------------------|----------------------|
| reihe | character varying(80) | | | | Browse | Alter | Privileges | Drop |
| stelle | character varying(80) | | | | Browse | Alter | Privileges | Drop |
| stellebis | character varying(80) | | | | Browse | Alter | Privileges | Drop |
| kindex | numeric(10,0) | | NULL::numeric |  | Browse | Alter | Privileges | Drop |

Shapefiles in Postgis umwandeln

shapefile2postgis.php:

: tombejo_gis_client:shapefile2postgis

Erzeuge gis spalte:

```
SELECT AddGeometryColumn ('public','my_spatial_table','geom',4326,'POINT',2);
```

<http://postgis.net/docs/AddGeometryColumn.html>

und gleich der ganze table:

```
CREATE TABLE items (
  id integer NOT NULL,
  geom geometry(Point,4326),
  clazz character varying(10)
);
```

```
<lang-sql prettyprint prettyprinted> ALTER TABLE items OWNER TO postgresql; ALTER
TABLE ONLY items ADD CONSTRAINT items_pkey PRIMARY KEY (id); </lang-sql>
```

From:

<http://wiki.girona.de/> - **Girona Wiki**

Permanent link:

http://wiki.girona.de/doku.php?id=tombejo_gis_client&rev=1462884632

Last update: **2016/12/20 10:03**

