Geopackage and Geoserver

JONG LEE

GEOSERVER FOCUS WORKING GROUP

4/9/2021

GeoPackage

- https://www.geopackage.org/
- The GeoPackage Encoding Standard describes a set of conventions for storing the following within an SQLite database:
  - vector features
  - tile matrix sets of imagery and raster maps at various scales
  - attributes (non-spatial data)
  - Extensions
- OGC GeoPackage specification 1.0.1 published in year 2015
- Latest version 1.3.0 in year 2020 (https://www.geopackage.org/spec130/index.html)

GeoPacakge Implementations

- GDAL
  - Vector Features and Tiles (raster) since v2.0
- QGIS
  - Vector Features (read/write) since 2.10.1
  - Tiles (read) since 2.18
- Geoserver
  - GeoPacakge plugin
  - Handle both Vector Features and Tiles
- ESRI ArcGIS
  - Vector features (since ArcGIS 10.2.2)
  - Tiles (since ArcGIS 10.3
- GeoTools
  - Vector Features and Tiles (raster) since 11.0
  - Recently added GeoPackage R-Trees
- NGA open source lib
  - http://ngageoint.github.io/GeoPackage/
  - OGC certified
- https://www.geopackage.org/implementations.html

A Quick Comparison

ESRI Shapefile

- Multiple files (.shp, .shx, .dbf, .prj)
- Limitation because of DBF
  - 10 ch length for column names
- 1 Shapefile has 1 Feature Type
  - Road.shp has "road" feature type
- Can't contain Raster data

- Galveston buildings shapefile
  - 40.7 MB

GeoPackage

- Single file
- No limitations like DBF

- 1 GeoPackage could have multiple Feature Types
- It contains raster data and other attribute tables
- Galveston buildings geopackage
  - 3.66 MB

GPKG Support in Geoserver

- GeoPackage is Core Functionality of Geoserver
  - Vector: https://docs.geoserver.org/latest/en/user/data/vector/geopkg.html
  - Raster: https://docs.geoserver.org/latest/en/user/data/raster/geopkg.html
- GeoPackage Extension (Community plugin)
  - https://docs.geoserver.org/latest/en/user/community/geopkg/
  - The GeoServer GeoPackage extension also allows to create a completely custom made GeoPackage with multiple layers, using the GeoPackage process.

Uploading GeoPackage to Geoserver

- Using Geoserver REST endpoints
- https://docs.geoserver.org/latest/en/user/rest/index.html#rest
- Uploading geopackage (or ESRI shapefile) means creating a store at Geoserver
  - https://docs.geoserver.org/latest/en/user/rest/stores.html

```
curl -v -u <USR>:<PASSWORD> -XPUT -H "Content-type: application/zip"
   --data-binary @<PATH&FILENAME.ZIP>
http://<HOSTNAME>:<PORT>//geoserver/rest/<WORKSPACE>/datastores/<FILENAME>/file.shp


curl -v -u <USR>:<PASSWORD> -XPUT -H "Content-type: application/x-sqlite3"
   --data-binary @<PATH&FILENAME.GPKG>
http://<HOSTNAME>:<PORT>/geoserver/rest/workspaces/<WORKSPACE>/datastores/<FILENAME>/file.gpkg
```

Uploading GeoPackage to Geoserver

- Using Jetty Httpclient
- Reason:
  - Current Geoserver manager java library is using Apache Common Httpcomponent (old version)
  - If I install another version (latest apache httpclient), there maybe a class loading issues.

```
public static boolean uploadGpkgToGeoserver(String store, File gpkgFile) {
    String url = GEOSERVER_REST_URL+"/rest/workspaces/" + GEOSERVER_WORKSPACE + "/datastores/"
            + store + "/file.gpkg";
    URI uri = URI.create(url);
    Authentication.Result auth = new BasicAuthentication.BasicResult(uri, GEOSERVER_USER, GEOSERVER_PW);


    HttpClient httpClient = new HttpClient();
    try {
        httpClient.start();
        Request request = httpClient.newRequest(uri);
        request.method(HttpMethod.PUT);
        request.file(gpkgFile.toPath(), "application/x-sqlite3");

        auth.apply(request);
        ContentResponse response = request.send();
        int responseStatus = response.getStatus();
        httpClient.stop();

        if ( (responseStatus == HttpStatus.CREATED_201) || (responseStatus == HttpStatus.ACCEPTED_202)
            || (responseStatus == HttpStatus.OK_200)) {
            return true;
        }
    } catch (Exception e) {
        logger.error("HttpClient error", e);
        return false;
    }

    return false;
```