**DownloadDump – documentation**

/src

### downloadAndProcessDumps.js

-main file

- calls fetchApp.getFilesAllMuseum()

- calls sqliteCode.mainSQLiteFunction()

### fetchApp.js

function getFilesAllMuseum()

* Calls download(file) for all museums

Function download(fileList, callback)

* Main function
* Downloads files and uzip

### sqliteCode\_functions.js

async function createDatabase(file, source, mainTable, collection)

* opens database. Create new one if it doesn't exist
* in:
  + file (string, name of file)
  + source (string, "corema" or "musit", indicates which database the data comes from)
  + mainTable (string, name of sqlite-table with musitdata)
  + collection (string, name of collection)
* out: a sqlite database-object
* is called in runCoremaStitch() og runMusitCoremaStitch()

async function deleteFromTable (db, tableName)

* deletes data from tables in database. Is not used when tables are only updated with new data from dump-file
* in:
  + db (sqlite-object?, sqlite-database containing one organismgroup)
  + tableName (string, name of table to be deleted in database)
* is called in runCoremaStitch() og runMusitCoremaStitch()

async function makeNewMycFile()

* concatenate fungus- and lichen-musit-dumpfiles to stitch musit data to fungi-lichen-corema-data
* out: a txt-file with data from both fungus-musit-dump and lichen-musit-dump
* is called in runMusitCoremaStich()

async function makeFileOnlyNew(db, tableName, dumpFolder, source)

* Finds latest time of change in record in table. Reads dumpfile, singels out records with newer modified-times and puts them into new file
* in:
  + db (string, name of sqlite-database)
  + tableName (string, name of table in sqlite-database)
  + dumpFolder (string, name of folder with musit-dump-files)
  + source (string, "musit" or "corema")
* out: txt-file with only records that where modified since last time
* is called in runCoremaStitch() og runMusitCoremaStitch()

async function makeOtherFileOnlyNew(tableName, coremaFolder)

* makes file for corema-archive-dump-files containing only records for which the corresponding record in the corresponding simpledwc-file (main file for corema-dumps) are modified since last time
* in:
  + tableName (string, name of table in sqlite-database)
  + coremaFolder (string, name of folder with corema-dump-files for a specific collection/org.-group)
* out: txt-file with only records that where modified since last time
* is called in runCoremaStitch() og runMusitCoremaStitch()

async function changeEncoding(infile)

* removes double quotes and part of haeder containing ":" in musit- or corema-dumpfile, changes encoding to utf8 (from possibly utf-8-bom)
* in: infile (string, name of musit-dumpfile)
* is called in runCoremaStitch() og runMusitCoremaStitch()

async function fillTable(db, tablename, filename, update)

* fills table in database (either deleting records to be updated, and then update, or fill entire table after data has been removed)
* in:
  + db (sqlite-database containing one organismgroup)
  + tableName (string, name of table to fill data into)
  + filename (string, name of dumpfile where data is fetched)
* is called in runCoremaStitch() og runMusitCoremaStitch()

const createViewMusitRel = (musitFile, double)

* creates sqlite-view combining musit-table and relatedresourceID in corema, to be able to match between them
* in:
  + musitFile (string, name of main-musit-table in sqlite database)
  + double (string, “yes” (meaning we are working with corema, where fungi and lichens are in the same dump-files) or “no”)
* is called in runMusitCoremaStitch()

const createViewCoremaFields = ()

* creates sqlite-view combining all corema-dump files (darwin core archive files)
* is called in runMusitCoremaStitch()

const itemToArraysOnSameLine = (rows, basedOn)

* puts item-information for several items in one line, resulting in one line for each object for coremadata.
* in:
  + rows (JSON object? or similar, coming from sqlite-query)
  + basedOn (string, “corema” or “musit”: whether corema or musit is starting point for stitching (only for fungi and lichens?))
* out: processedRows (array with objects (?))
* is called in runMusitCoremaStitch()

const removeCoremaDuplicates = (processedRows)

* finds duplicate entries in corema of one musit-entry, concatenate them, and remove superfluous row (happens when erroneously one musit-object is entered twice into corema, as accessions. should be items)
* in: processedRows (array of objects?)
* out: processedRows (array of objects?) (without duplicates, and sorted)
* is called in runMusitCoremaStitch()

function mapElementToColumns(fieldNames)

* turns array of objects containing records into text… or something that easily converts text (to write to file)
* in: fieldNames (array?, keys from one object containing a record, i.e. headers in final text file)
* is called in runMusitCoremaStitch()

async function findLastModified(db, tableName)

* finds highest date from last-modified-column in a sqlite-table
* in:
  + db: sqlite database
  + tableName: string (name of table in sqlite database)
* out: latestModified (string (?), a date)

async function runCoremaStitch(collection, coremaFile, coremaFolder, outfile, update)

* main-function that opens db, query database, stitch data, and writes to file for collections that only have data in corema (e.g. birds)
* in:
  + collection (string, name of collection)
  + coremaFile (string, here “no file”; is passed on to createDatabase() for mainfile, and for strict corema-data there is no mainfile)
  + coremaFolder (string, name of folder with corema dwc-archive files for that collection)
  + outfile (string, name of final stitched file)
  + update (string, either 'update' to update db-tables with only new and changed posts, or 'empty\_fill' to empty tables and fill from scratch)
* out: outfile (string, the final stitched file)
* is called in main-function

async function runMusitCoremaStitch(collection, musitFile, coremaFolder, outfile, basedOn, update)

* opens db, query database, stitch data, and writes to file for collections that have data in both musit and corema
* in:
  + collection (string, name of collection)
  + musitFile (string, name of musit-dump-file)
  + coremaFolder (string, name of folder with corema dwc-archive files for that collection)
  + outfile (string, name of final stitched file)
  + basedOn (string, either 'corema' or 'musit', deciding which select (musit data or corema-with-musit-data) to choose)
  + update (string, either 'update' to update db-tables with only new and changed posts, or 'empty\_fill' to empty tabels and fill from scratch)
* out: outfile (the final stitched file)
* is called in main-function