

EntryPoint
+ / db :Database
doGet () :void
doPost () :void
+ init () :void

initialisiert

Database
- dataSource :DataSource
- / schemaBuilder :SchemaBuilder
+ con :Connection
- / loader :Load
+ getDataSource (dataSource :DataSource)
+ setDataSource () :void
+ getSchemaBuilder (schemaBuilder :SchemaBuilder)
+ setSchemaBuilder () :void
+ getCon (con :Connection)
+ setCon () :void

ConnectorJDBC
- URL :String = "jdbc:sqlserver://lab-44-db1.lab.cas.de:14334;integratedSecurity=true"
- con :Connection
+ getCon () :Connection
+ setCon (in con :Connection) :void

VerbindungAufbautMit

SchemaBuilder
+ createSchema (in ds :DataSource) :void

verwendetZumBefüllenDerDB

Load
+ loadMainData (in con :Connection) :void
+ loadTownData (in con :Connection) :void
+ loadCountryData (in con :Connection) :void
+ loadSysUserData (in con :Connection) :void
+ loadSysGroupData (in con :Connection) :void
+ dropData (in con :Connection) :void
+ createIndexData (in con :Connection) :void
+ createIndexCountry (in con :Connection) :void
+ createIndexTown (in con :Connection) :void
+ createIndexUser (in con :Connection) :void
+ createIndexGroup (in con :Connection) :void
+ createIndexUserGroup (in con :Connection) :void

verwendetZumBefüllenDerDB

JerseyServer
- / queryBuilder :QueryBuilder
- / jdbcConnectorMSSQL :ConnectorJDBC
- / transform :Transform
- / loader :Load
- / logic :Logic
+ EPATH :String
+ TPATH :String
+ fullETL (in urlInfo :UrlInfo, in headers :HttpHeaders) :String
+ extract (in urlInfo :UrlInfo, in headers :HttpHeaders) :String
+ transform (in urlInfo :UrlInfo, in headers :HttpHeaders) :String
+ load (in urlInfo :UrlInfo, in headers :HttpHeaders) :String
+ post (in content :JSONObject, in urlInfo :UrlInfo, in headers :HttpHeaders) :JSONObject
+ userInDB (in urlInfo :UrlInfo, in headers :HttpHeaders, in content :String) :String
+ updateFromOrigin (urlInfo :UrlInfo, headers :HttpHeaders,) :String

erzeugtAbfrageMit

ExtrahiertMit

Logik
- whereBin :String = ""
- groupRelation :Map<String, ArrayList<String>>
- sysGroup :Map<String, String>
- EPATH :String
- TPATH :String
- groupHistory :Map<String, ArrayList<String[]>>
- factor :int = 10000
- startFinal :long
- endFinal :long
+ buildQuery (in con :Connection, in json :JSONObject) :JSONObject
- buildWhereClause (in data :ArrayList<String>) :String
- createWherePersonID (in json :JSONObject) :ArrayList<String>
- createWhereDateRange (in json :JSONObject) :ArrayList<String>
- createWhereIsEmployee (in json :JSONObject) :ArrayList<String>
- createWhereIsCompany (in json :JSONObject) :ArrayList<String>
- createWhereIsCompanyAndContact (in json :JSONObject) :ArrayList<String>
- createWhereIsContact (in json :JSONObject) :ArrayList<String>
- createWhereIgnorePerson (in json :JSONObject, in con :Connection) :ArrayList<String>
- createWhereTown (in json :JSONObject, in con :Connection) :ArrayList<String>
- createWhereCountry (in json :JSONObject, in con :Connection) :ArrayList<String>
- createWhereGroup (in json :JSONObject, in con :Connection) :ArrayList<String>
+ getDateDiff (in date1 :Date, in date2 :Date, in timeUnit :TimeUnit) :long

QueryBuilder
- csvBuilder :CSVBuilder
+ createNameDictonarie (in con :Connection) :void
+ createTown1Dictonarie (in con :Connection) :void
+ createCountry1Dictonarie (in con :Connection) :void
+ createGroupDictonarie (in con :Connection) :void
+ createSysUserDictonarie (in con :Connection) :void
+ buildObjektData (in con :Connection, in table :String, in type :String, in n :String) :void
+ buildDataWhichGoesOverADay (in con :Connection, in table :String, in type :String, in n :String) :void
+ buildAppointmentTimeShifts (in con :Connection, in n :String) :void
+ checkUserInDatabase (in con :Connection, in username :String) :String
+ buildGetGroupChanges (in con :Connection) :void
+ buildUserGGUID (in con :Connection) :void
+ buildGroupGGUID (in con :Connection) :void
+ buildSysUserAndGroupWithAddress (in con :Connection) :void
+ buildSysUserWithAddress (in con :Connection) :void
+ buildSysGroupRelation (in con :Connection) :void
+ build0RelData (in con :Connection, in table :String, in date :String, in type :String) :void
+ build0RelAppointmentMultipleDays (in con :Connection) :void
+ build0RelAppointmentDayShifts (in con :Connection) :void
+ buildFromUpdate (con :Connection, gguid :String) :void

TransformiertMit

Transform
- / csvBuilder :CSVBuilder
- / csv :CSVReaderWriter
+ transformMultipleAppointment () :void
+ transformMultipleAppointmentORel () :void
+ reuniteTables () :void
+ replaceTownAndCountry (in inputPath :String, in outputPath :String) :void
+ transformGroupHistory () :void
+ replaceTownAndCountrySysUserAndGroupWithAddress () :void
+ replaceTownAndCountrySysUserWithAddress () :void
+ addAddressToCSV (in index :Integer, in inputPath :String, in outputPath :String) :void

schreibtUndLiestCSVDateienMit

erstelltCSVDateienMit

erstelltCSVDateienMit

CSVReaderWriter
+ writeDataToCSV (in data :ArrayList<String[]>, in path :String) :void
+ readFromCSVFileAsStringArray (in path :String) :ArrayList<String[]>
+ readFromCSVFileAsIntegerArray (in path :String) :ArrayList<Integer[]>
+ readFromCSVFileAsStringStringMap (in path :String) :Map<String, String>
+ readFromCSVFileAsStringStringMap10 (in path :String) :Map<String, String>
+ readFromCSVFileAsStringIntegerMap (in keyIndex :Integer, in path :String) :Map<String, Integer>
+ readFromCSVFileAsIntegerStringMap (in keyIndex :Integer, in path :String) :Map<Integer, String>
+ readFromCSVFileAsStringStringArrayList (in path :String) :Map<String, ArrayList<String>>
+ readFromCSVFileAsStringStringArrayListArray (in path :String) :Map<String, ArrayList<String[]>>
+ readFromCSVFileAsStringStringArrayMap (in keyIndex :Integer, in path :String) :Map<String, String[]>

CSVBuilder
+ buildDictonarie (in fileName :String, in data :ArrayList<String>) :void
+ buildDictonarieWithOID (in fileName :String, in data :ArrayList<String[]>) :void
+ buildTableRelation (in data :ArrayList<String[]>, in fileName :String) :void
+ buildTransformMultipleAppointment (in data :ArrayList<String[]>) :void
+ buildTransformMultipleAppointmentORel (in data :ArrayList<String[]>) :void
+ buildGroupHistory (in data :ArrayList<String[]>) :void
+ buildGGUIDDictonarie (in fileName :String, in data :ArrayList<String[]>) :void
+ buildSysUserWithAddress (in fileName :String, in data :ArrayList<String[]>) :void