

com.novetta.clavin.gazetteer

BasicGeoName

•

getAdminCode

getAdminCode

getAdminCode

getAdminCode

getLongitude

getLongitude

getParentId

getPopulation

getPreferredName

buildAncestryKey

getAdminCode

getAdminName

getElevation

getElevation

getMultiAncestryKey

getMultiAncestryKey

hashCode

isAncestorOf

isAncestryResolved

equals

getAdminCode

getAncestryKey

getFeatureClass

getGeoNameID

getName

getGeoNameID

isDescendantOf

isTopLevelTerritory

parseGeoName

getAdminCode

getAdminLevel

getAsciiName

getFeatureCode

getLatitude

getParent

getTimezone

setParent

toString

LazyAncestryGeoName

•

equals

getAdmin1Code

getAdmin2Code

getAdmin3Code

getAdmin4Code

getAdminCountryCode

getAlternateNames

getAncestryKey

getAsciiName

getGeoNameID

getLatitude

getLongitude

getElevation

getMultiAncestryCode

getParentId

getPopulation

getPreferredName

getFeatureClass

getName

getAncestryCode

hashCode

isAncestorOf

getFeatureCode

getParent

isAncestryResolved

isTopLevelAdminDivision

isTopLevelTerritory

getGazetteerRecord

getParentAncestryKey

getTimezone

isDescendantOf

setParent

GeoName

getAdmin1Code

getAdmin2Code

getAdmin3Code

getAdmin4Code

getAdminCountryCode

getAncestryKey

getAsciiName

getFeatureClass

getFeatureCode

getGazetteerRecord

getGeoNameID

getName

getLatitude

getLongitude

getParent

getTimezone

isAncestorOf

isDescendantOf

setParent

BasicGeoNameTest

•

setUp

testAllAncestryRoutes

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

testAllAncestryKeys

LazyAncestryGeoNameTest

•

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

Mocking

FeatureCodeBuilder

com.novetta.clavin.gazetteer.query

LuceneGazetteerTest

•

setUp

testBorderCases

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

testFilterDups

QueryBuilder

•

addAdminCodes

addCityCodes

addCountryCodes

addFeatureCodes

addParentIds

ancestryMode

build

clearFeatureCodes

clearParentIds

featureCodes

filterDups

location

maxResults

parentIds

fuzzyMode

removeCityCodes

removeCountryCodes

toString

includeHorizontal

includeVertical

includeVertical

includeVertical

LuceneGazetteer

•

resolveParents

buildFilters

getGeoName

executeQuery

loadAncestry

getClosestLocations

sanitizeQueryText

getGeoName

getGeoName

getGeoName

getGeoName

GazetteerQuery

•

getGeoName

getGeoName

getFeatureCode

getFuzzyMode

getOccurrence

getParentIds

getGeoName

getGeoName

getGeoName

getGeoName

UniqueFuzzyScoringRewrite

•

build

getMaxSize

addClause

getTopLevelBuilder

getGeoName

loadAncestry

Gazetteer

•

getGeoName

loadAncestry

ClavinLocationResolverHeuristicsTest

•

setUp

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

testHeuristicsDC

ClavinLocationResolverTest

•

setUp

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

testIsDemonym

ResolvedLocation

•

getGeoName

hashCode

equals

getLocation

toString

ClavinLocationResolver

•

getGazetteer

pickBestCandidates

generateCandidates

isDemonym

resolveLocations

LocationResolver

•

getGazetteer

pickBestCandidates

LuceneLocationResolver

•

getGazetteer

pickBestCandidates

com.novetta.clavin.resolver.multipart

MatchedLocation

•

getMatchCount

getMatches

equals

getMatch

hashCode

toString

MultipartLocationName

•

getCity

getState

hashCode

equals

getCountry

toString

MultipartLocationResolverTest

•

setUpClass

verifyCity

verifyCountry

parameters

DefaultScorer

•

score

parameters

setUpClass

SearchResult

•

getBestLocation

Scorer

com.novetta.clavin.extractor

AdaptNlpExtractor

•

extractLocationNames

Entity

•

getText

getType

setConfidence

setEndPos

setType

LocationOccurrence

•

getText

hashCode

equals

TextBody

•

getText

setTest

toString

ApacheExtractor

•

extractLocationNames

testNullInput

ApacheExtractorTest

•

testNullInput

LocationOccurrenceTest

•

testEquals

testHashCode

LocationExtractor

•

extractLocationNames

com.novetta.clavin.index

IndexDirectoryBuilder

main

getOptions

indexGeoName

loadAlternateNames

logUnresolved

resolveAncestry

resolveUnresolved

printHelp

buildIndex

•

testNullInput

testTF

BinarySimilarity

•

idf

tf

normalize

WhitespaceLowerCaseTokenizer

•

normalize

WhitespaceLowerCaseAnalyzer

•

normalize

com.novetta.clavin.util

DamerauLevenshteinTest

•

testDistance

testDistance

testDistance

Null

•

hashCode

equals

toString

ListUtilsTest

•

testUtils

testUtils

testUtils

ListUtils

•

chunkUtils

TextUtils

DamerauLevenshtein

•

testDistance

testDistance

testDistance

InfiniteCharArray

•

get

remainder

com.novetta.clavin

GeoParser

•

parse

AllTestSuite

ClavinException

•

testException

testException

WorkflowDemo

•

testWorkflow

testWorkflow