

BasicGeoName

<code>*</code> <code>org.ahoyahoy</code>	<code>getAdmin3Code</code> <code>String</code>	<code>getAncestryKey</code> <code>String</code>	<code>getFeatureCode</code> <code>String</code>	<code>getLongitude</code> <code>String</code>	<code>getParentId</code> <code>none</code>	<code>getPopulation</code> <code>none</code>	<code>getRefName</code> <code>none</code>	<code>getRefParentCode</code> <code>none</code>
<code>buildAncestryKey</code> <code>String</code>	<code>getAdmin4Code</code> <code>String</code>	<code>getAsciiName</code> <code>String</code>	<code>getGazetteerRecord</code> <code>String</code>	<code>getModificationDate</code> <code>String</code>	<code>getRefParentCode</code> <code>none</code>	<code>isAncestorOf</code> <code>none</code>	<code>isAncestryResolved</code> <code>none</code>	<code>isDescendantOf</code> <code>none</code>
<code>equals</code> <code>String</code>	<code>getAdminLevel</code> <code>String</code>	<code>getGeoFeatureCode</code> <code>String</code>	<code>getRefParentCode</code> <code>String</code>	<code>getName</code> <code>String</code>	<code>getTimezone</code> <code>none</code>	<code>getRefParentCode</code> <code>none</code>	<code>getRefParentCode</code> <code>org.ahoyahoy</code>	<code>parseGeoName</code> <code>none</code>
<code>getAdmin1Code</code> <code>String</code>	<code>getRefParentCode</code> <code>String</code>	<code>getElevation</code> <code>String</code>	<code>getGeoNameId</code> <code>String</code>	<code>getParent</code> <code>String</code>	<code>hashCode</code> <code>none</code>	<code>getRefParentCode</code> <code>none</code>	<code>setParent</code> <code>org.ahoyahoy</code>	<code>toString</code> <code>none</code>
<code>getAdmin2Code</code> <code>String</code>	<code>getAlternateNames</code> <code>String</code>	<code>getFeatureClass</code> <code>String</code>	<code>getLatitude</code> <code>String</code>	<code>getParentAncestryKey</code> <code>String</code>				

LazyAncestryGeoName					GeoName			
.	equals	getAdmin1Code	getAdmin2Code	getAdmin3Code	getAdmin1Code	getAdmin2Code	getAdmin3Code	getAdmin4Code
none	none	none	none	none	none	none	none	none
getAdmin4Code	getAlternateCountryCodes	getAlternateNames	getAncestryKey	getAsciiName	getAlternateCountryCodes	getAlternateNames	getAncestryKey	getAsciiName
none	none	none	none	none	none	none	none	none
getCountrySourceModel	getCountrySourceModelKey	getGeoNameID	getLatitude	getLongitude	getCountrySourceModel	getCountrySourceModelKey	getGeoNameID	getLatitude
none	none	none	none	none	none	none	none	none
getElevation	getElevationData	getParentId	getPopulation	getPreferredName	getElevation	getElevationData	getParentId	getPopulation
none	none	none	none	none	none	none	none	none
getFeatureClass	getName	getHcrsCountryCode	hashCode	isAncestorOf	getFeatureClass	getName	getHcrsCountryCode	hashCode
none	none	none	none	none	none	none	none	none
getFeatureCode	getParent	getHcrsCountryName	isAncestorOf	isTopLevelAdminDivision	getFeatureCode	getParent	getHcrsCountryName	isAncestorOf
none	none	none	none		none	none	none	none
getGeoNameRecord	getHcrsCountryKey	getTimezone	isDescendantOf	isTopLevelTerritory	getGeoNameRecord	getHcrsCountryKey	getTimezone	isDescendantOf
none	none	none	none		none	none	none	none
				setParent				setParent

BasicGeoNameTest

setUp attributes: *	tearDown attributes: *	testAllAttributes attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *

LazyAncestryGeoNameTest

setUp attributes: *	tearDown attributes: *	testAllAttributes attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *
testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *	testTerritoryAncestryKeys attributes: *

[illegible]

The diagram illustrates the components and interactions of the Gazetteer API. It is divided into three main sections:

- LuceneGazetteer** (Left):
 - executeQuery**: Contains `org.apache.lucene.index` and `org.apache.lucene.search`.
 - getClosestLocations**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getGeoName**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - resolveParents**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - buildFilters**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - healthChecker**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - sanitizeQueryText**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
- GazetteerQuery** (Middle):
 - getDocument**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getFields**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getLocations**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getParents**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getScore**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getTerms**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getTypes**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getValues**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getWeights**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getZones**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
- UniqueFuzzyScoringRewrite** (Right):
 - build**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - addClause**: Contains `org.apache.lucene` and `org.apache.lucene.index`.
 - getTopLevelBuilder**: Contains `org.apache.lucene` and `org.apache.lucene.index`.

Arrows indicate the flow of data and dependencies between these components.

com.novetta.clavin.resolver

ClavinLocationResolverHeuristicsTest

- setUp
- resolveHeuristics
- testBorderCases
- testHeuristicsIllinois
- testHeuristicsOntario
- testHeuristicsDC
- testHeuristicsWA

ResolvedLocation

- getGazetteer
- pickBestCandidates
- generateAllCombos
- isDemonym
- resolveLocations
- toString

ClavinLocationResolverTest

- setUp
- testBorderCases
- testDemonym
- testResolveLocations

LuceneLocationResolver

- resolveLocations

LocationResolver

- ResolvedLocationTest

The diagram illustrates the relationships between classes in the Location API. It is organized into four main colored regions:

- MatchedLocation (Yellow):** Contains methods like `getMatchCount`, `getMatches`, `equals`, `isFullySpecified`, `getMatch`, `hashCode`, and `toString`.
- ResolvedMultipartLocation (Green):** Contains methods like `getCountry`, `getState`, `equals`, `hashCode`, `toString`, and `getCity`.
- DefaultScorer (Green):** Contains methods like `getScore`, `score`, and `toString`.
- MultipartLocationResolverTest (Blue):** Contains methods like `parameters`, `verifyCity`, and `verifyLocation`.

Other components shown include **SearchResult** and **Scorer**.

The diagram illustrates the structure of a Java project, organized into three main packages: `com.novetta.clavin.extractor`, `com.novetta.clavin.index`, and `com.novetta.clavin.indexer`.

com.novetta.clavin.extractor

- AdaptNlpExtractor** (Orange box):
 - Methods: `extractLocationNames`
- Entity** (Green box):
 - Methods: `getText`, `getType`, `setConfidence`, `setGeoName`, `setText`, `setEndPos`, `setType`
 - Attributes: `none`
- LocationOccurrence** (Green box):
 - Methods: `getText`, `equals`, `hashCode`
 - Attributes: `none`
- TextBody** (Green box):
 - Methods: `getText`, `setText`, `toString`
 - Attributes: `none`
- ApacheExtractor** (Pink box):
 - Methods: `extractLocationNames`
- ApacheExtractorTest** (Blue box):
 - Methods: `testNullInput`
- LocationOccurrenceTest** (Blue box):
 - Methods: `testEquals`, `testHashCode`
- AdaptNlpExtractorTest** (Blue box):
 - Methods: `testExtractLocationNames`
- LocationExtractor** (Green box):
 - Methods: `extractLocationNames`

com.novetta.clavin.index

- IndexDirectoryBuilder** (Orange box):
 - Methods: `main`, `getOptions`, `buildIndex`, `printHelp`
 - Attributes: `none`
- BinarySimilarity** (Orange box):
 - Methods: `idf`, `tf`
 - Attributes: `none`
- WhitespacLowerCaseTokenizer** (Orange box):
 - Methods: `normalize`
 - Attributes: `none`
- BinarySimilarityTest** (Orange box):
 - Methods: `testTF`
 - Attributes: `none`
- WhitespacLowerCaseAnalyzer** (Orange box):
 - Methods: `analyze`
 - Attributes: `none`

The screenshot shows the 'com.novetta.clavin.util' module in the IntelliJ IDE. The module is highlighted in orange. It contains several packages, each represented by a colored box with its name and a small icon. The packages are arranged in a grid-like structure:

- DamerauLevenshteinTest** (Orange box)
- Null** (Green box)
- ListUtilsTest** (Blue box)
- GeoParser** (Yellow box)
- ClavinException** (Green box)
- ListUtils** (Blue box)
- TextUtils** (Blue box)
- WorkflowDemo** (Blue box)

Each package box contains a small icon representing the package's content. The packages are arranged in a grid-like structure, with some packages having sub-packages or methods listed below them.