

CombinatorialQueries Snippet

Java Code Snippet

```
System.out.println("Example double restriction query with additional "
    +"application of sort criteria and restricted return values.");
// Declare the service...
LexBIGServiceImpl lbs = LexBIGServiceImpl.defaultInstance();

// Start with an unconstrained set of all codes for the vocabulary
CodingSchemeVersionOrTag csvt = new CodingSchemeVersionOrTag();
csvt.setVersion(VERSION2);
CodedNodeSet cns = lbs.getCodingSchemeConcepts(SAMPLE_SCHEME2, csvt);

// Constrain to concepts with designations (assigned text presentations
// that contain text that sounds like 'Short Saphenous Vein'
cns = cns.restrictToMatchingDesignations(
    "Short Safinus Vane",
    SearchDesignationOption.ALL,
    MatchAlgorithms.DoubleMetaphoneLuceneQuery.toString(),
    null);

// Further restrict the results to concepts with a semantic type of
// 'Anatomical Structure'
cns = cns.restrictToMatchingProperties(
    Constructors.createLocalNameList("Semantic_Type"),
    null, "Anatomical Structure",
    "exactMatch",
    null);

// Indicate that the resulting list should be sorted with the best
// results first and then sorted by code if there is a tie.
SortOptionList sortCriteria = Constructors.createSortOptionList(
    new String[] {"matchToQuery", "code"});

// Indicate to return only the assigned UMLS_CUI and
// textualPresentation properties.
LocalNameList restrictTo = ConvenienceMethods.createLocalNameList(
    new String[] {"UMLS_CUI", "textualPresentation"} );

// Still nothing computed yet.
// Perform the query && resolve the sorted/filtered list with a
// maximum of 6 items returned.
ResolvedConceptReferenceList list = cns.resolveToList(
    sortCriteria, restrictTo, null, 6);
// Print the results
ResolvedConceptReference[] rcr = list.getResolvedConceptReference();
for (ResolvedConceptReference rc : rcr)
{
    System.out.println("Resolved Concept: " + rc.getConceptCode());
}
```