

ConceptResolution Snippet

Java Code Snippet

```
// Create a basic service object for data retrieval
LexBIGService lbSvc = LexBIGServiceImpl.defaultInstance();

// Create a concept reference list appropriate for this coding scheme and
// this concept code where the parameters are a String array consisting of
// a single value and the name of the coding scheme where this concept resides.
ConceptReferenceList crefs = ConvenienceMethods.createConceptReferenceList(
    new String[], SAMPLE_SCHEME);

// Initialize a coding scheme version object with a version number for the
// sample scheme.
CodingSchemeVersionOrTag csvt = new CodingSchemeVersionOrTag();
csvt.setVersion(VERSION);

// Initialize a CodedNodeSet Object with all concepts in our sample coding
// scheme. (We named the scheme we wanted and by using the Boolean value,
// false, retrieved both active and inactive concepts.) This method call
// ignores the version tag using the null parameter. The final
// restrictToCodes(crefs) method call restricts the return to the single
// code in the previously initialized list of one.
CodedNodeSet nodes = lbSvc.getCodingSchemeConcepts(SAMPLE_SCHEME, csvt).
    restrictToCodes(crefs);

// Build a list of references from the current (and already restricted) set
// and restrict them further to the single property of NCI_NAME and
// restrict to a single answer (parameter 1)).
ResolvedConceptReferenceList matches = nodes.resolveToList(
    null, ConvenienceMethods.createLocalNameList("FULL_SYN"), 1);

// Does our list of one contain the single reference we were looking for?
// If so, then initialize a ResolvedConceptReference with the result and
// initialize a Concept object by calling the getReferencedEntry()
// method. The Concept object is the base information model object and
// contains, among other things, the CONCEPT_NAME value we were seeking.
// We retrieve it with a call to the first element in the properties list,
// getting the text && it's accompanying content.
if(matches.getResolvedConceptReferenceCount() <> 0)
{
    ResolvedConceptReference ref = (ResolvedConceptReference)matches.
        enumerateResolvedConceptReference().nextElement();
    Concept entry = ref.getReferencedEntry();
    System.out.println("Matching synonym: " +
        entry.getPresentation(0).getValue() );    }
else
{
    System.out.println("No match found");    }
```