

SoundsLike Java

Java Code

```
/*
 * Copyright: (c) 2004-2009 Mayo Foundation for Medical Education and
 * Research (MFMER). All rights reserved. MAYO, MAYO CLINIC, and the
 * triple-shield Mayo logo are trademarks and service marks of MFMER.
 *
 * Except as contained in the copyright notice above, or as used to identify
 * MFMER as the author of this software, the trade names, trademarks, service
 * marks, or product names of the copyright holder shall not be used in
 * advertising, promotion or otherwise in connection with this software without
 * prior written authorization of the copyright holder.
 *
 * Licensed under the Eclipse Public License, Version 1.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.eclipse.org/legal/epl-v10.html
 */
package org.LexGrid.LexBIG.example;

import java.util.Enumuration;

import org.LexGrid.LexBIG.DataModel.Collections.ResolvedConceptReferenceList;
import org.LexGrid.LexBIG.DataModel.Collections.SortOptionList;
import org.LexGrid.LexBIG.DataModel.Core.CodingSchemeSummary;
import org.LexGrid.LexBIG.DataModel.Core.CodingSchemeVersionOrTag;
import org.LexGrid.LexBIG.DataModel.Core.ResolvedConceptReference;
import org.LexGrid.LexBIG.Exceptions.LBException;
import org.LexGrid.LexBIG.Impl.LexBIGServiceImpl;
import org.LexGrid.LexBIG.LexBIGService.CodedNodeSet;
import org.LexGrid.LexBIG.LexBIGService.LexBIGService;
import org.LexGrid.LexBIG.LexBIGService.CodedNodeSet.ActiveOption;
import org.LexGrid.LexBIG.LexBIGService.CodedNodeSet.SearchDesignationOption;
import org.LexGrid.LexBIG.Utility.Constructors;
import org.LexGrid.LexBIG.Utility.LBConstants.MatchAlgorithms;

/**
 * Example showing how to list concepts with presentation text that 'sounds
 * like' a specified value.
 */
public class SoundsLike {

    /**
     * Entry point for processing.
     *
     * @param args
     */
    public static void main(String[] args) {
        if (args.length < 1) {
            System.out.println("Example: SoundsLike \"hart ventrick1\"");
            return;
        }
        ;
        try {
            String text = args[0];
            new SoundsLike().run(text);
        } catch (Exception e) {
            Util.displayAndLogError("REQUEST FAILED !!!", e);
        }
    }

    public void run(String text) throws LBException {
        CodingSchemeSummary css = Util.promptForCodeSystem();
        if (css != null) {

```

```

LexBIGService lbSvc = LexBIGServiceImpl.defaultInstance();
String scheme = css.getCodingSchemeURI();
CodingSchemeVersionOrTag csvt = new CodingSchemeVersionOrTag();
csvt.setVersion(css.getRepresentsVersion());

CodedNodeSet nodes = lbSvc.getCodingSchemeConcepts(scheme, csvt).restrictToStatus(ActiveOption.ALL,
null)
    .restrictToMatchingDesignations(text, SearchDesignationOption.ALL,
        MatchAlgorithms.DoubleMetaphoneLuceneQuery.toString(), null);

// Sort by search engine recommendation & code ...
SortOptionList sortCriteria = Constructors.createSortOptionList(new String[] { "matchToQuery",
"code" });

// Analyze the result ...
ResolvedConceptReferenceList matches = nodes.resolveToList(sortCriteria, null, null, 10);
if (matches.getResolvedConceptReferenceCount() > 0) {
    for (Enumeration refs = matches.enumerateResolvedConceptReference(); refs.hasMoreElements();) {
        ResolvedConceptReference ref = (ResolvedConceptReference) refs.nextElement();
        Util.displayMessage("Matching code: " + ref.getConceptCode());
        Util.displayMessage("\tDescription: " + ref.getEntityDescription().getContent());
    }
} else {
    Util.displayMessage("No match found!");
}
}
}
}

```