

```
/*
  JSPWiki - a JSP-based WikiWiki clone.

  Licensed to the Apache Software Foundation (ASF) under one
  or more contributor license agreements. See the NOTICE file
  distributed with this work for additional information
  regarding copyright ownership. The ASF licenses this file
  to you under the Apache License, Version 2.0 (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.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
  software distributed under the License is distributed on an
  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
  KIND, either express or implied. See the License for the
  specific language governing permissions and limitations
  under the License.
*/
package com.ecyrd.jspwiki.plugin;

import java.util.Iterator;
import java.util.Map;

import org.apache.log4j.Logger;

import com.ecyrd.jspwiki.WikiContext;

/**
 * Provides a list or table of InterWiki links.
 * <p>
 * Parameters:
 * </p>
 * <ul>
```

```

* <li><b>type</b> - The type of output: acceptable values are 'text', 'ulist'
* (unordered list), 'olist' or 'table'. Default = 'text'.</li>
* <li><b>tableTitle</b> - The title of the table, when specified.</li>
* <li><b>separator</b> - The separator string to be used between two entries in
* the output of type 'text'. Default = "<br />".</li>
* </ul>
*
* @author devjunkie@nukleos.com
* @version 2009-03-16
*/
public class InterWikiLinksPlugin implements WikiPlugin
{
    private static enum OutputTypes
    {
        TEXT_OUTPUT, U_LIST_OUTPUT, O_LIST_OUTPUT, TABLE_OUTPUT
    };

    private static Logger log = Logger.getLogger( InterWikiLinksPlugin.class );

    /** Parameter name for setting the type. */
    public static final String PARAM_TYPE = "type";

    /** Parameter name for setting the title. */
    public static final String PARAM_TABLE_TITLE = "tableTitle";

    /** Parameter name for setting the separator. */
    public static final String PARAM_SEPARATOR = "separator";

    private OutputTypes default_outputtype = OutputTypes.TEXT_OUTPUT;

    private String default_tableTitle = "Interwiki Links Table";

    private String default_separator = "<br />";

    private String crlf = System.getProperty( "line.separator" );

```

```
/**
 * {@inheritDoc}
 */
public String execute( WikiContext context, Map params ) throws PluginException
{
    try
    {
        StringBuffer sb = new StringBuffer();
        OutputTypes type = default_outputtype;

        // Get parameters, if any
        String outputtype = (String) params.get( PARAM_TYPE );
        String tableTitle = (String) params.get( PARAM_TABLE_TITLE );
        String separator = (String) params.get( PARAM_SEPARATOR );

        // Handle default cases
        if( null == outputtype )
        {
            type = OutputTypes.TEXT_OUTPUT;
        }
        else if( "TEXT".equals( outputtype.toUpperCase() ) )
        {
            type = OutputTypes.TEXT_OUTPUT;
        }
        else if( "ULIST".equals( outputtype.toUpperCase() ) )
        {
            type = OutputTypes.U_LIST_OUTPUT;
        }
        else if( "OLIST".equals( outputtype.toUpperCase() ) )
        {
            type = OutputTypes.O_LIST_OUTPUT;
        }
        else if( "TABLE".equals( outputtype.toUpperCase() ) )
        {

```

```

    type = OutputTypes.TABLE_OUTPUT;
}

if( (tableTitle == null) || "".equals( tableTitle ) )
{
    tableTitle = default_tableTitle;
}

if( (separator == null) || "".equals( separator ) )
{
    separator = default_separator;
}

// Create output buffer
log.debug( "TITLE: " + tableTitle + ", SEPARATOR: " + separator + ", TYPE: " + type.name() );

sb.append( "<div class=\"interwikilinks\">" + crlf );

if( type.equals( OutputTypes.TEXT_OUTPUT ) )
{
    sb.append( crlf + "<span id=\"interwikilisttext\">" );

    for( Iterator i = context.getEngine().getAllInterWikiLinks().iterator(); i.hasNext(); )
    {
        String link = (String) i.next();
        sb.append( link );
        sb.append( " == " );
        sb.append( context.getEngine().getInterWikiURL( link ) );
        if( i.hasNext() )
        {
            sb.append( separator );
        }
    }
    sb.append( "</span>" + crlf );
}

```

```

if( type.equals( OutputTypes.U_LIST_OUTPUT ) )
{
    sb.append( generateListOutput( context, "<ul>", "</ul>" ) );
}

if( type.equals( OutputTypes.O_LIST_OUTPUT ) )
{
    sb.append( generateListOutput( context, "<ol>", "</ol>" ) );
}

if( type.equals( OutputTypes.TABLE_OUTPUT ) )
{
    sb.append( "<table><caption>" + tableTitle + "</caption>" + crlf );
    sb.append( "<tbody>" + crlf );

    for( Iterator i = context.getEngine().getAllInterWikiLinks().iterator(); i.hasNext(); )
    {
        String link = (String) i.next();
        sb.append( "<tr><td>" );
        sb.append( link );
        sb.append( "</td><td>" );
        sb.append( context.getEngine().getInterWikiURL( link ) );
        sb.append( "</td></tr>" + crlf );
    }
    sb.append( "</tbody>" + crlf );
    sb.append( "</table>" + crlf );
}
sb.append( crlf + "</div>" + crlf );

return sb.toString();
}
catch( Exception e )
{
    log.error( "Could not construct InterWikiLinksPlugin output", e );
}

```

```

        throw new PluginException( "Unable to construct InterWikiLinksPlugin output (see logs)" );
    }
}

private StringBuffer generateListOutput( WikiContext context, String startTag, String endTag )
{
    StringBuffer sb = new StringBuffer();

    sb.append( startTag + crlf );
    for( Iterator i = context.getEngine().getAllInterWikiLinks().iterator(); i.hasNext(); )
    {
        String link = (String) i.next();
        sb.append( "    <li>" );
        sb.append( link );
        sb.append( " == " );
        sb.append( context.getEngine().getInterWikiURL( link ) );
        sb.append( "    </li>" + crlf );
    }
    return sb.append( endTag + crlf );
}
}

```