

# LexEVS in SpringSource MVC Maven Project

## Contents of this Page

- [Overview](#)
- [Prerequisites](#)
  - [LexEVS Local Runtime Install](#)
  - [Installing Tomcat 6](#)
  - [Installing SpringSource Tool Suite \(Tested with Version 2.7.1\)](#)
  - [Setting up Tomcat 6 in STS](#)
- [Creating a New Spring MVC 3 Project](#)
  - [Small STS Warning That Can be Fixed](#)
  - [Verification](#)
- [Adding LexEVS as a Maven Dependency](#)
  - [Downloading the Required Files](#)
  - [Installing a LexEVS Dependency](#)
    - [JAVA\\_HOME Environment Variable Missing](#)
    - [MVN Commands](#)
    - [Adjusting Scope of JUnit](#)
- [Including LexEVS in the MVC Project](#)
  - [Creating a Test File](#)
  - [Adding a JVM Parameter to Locate the lbconfig.props](#)
- [Integrating LexEVS in a Web Page](#)
  - [Adding a JVM Parameter to Locate the lbconfig.props](#)
  - [Adjusting the Controller](#)
  - [Adjusting the JSP Page](#)
  - [Running the Project](#)
- [Acknowledgement](#)

## Overview

The spring framework has emerged over the last few years as a versatile and efficient support structure for multiple types of Java applications. The MVC component facilitates the development of rich web-driven applications. The SpringSource tool suite(STS) is an Eclipse derivative software package bundling the necessary tools to develop efficiently with the Spring framework. The Spring 3 MVC project is a Maven compliant project and the guide will make use of it to include a LexEVS dependency.

This guide will illustrate how to use a local instance of LexEVS in a Spring 3.0.x MVC project and how to deploy it to a Tomcat 6 server from within STS. It was tested using LexEVS 6.0 and Windows 7. This should be applicable to other environments and future versions of LexEVS, but some parameters, especially paths, might need to be formatted differently.

This guide will NOT explore how to leverage LexEVS programmatically or how to best develop with the Spring framework. The reader is invited to explore the multiple resources on the web that can help in this endeavor.

## Prerequisites

You will need internet access to complete this guide.

## LexEVS Local Runtime Install


Please follow carefully the steps provided in [the guide for installing LexEVS 6.0 Local Runtime](#).

Once all the steps have been followed successfully you should have a functional LexEVS Local Runtime instance.


In order to facilitate subsequent testing, [load and activate](#) the test terminology Automobiles.xml located in the folder {LEXEVS\_HOME} \test\resources\testData\Automobiles.xml.

Make note of your installation folder.

## Installing Tomcat 6

1. Navigate to the [Tomcat 6 download page](#)  and download the appropriate package for your system. The binary core distribution (zip) is the most usual one.
2. Unzip the content of the file in the desired folder and make note of it.

## Installing SpringSource Tool Suite (Tested with Version 2.7.1)

1. Navigate to the [STS download page](#)  to download the software. Please note that you don't have to fill out the form. If you do not wish to fill it all, just check the checkbox to accept the license terms and click on the link "Download Page" below the "Download Now" button.
2. Choose the appropriate version for your environment and install it. Make sure that you have the Java JDK 1.6 (or less ideally 1.5) installed.

3. During setup, make sure to make a note of where you installed the program. During step 4, make sure "Apache Maven 2..." is selected.
4. You can launch STS and select the location of your workspace. You will then be presented with the Welcome page. You can close it with the "x" next to the tab on top left. This will bring a Spring perspective.

## Setting up Tomcat 6 in STS

1. Open STS. It should be enabled by default but make sure you're in the Spring perspective. In the left lower corner should be the server view.
  2. Right-click in the view and choose "New -> Server" in the context menu.
  3. Type "Tomcat" in the "Select the server type:" textbox. Then scroll down in the selectbox below and choose Tomcat v6.0 Server. Click "Next".
  4. On the next screen, browse to your tomcat install folder and select it.
  5. Click on "Finish"
- You should now see an entry named "Tomcat v6.0 Server at localhost" in the server view.

## Creating a New Spring MVC 3 Project

1. Click on the menu File -> New -> Spring Template Project.
  2. Select Spring MVC Project from the list and click Next.
    - a. It will ask you if you want to download file: click Proceed.
  3. Enter a project name. For this guide we will name it "testlex".
  4. Enter the top level package (x.x.x). We will name it "org.test.test".
  5. Click on Finish.
- It might take some time before your back to a responsive Spring perspective. STS will need to download some dependencies for the project. You should then see your project [testlex] in the package explorer view on top left part of the windows.

## Small STS Warning That Can be Fixed

There is a little bug that is annoying in STS regarding missing XML definition. You should see a tiny warning icon on your project and if you drill down, it relates to the file "log4j.xml".

The issue has been described in a Spring forum [post](#) but the "Key" that needs to be entered is:

```
-//APACHE//DTD LOG4J 1.2//EN
```

and **NOT** -//LOGGER - this changed from the previous STS version.

The solution is:

1. Menu Windows -> Preferences -> XML -> XML Catalog
  2. Click on add
  3. Click on File System and select the jar. It should be something similar to: {SpringSource install folder}\sts-2.7.1.RELEASE\plugins\org.apache.log4j\_1.2.15.v201012070815.jar
  4. Add "jar:file:" in front of the path and !/org/apache/log4j/xml/log4j.dtd at the end of the path (without the ")
  5. In the key textbox, enter -//APACHE//DTD LOG4J 1.2//EN (without the ")
  6. Click **Ok** to close this window, then **ok** again to close the preference window.
  7. Once you follow the instructions, you need to add and remove a space to trigger a modification then save the file. The warning should be gone.
- Do this for both log4j.xml files:
- one in src/main/resources
  - one in src/test/resources

## Verification

1. Select the [testlex] folder project in the package explorer window then click on play button or click on the menu "Run" -> Run  
The next window will ask you how you want to run the project. Select "Run on server".
2. In the list on the next window, you should see the "Tomcat v6.0 Server at localhost" entry that you created earlier. Select it and check the box "Always use this server..."
3. Click **Next**, then click **Finish**.  
You should then see the "Hello world!" page with the current time on it.  
You now have a functioning Spring 3 MVC project.

## Adding LexEVS as a Maven Dependency

The Spring 3 MVC project being a Maven compliant project, we will add LexEVS as dependency. It is not available in a public repository so we will need to install in our local repository first.

## Downloading the Required Files

You will need four files to successfully add LexEVS as a dependency.

1. LexEVS jar: Can be found in your local LexEVS install: {LEXEVS\_HOME}\runtime-components\lexbig.jar
  - For easier access, copy it to a short location. We will use E:\
2. LexEVS POM file: This includes the information on the dependencies LexEVS will rely on. You can download it [Lexevs-6.0.0.pom.zip](#)

- Unzip it to the same location [E:]
3. LexEVS uses a patched version of the dependency ddlutils. You need to [download it](#)
    - Place it in the same directory [E:]
  4. This dependency also has a POM file which you can download [Ddlutils-1.0-PATCHED.pom.zip](#)
    - Unzip it to the same location [E:]

## Installing a LexEVS Dependency

As these Maven dependencies are not available in a public repository, you will need to add them to our local repository.

Open a command window and navigate to the {SpringSource install folder}\maven-2.2.1.RELEASE\bin folder. On Windows 7, you can use the file explorer to navigate to this folder and then "Shift+right click" in the folder and select "Open command window here".

## JAVA\_HOME Environment Variable Missing

Should you have an error message stating the need for a JAVA\_HOME system variable when you execute the MVN commands, please follow the appropriate steps for your system to add it (steps provided below for Windows 7).

1. On Windows 7, click on the start button, then right-click on computer and select properties.
2. Select Advanced system settings in the left pane and select the advanced tab in the window that opens up. Click on Environment Variables button.
3. Click on the button New in the lower half of the window and add JAVA\_HOME as a name and the path to your JRE installation as a value. Click OK 3 times to close the windows. Close and reopen your command window and it should now work.

## MVN Commands

For all the following commands, please change \*E:\* to the path of the folder chosen to place the four files above.

1. Then enter this command at the command prompt:

```
mvn install:install-file -Dfile="E:\ddlutils-1.0-PATCHED.jar" -DpomFile="E:\ddlutils-1.0-PATCHED.pom"
```

You should then see a build successful message. The ddlUtils dependency is now available in your local repository.

2. Now enter this command at the command prompt:

```
mvn install:install-file -Dfile="E:\lexbig.jar" -DpomFile="E:\lexevs-6.0.0.pom"
```

You should also see a build successful message. The LexEVS dependency is now available in your local repository.

3. Open STS and select the menu Window -> Show View -> Other
4. Scroll down to the folder Maven and select Maven Repositories. Click OK.  
You should now have a Maven Repositories view in the lower part of the window.
5. Select and unfold Local Repositories.
6. Right-click Local Repository and select Rebuild Index from the context menu. Click Ok when asked if you're sure you want to rebuild it.

## Adjusting Scope of JUnit

LexEVS currently uses JUnit during runtime. This is a bit unusual and might change in a future release but for now, it needs to be adjusted.

1. Select the project folder [testlex] in the Package Explorer view and unfold it.
2. Double click on the pom.xml file to open it.
3. Select the Dependencies tab at the bottom of the window.
4. In the Dependencies pane, scroll down in the dependencies selectbox and click once on junit. On the right side, the Dependency Details area should present a scope which is "test" in the combobox.
5. Change it to compile.
6. Click on menu File -> Save or press Ctrl-s.

## Including LexEVS in the MVC Project

1. Select the [testlex] project. Right-click on it and select Maven -> Add Dependency from the context menu.
2. Enter lexevs in the first textbox.  
The search results should show: lexevs lexevs
3. Select it and click ok.
4. STS will now download all the dependencies needed to work with LexEVS. You can follow the progress in the progress view (tab).

## Creating a Test File

1. Select and unfold the project folder [testlex] in the package explorer.
2. Navigate to src/test/java. You should see an empty package [org.test.test]
3. Right-click on it and select New -> JUnit Test Case
4. Put the name of your test in the name textbox. We will name it LexBigConnect. Click on Finish.
5. Copy this in the file (including the last bracket):

```

package org.test.test;

import static org.junit.Assert.*;

import org.LexGrid.LexBIG.Exceptions.LBInvocationException;
import org.LexGrid.LexBIG.Impl.LexBIGServiceImpl;
import org.LexGrid.LexBIG.LexBIGService.LexBIGService;
import org.LexGrid.LexBIG.DataModel.InterfaceElements.CodingSchemeRendering;

import org.junit.Test;

public class LexBigConnect {

    @Test
    public void test() throws LBInvocationException {

        LexBIGService lbSvc = LexBIGServiceImpl.defaultInstance();
        System.out.println("The currently active terminologies are:");
        for(CodingSchemeRendering csr : lbSvc.getSupportedCodingSchemes().getCodingSchemeRendering())
            System.out.println(csr.getCodingSchemeSummary().getFormalName());

        assertTrue(true);
    }
}

```

6. Make sure to change the package name if you didn't use org.test.test initially. Save the file.

## Adding a JVM Parameter to Locate the lbconfig.props

1. In STS, select the menu Run -> Run Configurations...
2. Select JUnit in the left list.
3. Select the new button (white rectangle with a little yellow +)  
The name of the profile will be defaulted to the name of your class [LexBigConnect]
4. Click on the tab Arguments and add this parameter to the VM textbox:

```
-DLG_CONFIG_FILE="\{LEXEVS_HOME\}\resources\config\lbconfig.props"
```

5. Replace {LEXEVS\_HOME} by the absolute path to the LexEVS local install done earlier in the process.
6. Then click Apply followed by Run.  
You should then see a successful completion of a JUnit test. The console log will also show "The currently active terminologies are:" followed by the name of the terminologies [Automobiles] if you loaded it.

## Integrating LexEVS in a Web Page

### Adding a JVM Parameter to Locate the lbconfig.props

In STS, double-click on the "Tomcat v6.0..." entry in the server view (left lower corner).

1. This will open the server Overview. Click on the "Open launch configuration" link in the General Information section.
2. Select the Arguments tab and in the VM arguments textbox, at the bottom, after the existing arguments add this:

```
-DLG_CONFIG_FILE="\{LEXEVS_HOME\}\resources\config\lbconfig.props"
```

3. Replace {LEXEVS\_HOME} by the absolute path to the LexEVS local install done earlier in the process. Make sure there's a space between the end of the last existing argument and the dash of this one.
4. Click Apply then click OK.

### Adjusting the Controller

Although this is unlikely the way in which you will integrate LexEVS in a MVC project, it will confirm that using the project with Tomcat works. This assumes that you have at least one loaded and active terminology in your local LexEVS Server.

1. Navigate the project tree to src/main/java and double-click on HomeController.java to open it.
2. Copy this in the file (including the last bracket):

```

package org.test.test;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

import org.LexGrid.LexBIG.DataModel.InterfaceElements.CodingSchemeRendering;
import org.LexGrid.LexBIG.Exceptions.LBInvocationException;
import org.LexGrid.LexBIG.Impl.LexBIGServiceImpl;
import org.LexGrid.LexBIG.LexBIGService.LexBIGService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 * Handles requests for the application home page.
 */
@Controller
public class HomeController {

    private static final Logger logger = LoggerFactory.getLogger(HomeController.class);

    /**
     * Simply selects the home view to render by returning its name.
     * @throws LBInvocationException
     */
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) throws LBInvocationException {
        logger.info("Welcome home! the client locale is " + locale.toString());

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG, DateFormat.LONG, locale);

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        Integer n = 1;

        LexBIGService lbSvc = LexBIGServiceImpl.defaultInstance();
        for(CodingSchemeRendering csr : lbSvc.getSupportedCodingSchemes().getCodingSchemeRendering()){
            System.out.println(csr.getCodingSchemeSummary().getFormalName());
            model.addAttribute("term"+n.toString(), csr.getCodingSchemeSummary().getFormalName());
            n++;
        }

        return "home";
    }
}

```

3. Make sure to change the package name if you didn't use org.test.test initially. Save the file.

## Adjusting the JSP Page

1. In the Package Explorer, navigate to src/main/webapp/WEB-INF/views and open home.jsp by double-clicking.
2. Replace the text in the file with this one:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>
    Hello world!
</h1>

<p>The time on the server is ${serverTime}.</p>
<p>One of the active terminologies on the LexEVS server is: ${term1}</p>
</body>
</html>
```

3. Save the file and close it.

## Running the Project

1. Select the project [textlex] in the Package Explorer.
2. Select the menu Run -> Run. Select "Run on Server" then click OK.
3. You should see the home page displayed with the date/time and, on the last line, the name of one of your active terminology in LexEVS.
4. You now have a functional Spring 3 MVC project interacting successfully with a local instance of LexEVS.

## Acknowledgement

This guide was contributed by the [TRANSFoRM group](#)  / [INSERM U936](#) 