ExpressUI Development Environment Setup Guide

This guide walks you through setting up the **optimal** development environment where everything "just works."

Of course, ExpressUI will work fine in many other environments, with other tools and other versions. However, the philosophy behind ExpressUI is to speed up development as much as possible and lower cost. So, that is why we provide this guide, to help you get going quickly and achieve maximum velocity.

Install JDK

1. Download latest stable Java SE 7 from here: http://www. oracle.com/technetwork/java/javase/downloads/index.html
2. Install to **C:\jdk1.7**
   Note: ExpressUI has also been extensively tested against Java SE 6.

Install Maven

1. Download latest stable Maven 3 from here: http://maven.apache.org/download.html
2. Unzip to **C:\maven**

Configure Maven for Corporate HTTP Proxy (Depends)

Some corporate environments require that all HTTP traffic go through a corporate proxy. If this is the case for you, then edit or create the file: **~/.m2/settings.xml**.

Here's a sample **settings.xml** of how to configure Maven to use an HTTP proxy:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                      http://maven.apache.org/xsd/settings-1.0.0.xsd">
    <proxies>
        <proxy>
            <active>true</active>
            <protocol>http</protocol>
            <host>proxy.somecompany.com</host>
            <port>80</port>
        </proxy>
    </proxies>
</settings>
```

Set up Environment Variables

1. MAVEN_HOME= C:\maven
2. M2_HOME= C:\maven
3. JAVA_HOME=C:\jdk1.7
4. Add to PATH: %MAVEN_HOME%\bin;%JAVA_HOME%\bin

Install Jetty

1. Download latest stable Jetty 7 from here: http://download.eclipse.org/jetty/
2. Unzip to C:\jetty
3. Optionally, change the port number in **C:\jetty\etc\jetty.xml** like so:

   ```
   <Property name="jetty.port" default="7001"/>
   ```

   Note: using port 7001 allows swapping Jetty for Weblogic during development without changing the URL. In corporate environments, it is good to test the application periodically against whatever application server will be used in production.

Install Intellij

1. Download the latest Ultimate version from here: http://www.jetbrains.com/idea/download/index.html
2. Optionally, go to **File:Import Settings** and import **intellij_settings.jar** downloaded from here: https://github.com/juanosuna/expressui-sample/blob/master/extra/intellij_settings.jar
   Note: this is an optional step for novice Intellij users who want to make sure everything in Intellij "just works."

Alternatively Install Eclipse

1. Download the latest Eclipse version from here: http://www.eclipse.org/downloads/
2. Go to **File:Import:Existing Maven Projects** and import the sample application.

Install ExpressUI Sample Application

1. Go to project page: https://github.com/juanosuna/expressui-sample
2. Click **Downloads** button to download project as a zip. Unzip to **C:\Projects\expressui-sample**.
3. Some corporate environments require that all HTTP traffic go through a corporate proxy. If this is the case for you, then edit the file: **src/main/resources/application.properties** and set the two properties: **http.proxyHost** and **http.proxyPort**.
4. Open DOS window in the root of the project and type following commands:
   - **mvn clean install**
   - **mvn jetty:run**

Note: running these commands for the first time could take 10-15 minutes to download all the dependencies. Not all downloaded dependencies are actually used in the project. Maven just aggressively caches libraries in ~/.m2/repository.

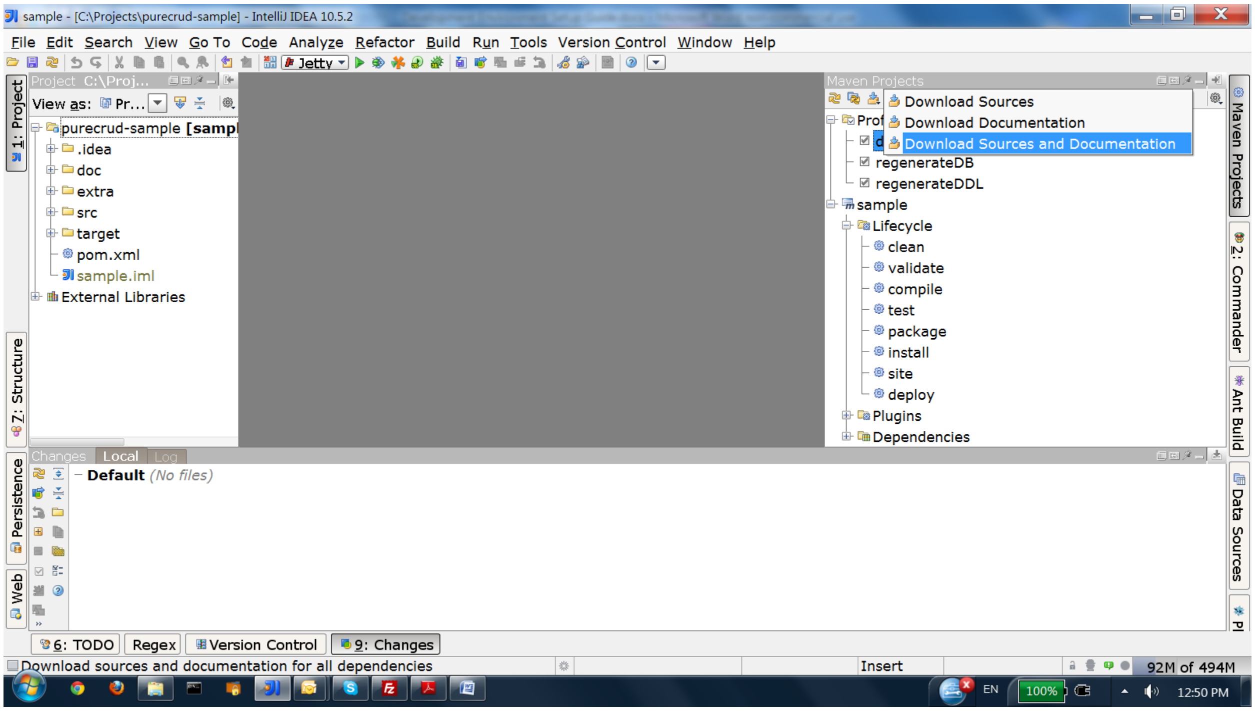
5. Try the application URL: http://localhost:7001/sample/app

Import Sample Application into IDE (Intellij or Eclipse)

The import process has been tested with both Intellij and Eclipse, and both IDEs are fully compatible with the ExpressUI technology stack: Spring, Hibernate, Vaadin, Maven and Jetty. However, generally speaking, Intellij is higher quality and provides a smoother experience. So, if you have the choice and the money, Intellij is the way to go.

For Intellij, see this video tutorial for instructions on how to import the sample application and configure the data source and the Jetty plugin: http://www.brownbagconsulting.com/wp-content/uploads/2011/09/intellij_expressui_tutorial.wmv.

One cool little feature of Intellij is that you can click an icon in the Maven panel to download sources and Javadoc for the ExpressUI framework and all its library dependencies. This not only provides instant Javadoc within the IDE but allows you to navigate and even debug into source code for ExpressUI, Spring, Hibernate and Vaadin. What a great way to learn!



Download sources and documentation for all dependencies

Alternatively, for Eclipse, go to **File:Import:Existing Maven Projects** and import the sample application into Eclipse.

Install and Configure JRebel (Optional)

The ExpressUI framework and sample application were developed with JRebel, Intellij and Intellij's built-in Jetty plugin. This combination makes for a lightning-fast development environment where no server restarts or redeployments are required.

Even without JRebel, restarting should only take about 15-30 seconds, depending on your hardware. But why not save those precious seconds and maximize productivity?

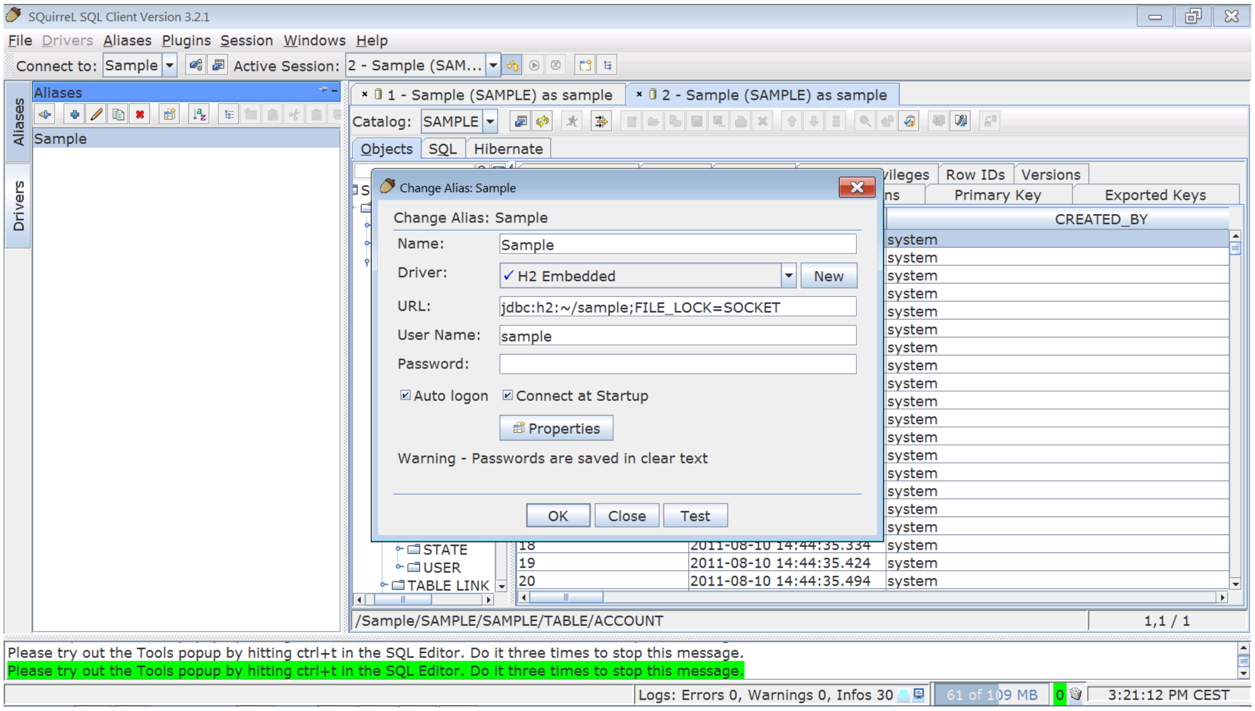
You can download a trial of JRebel here: http://www.zeroturnaround.com/jrebel/current/

The sample application already contains a rebel.xml file. After installing JRebel, just edit the file and change the absolute file path to one that is correct for your environment.

Install and Configure SQuirreL SQL Client

SQuirreL SQL is a free tool that is capable of reading H2 database files, which ExpressUI uses as the lightweight database for development. This is useful for debugging.

1. Download and following instructions here: http://www.squirrelsql.org/#installation
2. Grab the JDBC URL from **database.dev.properties** in the project and paste into the SQuirreL dialog box for configuring the database connection:


SQuirreL SQL Client Version 3.2.1
File Drivers Aliases Plugins Session Windows Help
Connect to: Sample
Active Session: 2 - Sample (SAM…
Aliases
Drivers
Sample
1 - Sample (SAMPLE) as sample
2 - Sample (SAMPLE) as sample
Catalog: SAMPLE
Objects SQL Hibernate
ileges Row IDs Versions
ns Primary Key Exported Keys
CREATED_BY
STATE
USER
TABLE LINK

Change Alias: Sample

Change Alias: Sample

Name: Sample

Driver: ✓ H2 Embedded [New]

URL: jdbc:h2:~/sample;FILE_LOCK=SOCKET

User Name: sample

Password:

☑ Auto logon ☑ Connect at Startup

[Properties]

Warning - Passwords are saved in clear text

[OK] [Close] [Test]

18	2011-08-10 14:44:35.334	system
19	2011-08-10 14:44:35.424	system
20	2011-08-10 14:44:35.494	system

/Sample/SAMPLE/SAMPLE/TABLE/ACCOUNT 1,1 / 1

```
Please try out the Tools popup by hitting ctrl+t in the SQL Editor. Do it three times to stop this message.
Please try out the Tools popup by hitting ctrl+t in the SQL Editor. Do it three times to stop this message.
```

Logs: Errors 0, Warnings 0, Infos 30 61 of 109 MB 0 3:21:12 PM CEST