

1. Download and install JDK 8 latest update

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Oracle Technology Network > Java > Java SE > Downloads

Products Solutions Downloads Store Support Training Partners About OTN

Overview Downloads Documentation Community Technologies Training

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u81 Checksum
JDK 8u92 Checksum

Java SE Development Kit 8u91

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.72 MB	jdk-8u91-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.69 MB	jdk-8u91-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u91-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u91-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u91-linux-x64.rpm
Linux x64	172.97 MB	jdk-8u91-linux-x64.tar.gz
Mac OS X	227.29 MB	jdk-8u91-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.59 MB	jdk-8u91-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.95 MB	jdk-8u91-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.29 MB	jdk-8u91-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u91-solaris-x64.tar.gz
Windows x86	182.11 MB	jdk-8u91-windows-i586.exe
Windows x64	187.41 MB	jdk-8u91-windows-x64.exe

Java SE Development Kit 8u92

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	160.26 MB	jdk-8u92-linux-i586.rpm
Linux x86	174.94 MB	jdk-8u92-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u92-linux-x64.rpm
Linux x64	172.99 MB	jdk-8u92-linux-x64.tar.gz
Mac OS X	227.32 MB	jdk-8u92-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.47 MB	jdk-8u92-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.93 MB	jdk-8u92-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.35 MB	jdk-8u92-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u92-solaris-x64.tar.gz
Windows x86	188.43 MB	jdk-8u92-windows-i586.exe
Windows x64	193.66 MB	jdk-8u92-windows-x64.exe

2. Make sure Java and Javac work from the CLI

Type 'java -version' and 'javac -version' and that their version is _92 (in the screenshot it's _66, but you get the point)

```
koen@koen-HP-EliteBook-8570w: ~  
koen@koen-HP-EliteBook-8570w:~$ java -version  
java version "1.8.0_66"  
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)  
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)  
koen@koen-HP-EliteBook-8570w:~$
```

```
koen@koen-HP-EliteBook-8570w: ~  
koen@koen-HP-EliteBook-8570w:~$ javac -version  
javac 1.8.0_66  
koen@koen-HP-EliteBook-8570w:~$
```

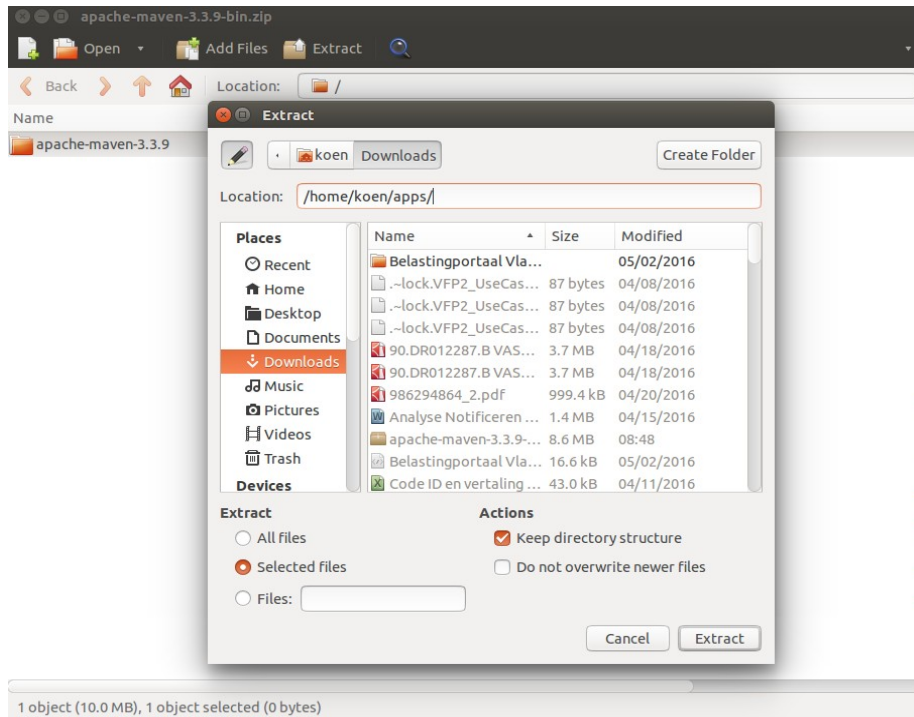
Note: If this doesn't work, create a (System wide) System variable "JAVA_HOME" and let it point to the install directory. Eg. `JAVA_HOME="c:\Program Files (x86)\Java\jdk1.8.0_92"`. Then add this variable to the end of the PATH variable (if the PATH variable does not exist, create it) `PATH=blabla;blabla;%JAVA_HOME%`. **Reboot**. The "JAVA_HOME" variable is just some best

practice (you could have added the Java path directly to the PATH variable as well) so you can easily change the path later on. Otherwise you have to start looking into the exploded paths in the PATH variable which is not so easy if you have many entries. Also, if you would need the java home from different variables (eg. Application specific system variables) you will only need to change it in one place afterwards.

3. Download and install Maven 3

<http://apache.eu.be/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.zip>

Extract the zip contents to somewhere on your disk (and remember where it is)



Create a (System wide) System variable “MAVEN_HOME” and let it point to the install directory. Eg. `MAVEN_HOME= "c:\users\koen\development\apps\apache-maven-3.3.9"` . Then add this variable to the end of the PATH variable (if the PATH variable does not exist, create it) `PATH=blabla;blabla;%JAVA_HOME%;%MAVEN_HOME%` . **Reboot**. Again, the “MAVEN_HOME” variable is just some best practice. Test by opening a command prompt and type '`mvn -version`'

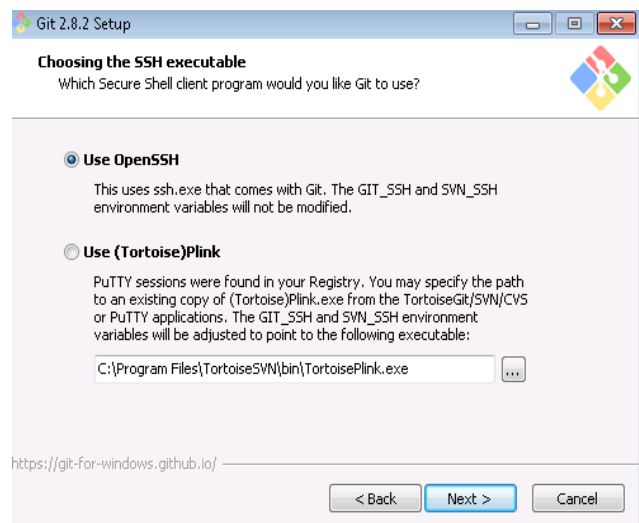
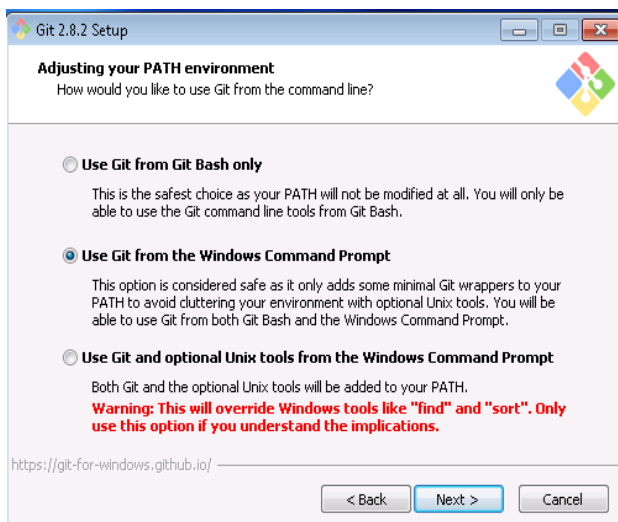
```
koen@koen-HP-EliteBook-8570w: ~  
koen@koen-HP-EliteBook-8570w:~$ mvn -version  
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T17:41:47+01:00)  
Maven home: /home/koen/devel/apps/apache-maven  
Java version: 1.8.0_66, vendor: Oracle Corporation  
Java home: /home/koen/devel/apps/jdk1.8.0_66/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "3.13.0-85-generic", arch: "amd64", family: "unix"  
koen@koen-HP-EliteBook-8570w:~$
```

4. Download and install Git

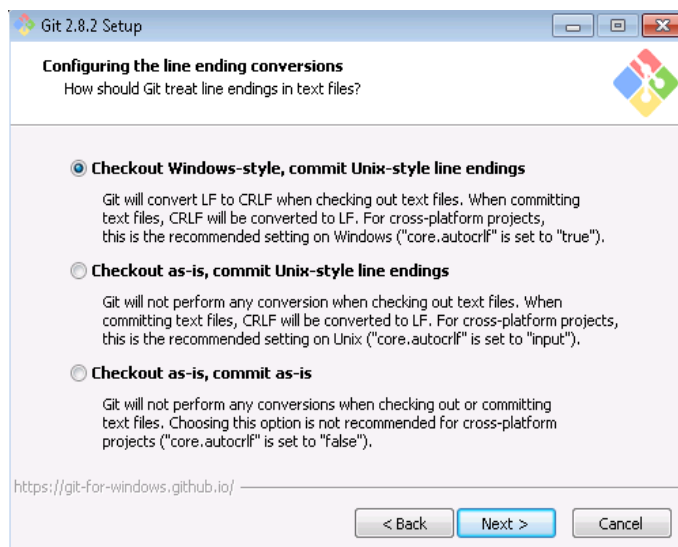
If for whichever reason you already installed Git, please make sure you have the latest version. Although the ports became stable, it is still a good idea to be up to date to get the best possible experience.

<https://git-scm.com/download/win>

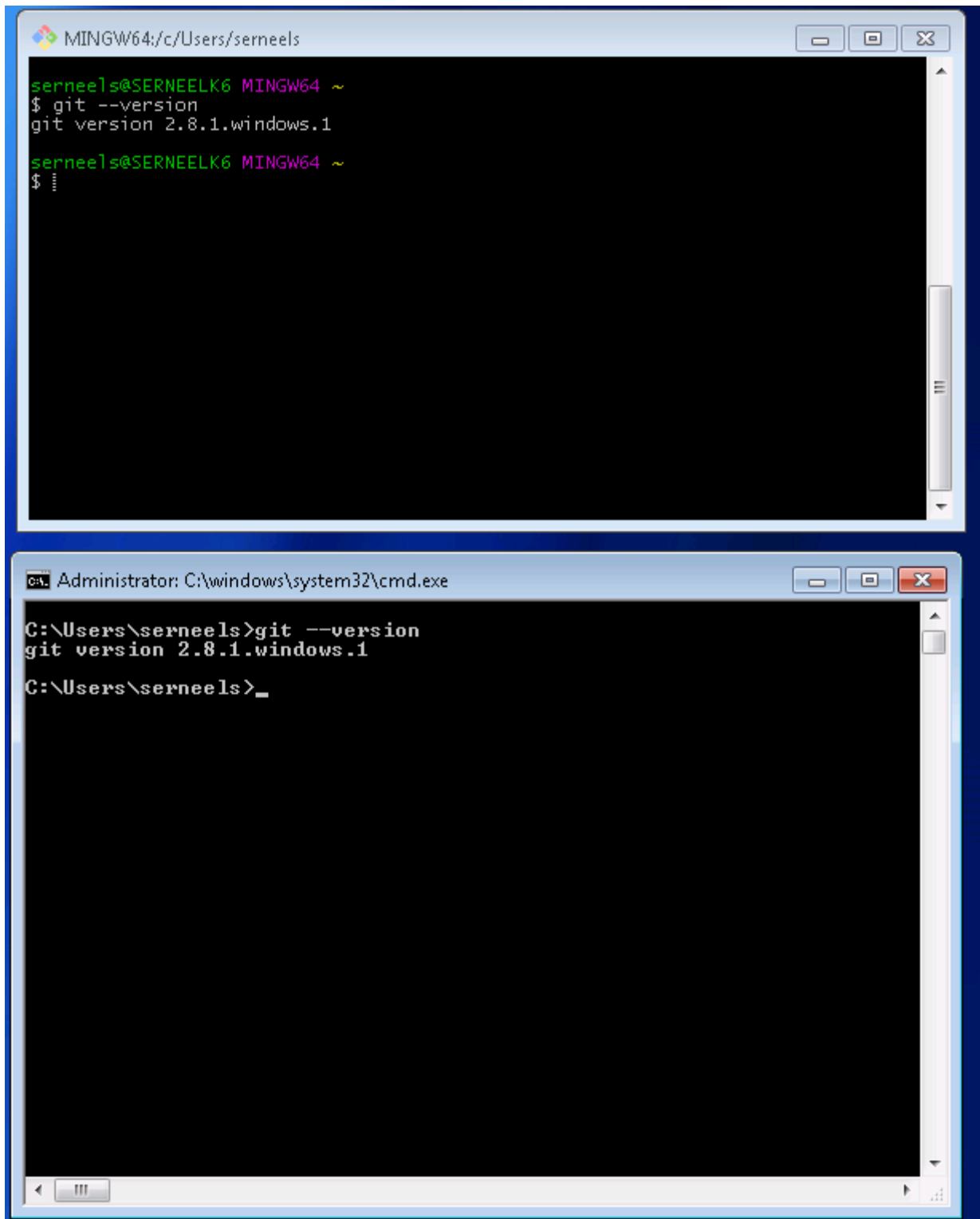
All the defaults are actually OK. Just make sure the “Use Git from the windows Command Prompt” is selected and that you use OpenSSH (both should be the default)



For the CRLF issues, leave it set to the first option :



Test git by opening the windows CLI or git bash and type '**git --version**'



The image displays two terminal windows side-by-side, both showing the output of the 'git --version' command. The top window is a MINGW64 terminal with the title 'MINGW64:/c/Users/serneels'. It shows the prompt 'serneels@SERNEELK6 MINGW64 ~', the command '\$ git --version', and the output 'git version 2.8.1.windows.1'. The bottom window is a Windows Command Prompt with the title 'Administrator: C:\windows\system32\cmd.exe'. It shows the prompt 'C:\Users\serneels>', the command 'git --version', and the output 'git version 2.8.1.windows.1'.

```
MINGW64:/c/Users/serneels
serneels@SERNEELK6 MINGW64 ~
$ git --version
git version 2.8.1.windows.1
serneels@SERNEELK6 MINGW64 ~
$

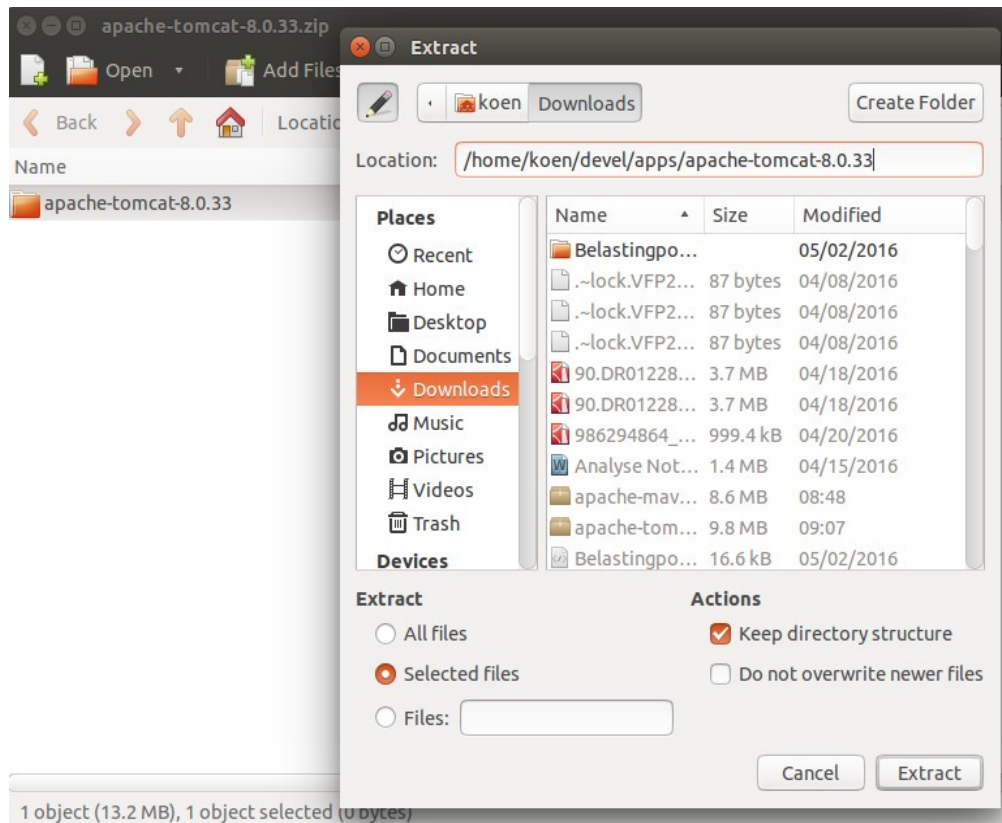
Administrator: C:\windows\system32\cmd.exe
C:\Users\serneels>git --version
git version 2.8.1.windows.1
C:\Users\serneels>
```

Note: I also included a PDF from our Wiki (Git-vfp-wiki.pdf) in which an article gives an intro into git, in case your interested

5. Download and unzip tomcat

Again, if you already have tomcat, make sure it's the latest one, which is 8.0.33 . Version 9 is still in milestone btw and is a Java EE8 implementation of the servlet spec. This version is also not finalized. Version 8 of tomcat is Java EE7 servlet spec (Servlet 3.1 JSR 340)

<http://apache.cu.be/tomcat/tomcat-8/v8.0.33/bin/apache-tomcat-8.0.33.zip>



Thats it.

6. Download and install IntelliJ

<https://www.jetbrains.com/idea/download/download-thanks.html?platform=windows>

Nothing special to mention here.

Some general information: there are different versions of IntelliJ: ultimate, community and a different branding called webstorm (all other ones are non-java, JetBrains for example also has .net products). The ultimate edition contains everything. Community is a free edition which has lack of framework and server support. Webstorm has specific JS support which is not there in community version but is included in the ultimate. So basically we use the ultimate but it requires a license .

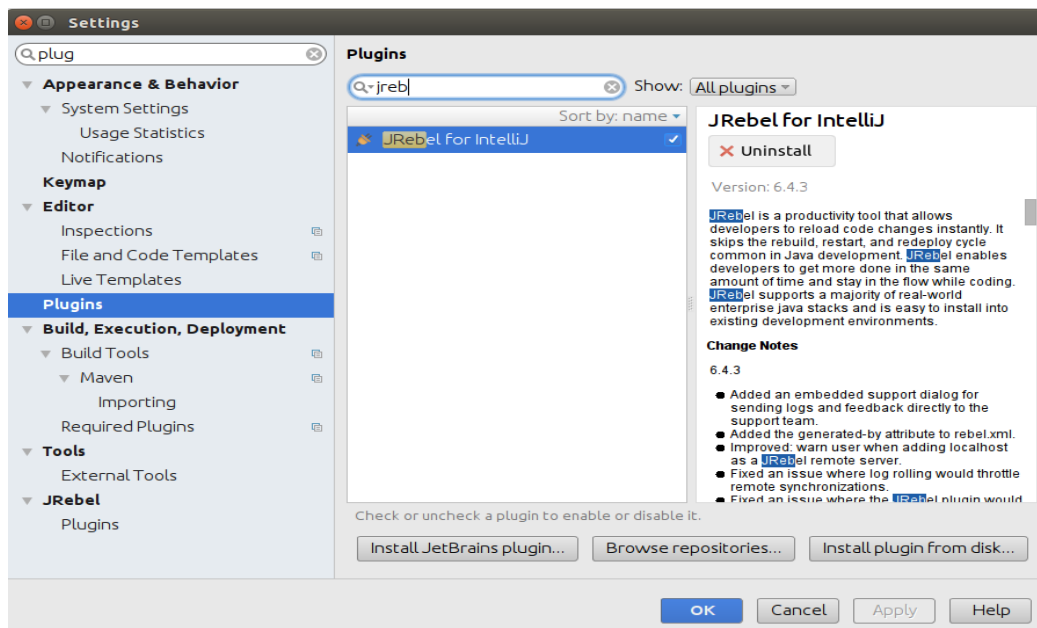
Peter DB will foresee licenses for everyone. For now we will be using a trial version which is valid for 30 days. All functionality is available. Registering afterwards is as simply as entering your username and password for your JetBrains account to which the license will be added (you'll be invited to create an account once your licence will be assigned). In other words there is no need to reinstall, you'll just continue using your setup as-is

7. JRebel

We use JRebel as an extra tool for reloading classes inside a JRE. A JRE is the Java runtime, if you start anything java related (like for example, Tomcat also runs inside the JRE) it will use the JRE. However, once the JRE is started, changes to class files are not always hot-reloaded. The JRE has some support for this, but it is rather limited. The solution is of course to restart the JRE (in our case restarting Tomcat). This is however time consuming. JRebel is a tool which will help us doing exactly that (complete hot-reloading without restarting anything)

JRebel is a plugin, you can download it in IntelliJ by going to **File → Settings**

Type “plug” in the top right search field, this will limit the options that have 'plug' in their name. Click “plugins”. Then, in the middle page of the plugin window type 'jreb' it will then show all plugins with 'jreb' in their name (which is only one). Install plugin, restart IntelliJ



After restart, enter the URL for the license server (see next page). At this time you'll be using our license server (we only have 8 licenses). Peter DB will foresee extra licenses for the DVS team which will be added to the License server

License server: <https://199.228.84.204/jrebel/06cfada4-5fe5-4509-afd2-694f189b4c58>

