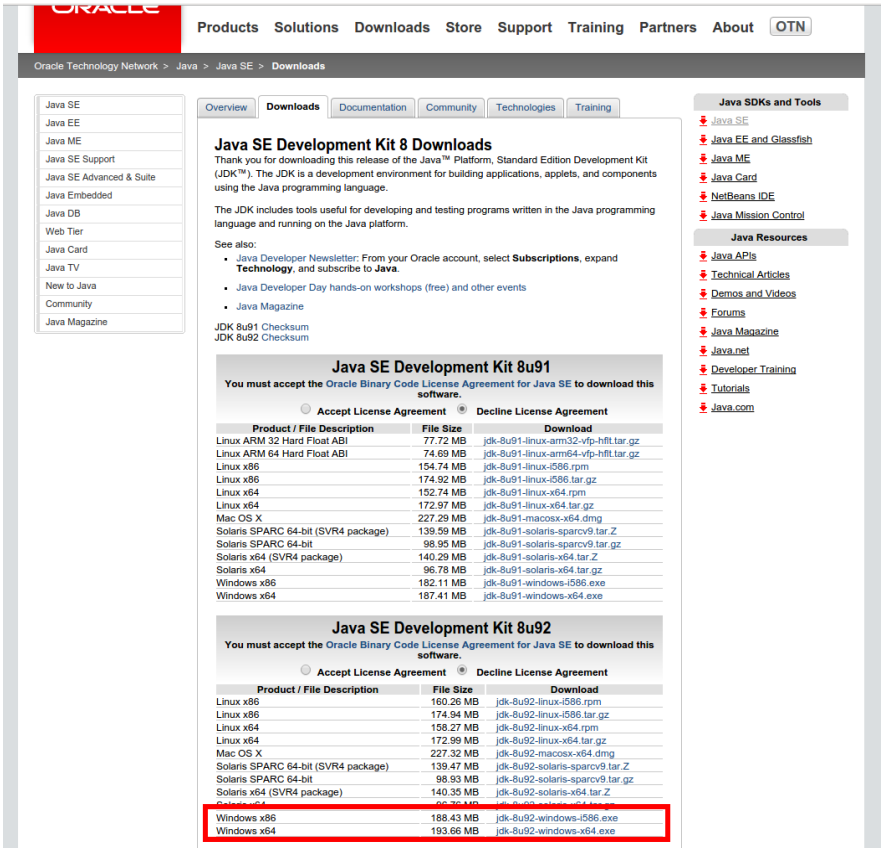
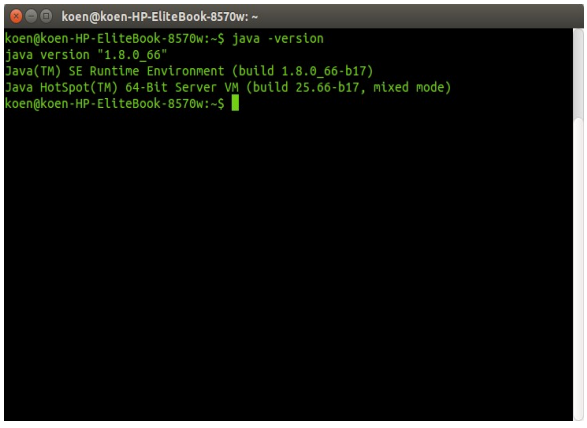
1. Download and install JDK 8 latest update

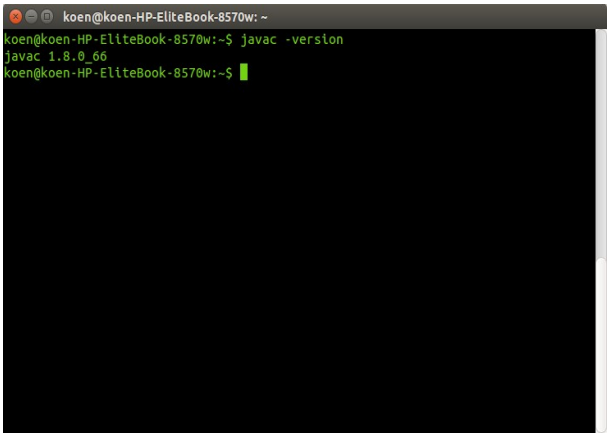
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html



2. Make sure Java and Javac work from the CLI

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





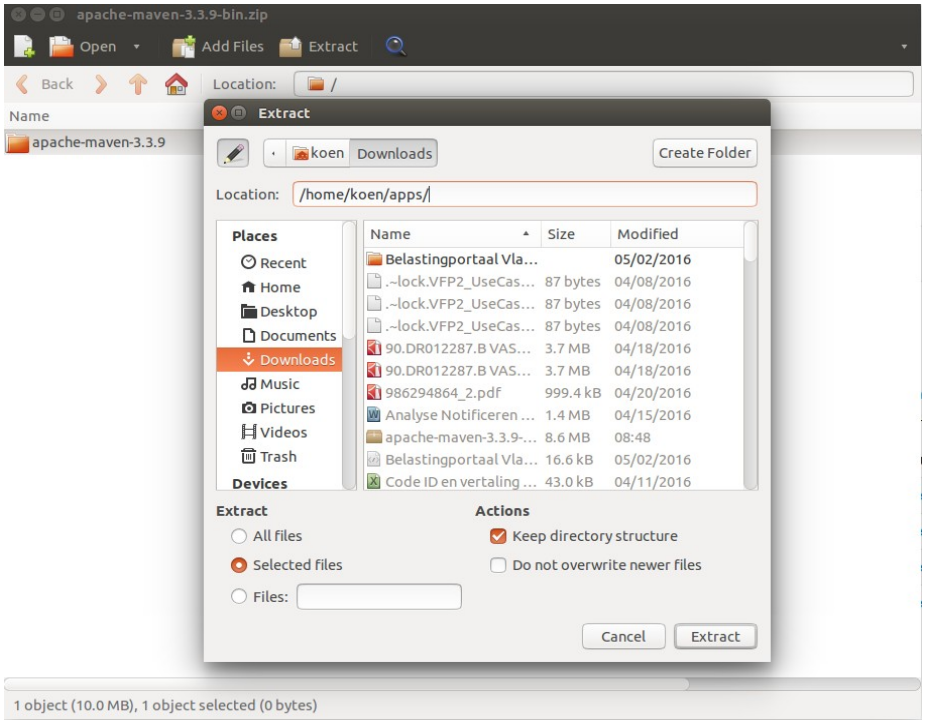
Note: If this doesn't work, create a (System wide) System variable “JAVA_HOME” and let it point to the install directory. Eg. *JAVA_HOME=”c:\Program Files (x86)\Java\jdk1.8.0_92”* . Then add this variable to the end of the PATH variable (if the PATH variable does not exist, create it) *PATH=blabla;blabla;%JAVA_HOME%*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

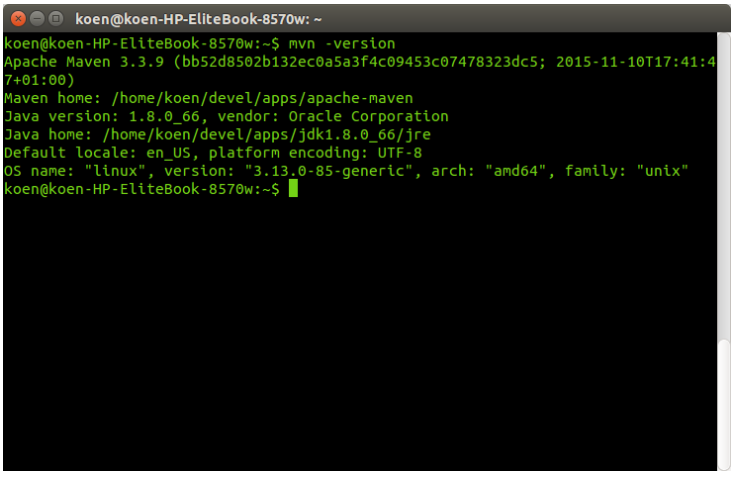
3. Download and install Maven 3

http://apache.cu.be/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.zip

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



Create a (System wide) System variable “MAVE_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**'

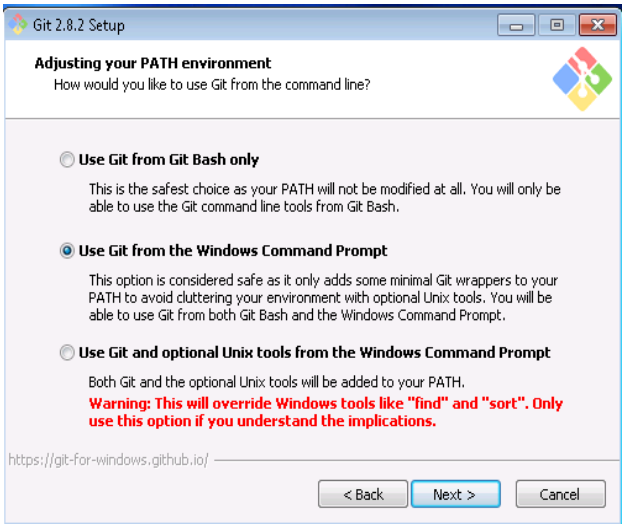


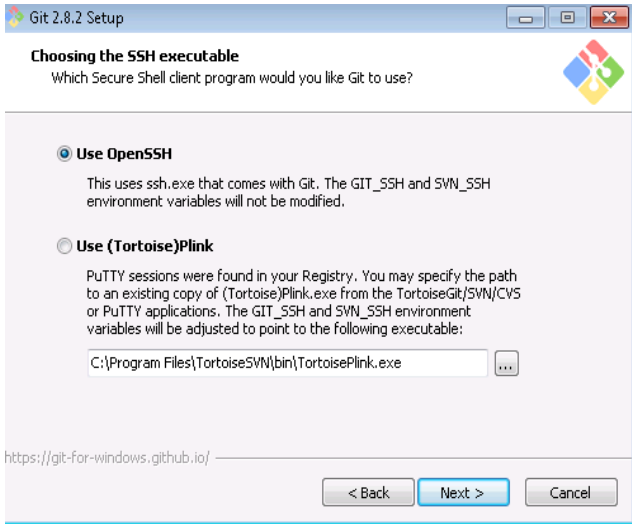
4. Download and install Git

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

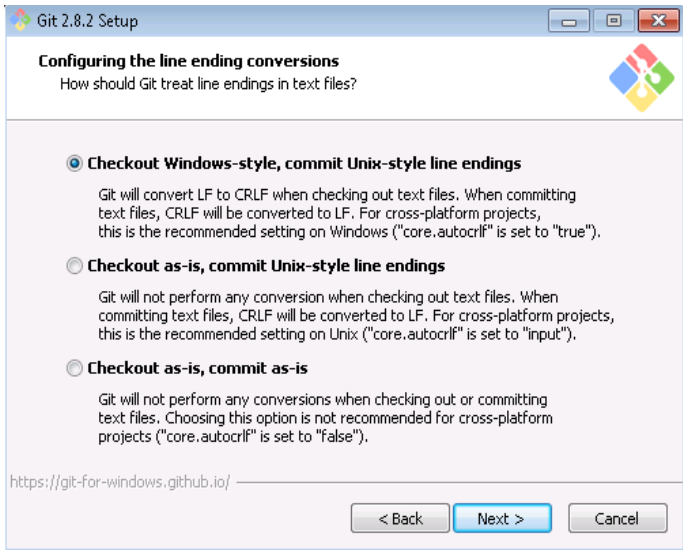
https://git-scm.com/download/win

All the deaults 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)

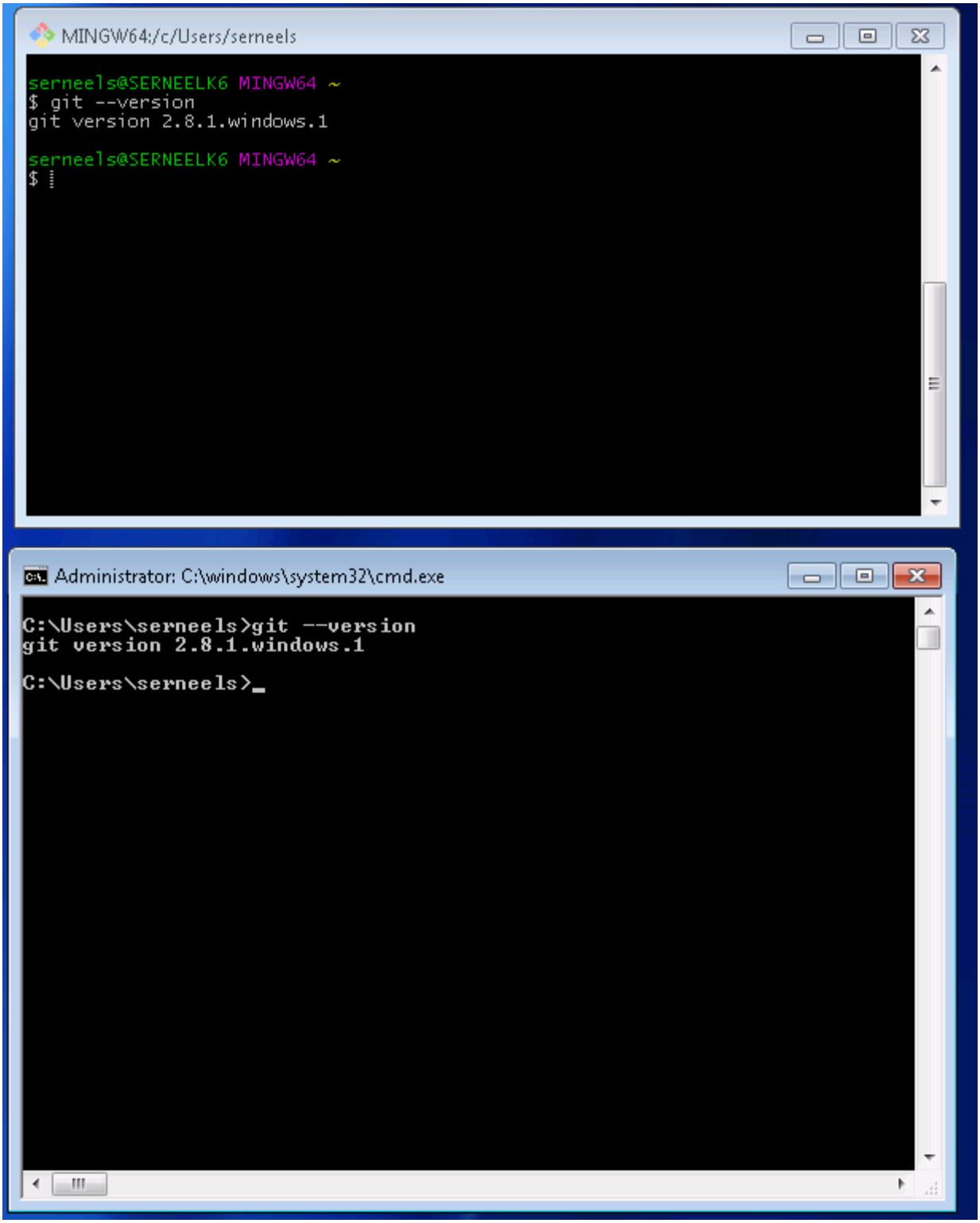




For the CRLF issues, leave it set to the first option :



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

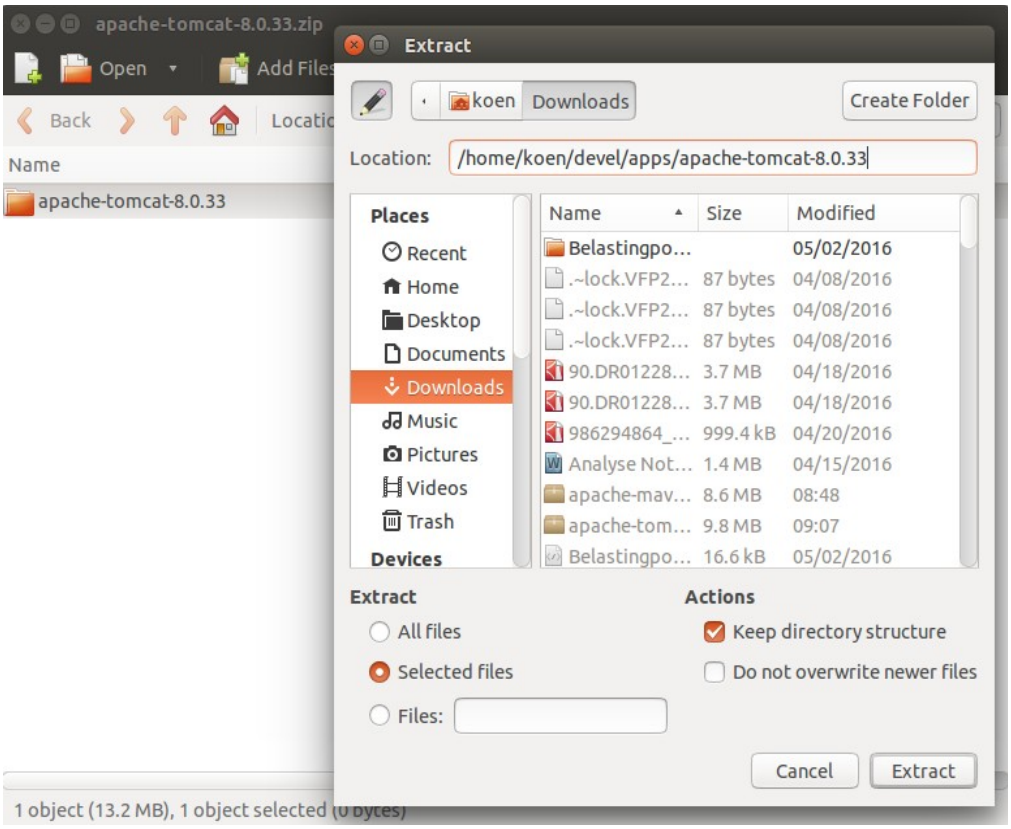


Note: I also included a PDF from our Wiki (Git-vfp-wiki.pdf) in which an article gives an intro into git, in case your interested

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.

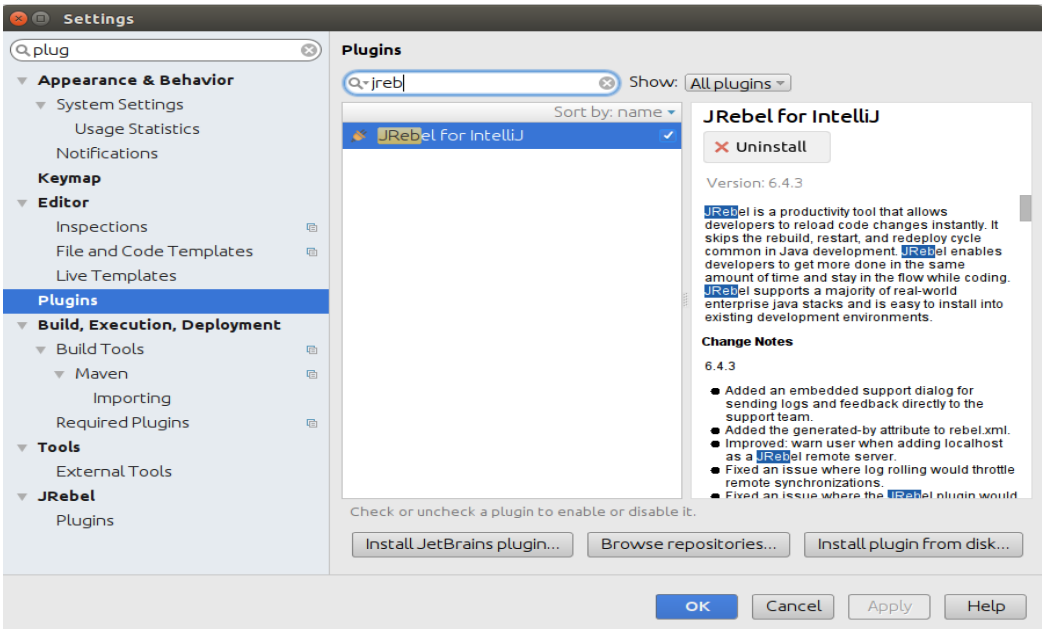
Some general information: there are different versions of Intellij: ultimate, community and a different branding called webstorm (all other ones are non-java, Jetbrains for example also has .net products). The ultimate edition contains everything. Community is a free edition which has lack of framework 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 anyting 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 n ame (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