

Tridium_N4_开发环境设置_基础开发环境篇

作者：杨超 (nightycd@163.com QQ 群里的 ID: ycd)

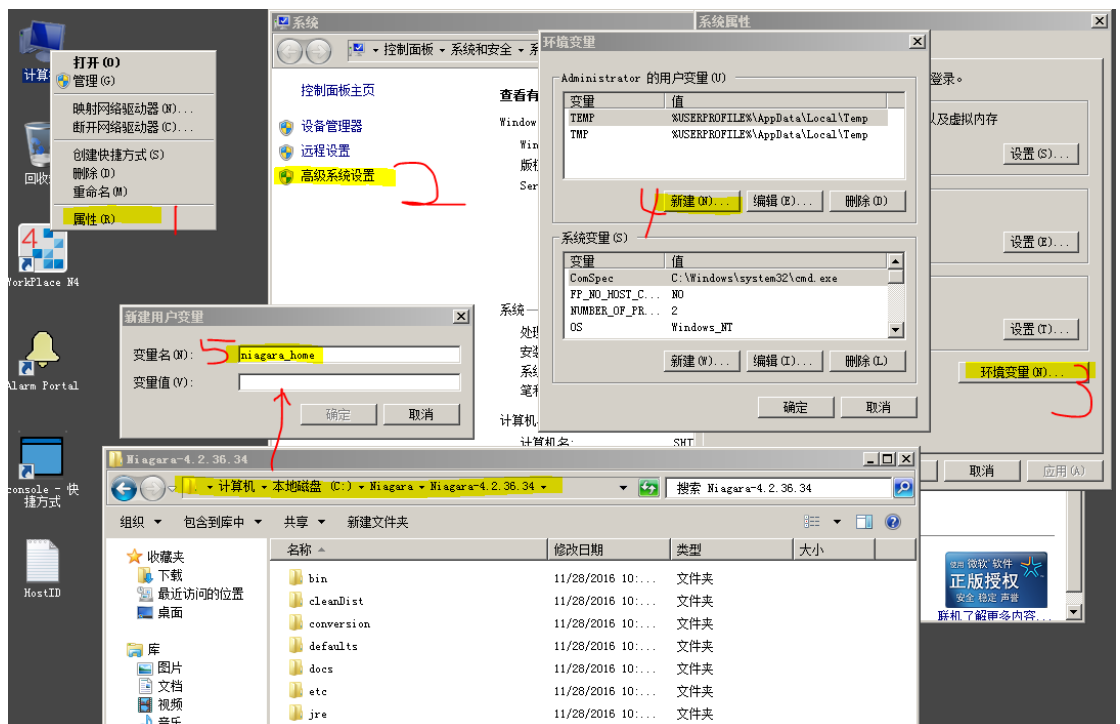
前几日看了黄工的环境设置视频录像，可能对新手来说还是有点难，这里我做了一个逐步图解的详细的的环境设置文档献给大家，同时也算是自己做个笔记免得日后忘了。

本文基于黄工的一些提示，以及官方开发文档，其余为我个人的自学经验，如有错误欢迎指正。

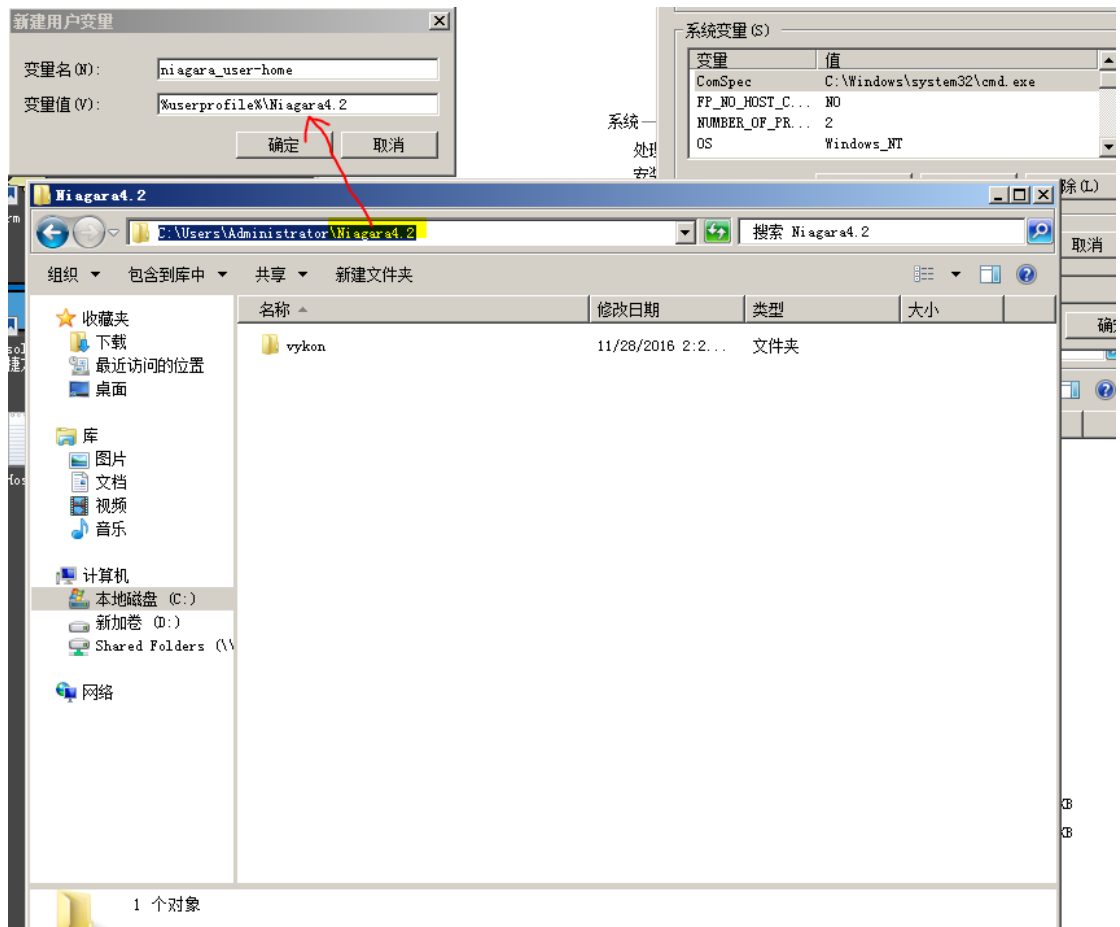
首先还是感谢黄工的一些提示，然后是要感谢官方文档，最后感谢 google 翻译^_^。

其次，进入 N4 后，它的开发工具经常会自动下载工具部件，而这些部件时常会被墙，所以建议你最好能有个稳定的翻墙 VPN，没有的同学可以参考一下 <http://jsq.re/>，里面有免费的也有收费的。

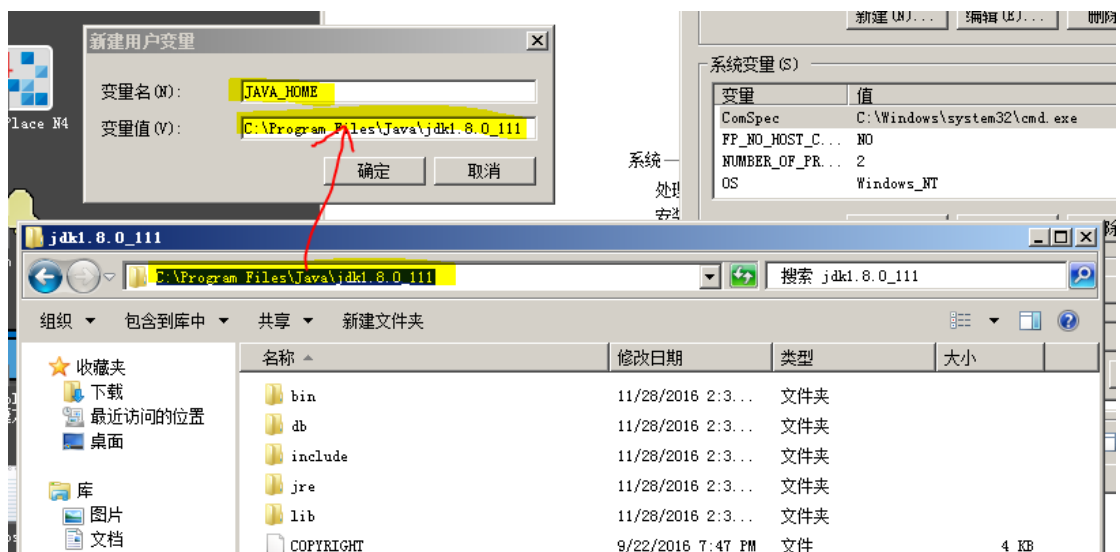
1. 首先，你要有一份 Tridium N4，并安装好。
2. 安装 JRE、JDK，参见 java.com
3. N4 的开发环境需要设置环境变量。设置环境变量，需要右击“我的电脑”->“属性”->“高级系统设置”->“环境变量”->“新建”。
4. 建如下变量：
 - a) 变量名: niagara_home 变量值: C:\Niagara\Niagara-4.2.36.34 (黄字部分是你的具体的 Niagara 软件的安装目录路径，此处为实例)



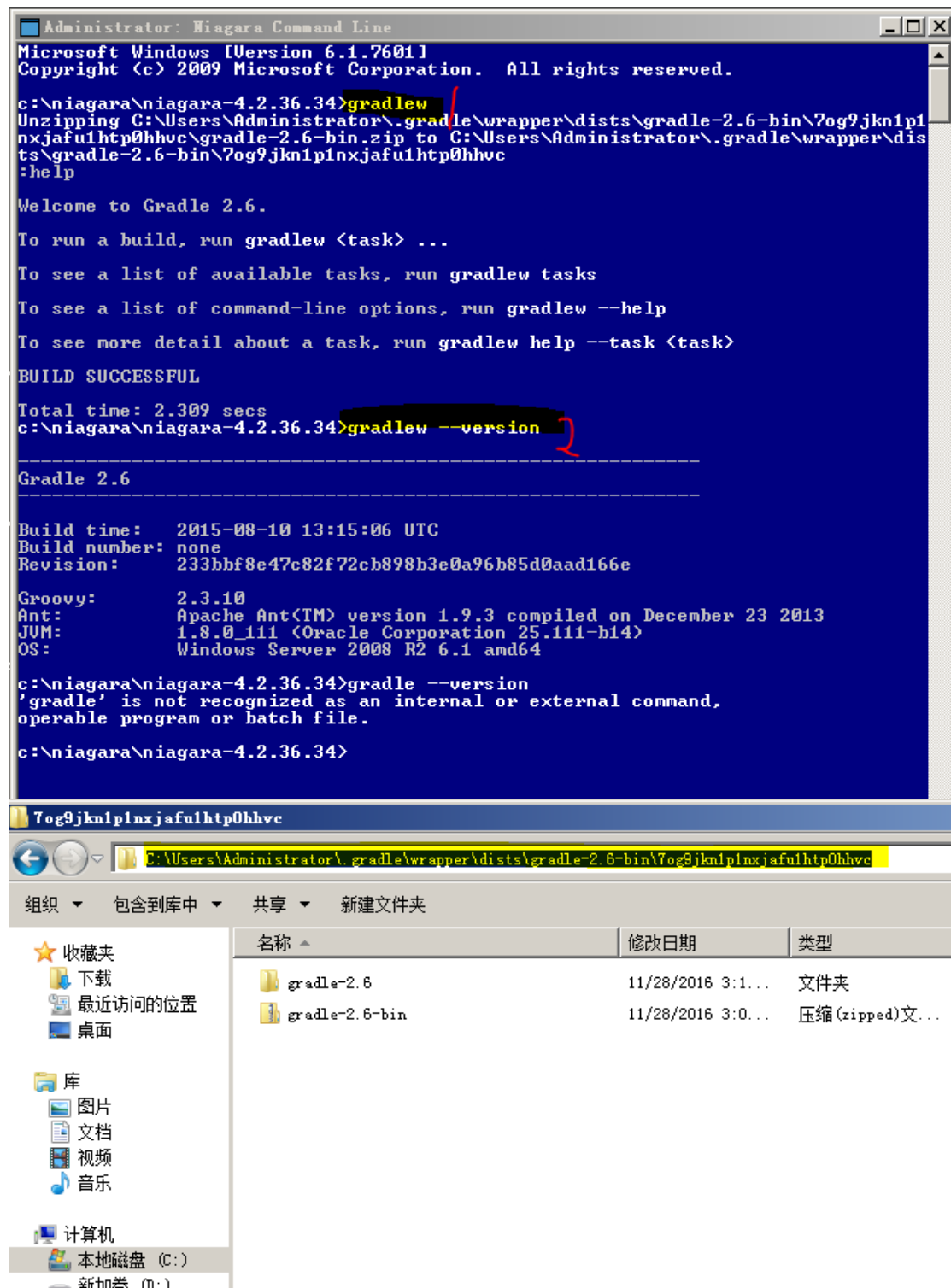
- b) 变量名: niagara_user_home 变量值: %userprofile%\Niagara4.2 (黄字部分随安装版本不同而不同, 具体要看 C:\Users\用户名下的具体 N4 文件夹名, 此处为实例)



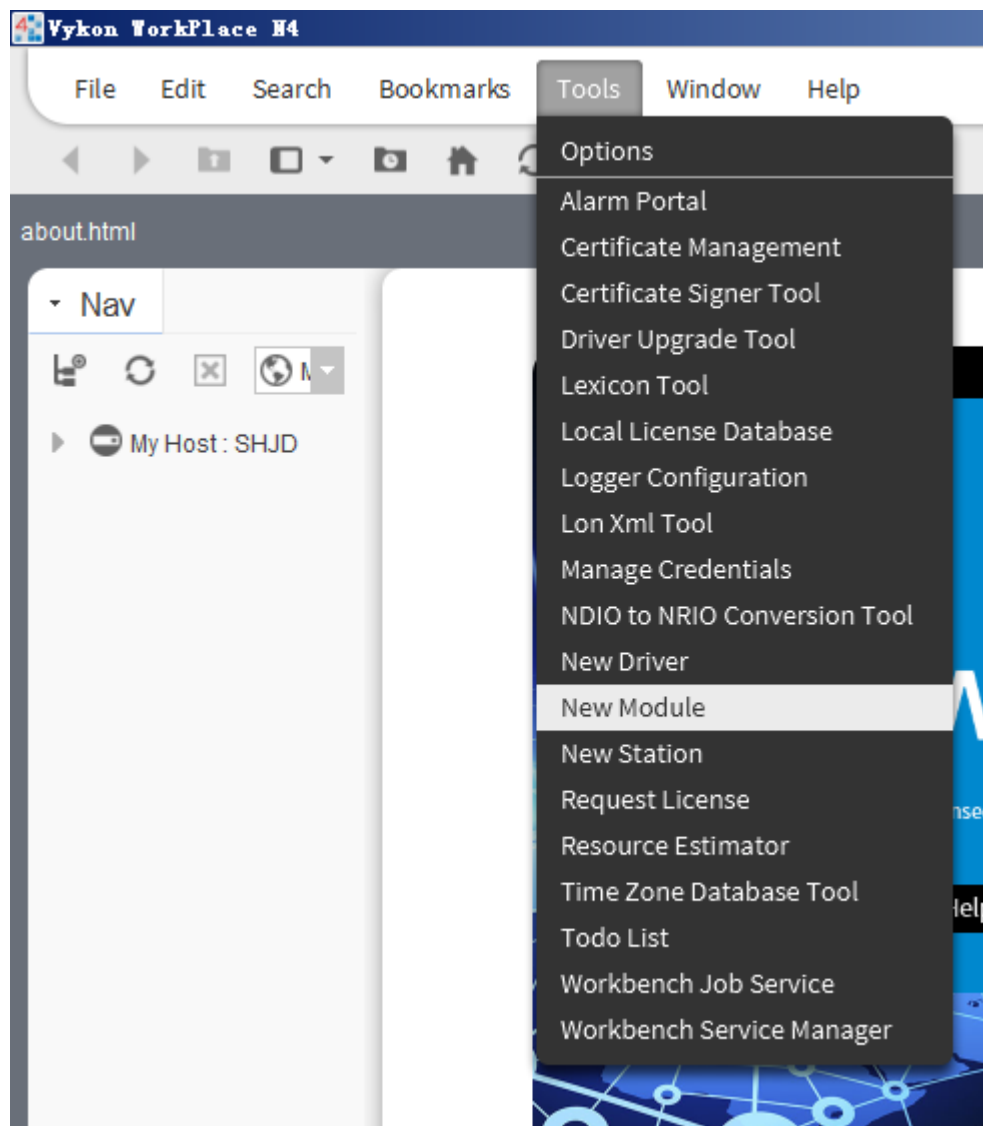
- c) 变量名: niagara_dev_home 变量值: %niagara_user_home%
- d) 变量名: JAVA_HOME 变量值: C:\Program Files\Java\jdk1.8.0_111 (黄字处为你的具体的 JDK 的安装目录路径, 此处为实例) (此变量可选, 具体是后面的 IntelliJ IDEA 用)



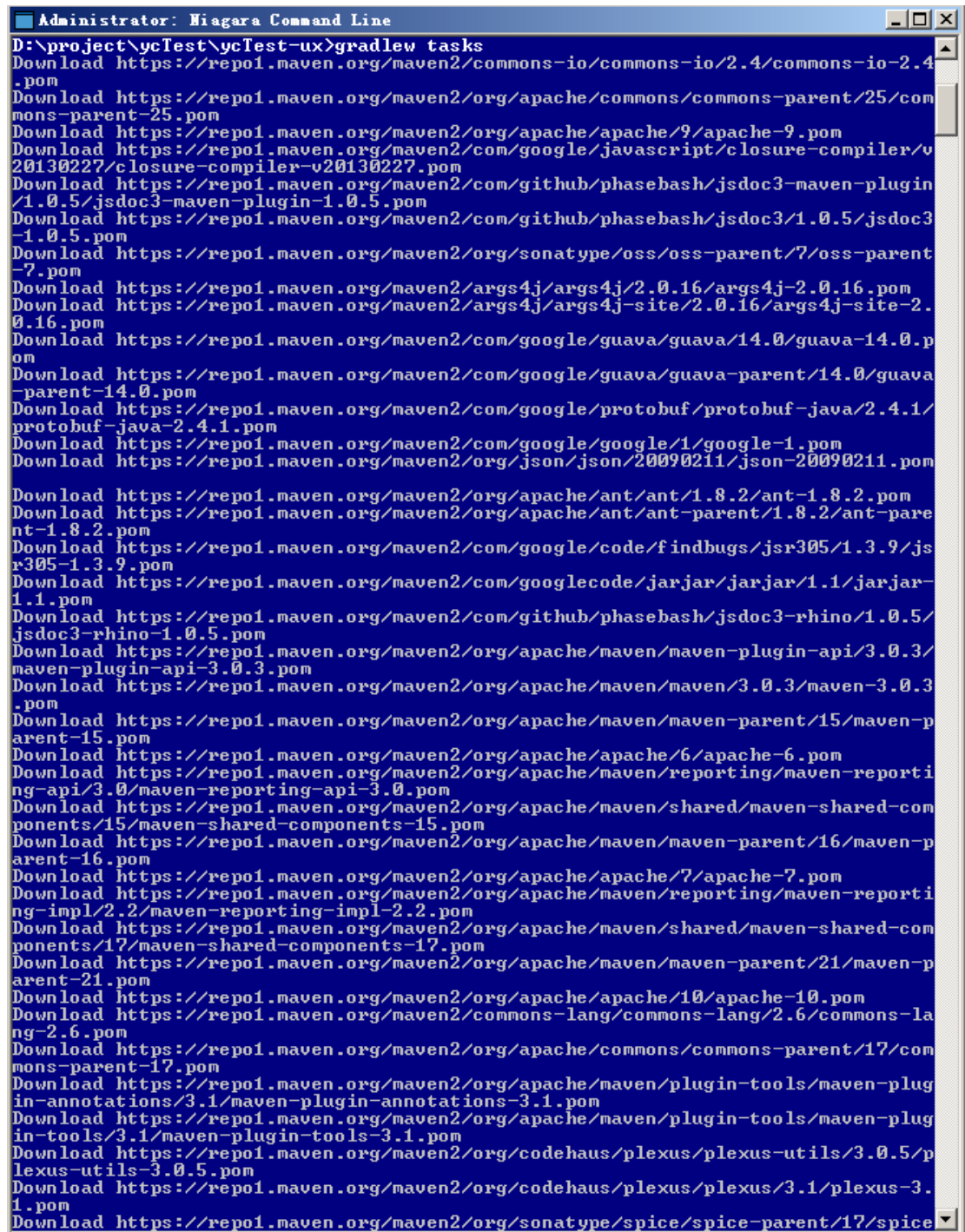
5. 进入 N4 安装目录里的 bin 目录，运行它下面的 console.exe 命令，建议将这个做成快捷方式放到桌面，后续教程里命令行都是通过它执行的。
6. 在上条所述控制台中运行命令：gradlew 。它会自动开始下载 gradle 软件（很慢），此处建议开 VPN 翻墙下（慢），或者使用迅雷按照命令里提示的网址下载好 gradle 的 zip（不用翻墙飞快），然后用 Ctrl+c 终止当前下载任务，然后找到 C:\Users\Administrator\.gradle\dists\gradle 版本号目录下的那个随机数字目录下（黄字部分根据登陆进来的用户名更改），你会看到下载的临时文件，把你下载好的安装包拷进来，然后重新运行 gradlew，可以看到很快



7. 进入 N4，建立一个新模块工程。



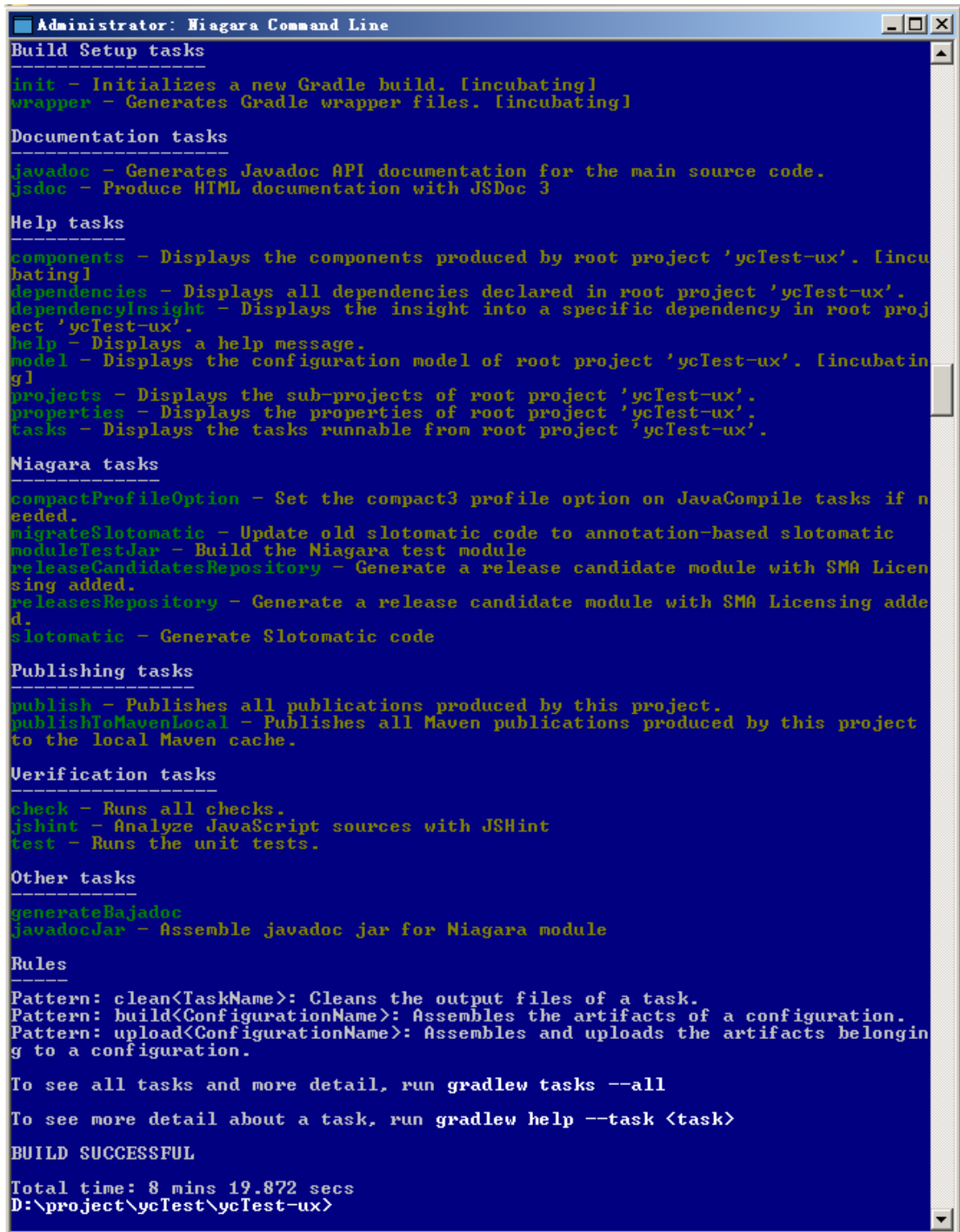
8. 使用 N4 的控制台，进入你的模块工程内，运行命令：gradlew tasks（注意此处建议开翻墙 VPN!!! 不然会超慢，或者干脆完成不了）
9. 开始第一次运行 gradlew，此处会安装 N 多东西，不开翻墙 VPN 的话会等非常久还未必能好。第一次运行安装玩后，以后不会再安装。



```
Administrator: Niagara Command Line
D:\project\ycTest\ycTest-ux>gradlew tasks
Download https://repo1.maven.org/maven2/commons-io/commons-io/2.4/commons-io-2.4
.pom
Download https://repo1.maven.org/maven2/org/apache/commons/commons-parent/25/com
mons-parent-25.pom
Download https://repo1.maven.org/maven2/org/apache/apache/9/apache-9.pom
Download https://repo1.maven.org/maven2/com/google/javascript/closure-compiler/v
20130227/closure-compiler-v20130227.pom
Download https://repo1.maven.org/maven2/com/github/phasebash/jsdoc3-maven-plugin
/1.0.5/jsdoc3-maven-plugin-1.0.5.pom
Download https://repo1.maven.org/maven2/com/github/phasebash/jsdoc3/1.0.5/jsdoc3
-1.0.5.pom
Download https://repo1.maven.org/maven2/org/sonatype/oss/oss-parent/7/oss-parent
-7.pom
Download https://repo1.maven.org/maven2/args4j/args4j/2.0.16/args4j-2.0.16.pom
Download https://repo1.maven.org/maven2/args4j/args4j-site/2.0.16/args4j-site-2.
0.16.pom
Download https://repo1.maven.org/maven2/com/google/guava/guava/14.0/guava-14.0.p
om
Download https://repo1.maven.org/maven2/com/google/guava/guava-parent/14.0/guava
-parent-14.0.pom
Download https://repo1.maven.org/maven2/com/google/protobuf/protobuf-java/2.4.1/
protobuf-java-2.4.1.pom
Download https://repo1.maven.org/maven2/com/google/google/1/google-1.pom
Download https://repo1.maven.org/maven2/org/json/json/20090211/json-20090211.pom

Download https://repo1.maven.org/maven2/org/apache/ant/ant/1.8.2/ant-1.8.2.pom
Download https://repo1.maven.org/maven2/org/apache/ant/ant-parent/1.8.2/ant-pare
nt-1.8.2.pom
Download https://repo1.maven.org/maven2/com/google/code/findbugs/jsr305/1.3.9/js
r305-1.3.9.pom
Download https://repo1.maven.org/maven2/com/googlecode/jarjar/jarjar/1.1/jarjar-
1.1.pom
Download https://repo1.maven.org/maven2/com/github/phasebash/jsdoc3-rhino/1.0.5/
jsdoc3-rhino-1.0.5.pom
Download https://repo1.maven.org/maven2/org/apache/maven/maven-plugin-api/3.0.3/
maven-plugin-api-3.0.3.pom
Download https://repo1.maven.org/maven2/org/apache/maven/maven/3.0.3/maven-3.0.3
.pom
Download https://repo1.maven.org/maven2/org/apache/maven/maven-parent/15/maven-p
arent-15.pom
Download https://repo1.maven.org/maven2/org/apache/apache/6/apache-6.pom
Download https://repo1.maven.org/maven2/org/apache/maven/reporting/maven-reporti
ng-api/3.0/maven-reporting-api-3.0.pom
Download https://repo1.maven.org/maven2/org/apache/maven/shared/maven-shared-com
ponents/15/maven-shared-components-15.pom
Download https://repo1.maven.org/maven2/org/apache/maven/maven-parent/16/maven-p
arent-16.pom
Download https://repo1.maven.org/maven2/org/apache/apache/7/apache-7.pom
Download https://repo1.maven.org/maven2/org/apache/maven/reporting/maven-reporti
ng-impl/2.2/maven-reporting-impl-2.2.pom
Download https://repo1.maven.org/maven2/org/apache/maven/shared/maven-shared-com
ponents/17/maven-shared-components-17.pom
Download https://repo1.maven.org/maven2/org/apache/maven/maven-parent/21/maven-p
arent-21.pom
Download https://repo1.maven.org/maven2/org/apache/apache/10/apache-10.pom
Download https://repo1.maven.org/maven2/commons-lang/commons-lang/2.6/commons-la
ng-2.6.pom
Download https://repo1.maven.org/maven2/org/apache/commons/commons-parent/17/com
mons-parent-17.pom
Download https://repo1.maven.org/maven2/org/apache/maven/plugin-tools/maven-plug
in-annotations/3.1/maven-plugin-annotations-3.1.pom
Download https://repo1.maven.org/maven2/org/apache/maven/plugin-tools/maven-plug
in-tools/3.1/maven-plugin-tools-3.1.pom
Download https://repo1.maven.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/p
lexus-utils-3.0.5.pom
Download https://repo1.maven.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.
1.pom
Download https://repo1.maven.org/maven2/org/sonatype/spice/spice-parent/17/spice
```

10. 安装好后列出可执行的 gradlew 任务。



```
Administrator: Niagara Command Line
Build Setup tasks
-----
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]

Documentation tasks
-----
javadoc - Generates Javadoc API documentation for the main source code.
jsdoc - Produce HTML documentation with JSDoc 3

Help tasks
-----
components - Displays the components produced by root project 'ycTest-ux'. [incubating]
dependencies - Displays all dependencies declared in root project 'ycTest-ux'.
dependencyInsight - Displays the insight into a specific dependency in root project 'ycTest-ux'.
help - Displays a help message.
model - Displays the configuration model of root project 'ycTest-ux'. [incubating]
projects - Displays the sub-projects of root project 'ycTest-ux'.
properties - Displays the properties of root project 'ycTest-ux'.
tasks - Displays the tasks runnable from root project 'ycTest-ux'.

Niagara tasks
-----
compactProfileOption - Set the compact3 profile option on JavaCompile tasks if needed.
migrateSlotomatic - Update old slotomatic code to annotation-based slotomatic
moduleTestJar - Build the Niagara test module
releaseCandidatesRepository - Generate a release candidate module with SMA Licensing added.
releasesRepository - Generate a release candidate module with SMA Licensing added.
slotomatic - Generate Slotomatic code

Publishing tasks
-----
publish - Publishes all publications produced by this project.
publishToMavenLocal - Publishes all Maven publications produced by this project to the local Maven cache.

Verification tasks
-----
check - Runs all checks.
jshint - Analyze JavaScript sources with JSHint
test - Runs the unit tests.

Other tasks
-----
generateBajadoc
javadocJar - Assemble javadoc jar for Niagara module

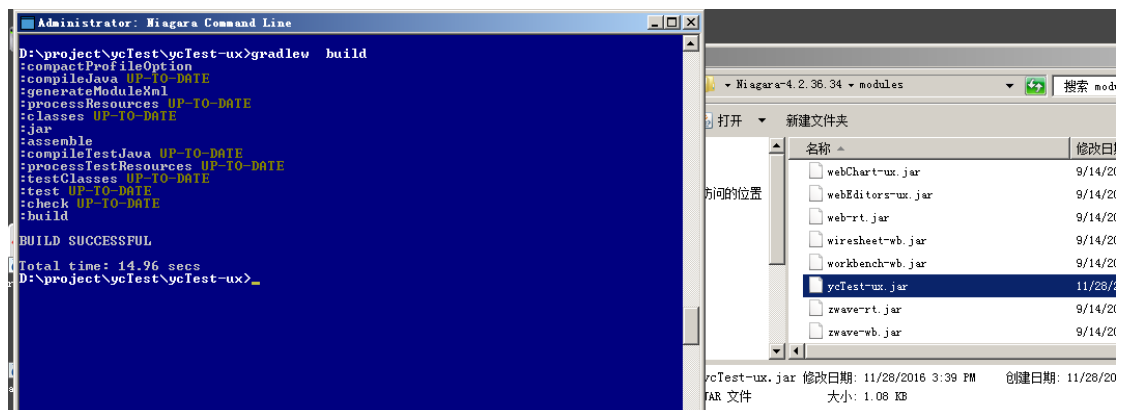
Rules
-----
Pattern: clean<TaskName>: Cleans the output files of a task.
Pattern: build<ConfigurationName>: Assembles the artifacts of a configuration.
Pattern: upload<ConfigurationName>: Assembles and uploads the artifacts belonging to a configuration.

To see all tasks and more detail, run gradlew tasks --all
To see more detail about a task, run gradlew help --task <task>

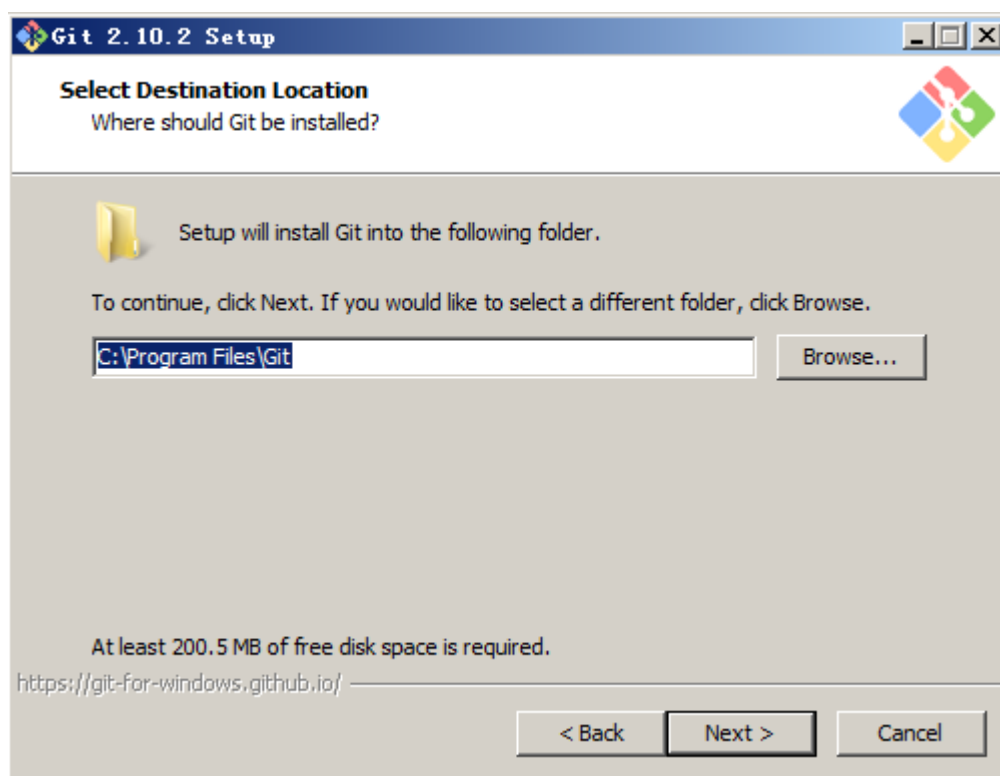
BUILD SUCCESSFUL

Total time: 8 mins 19.872 secs
D:\project\ycTest\ycTest-ux>
```

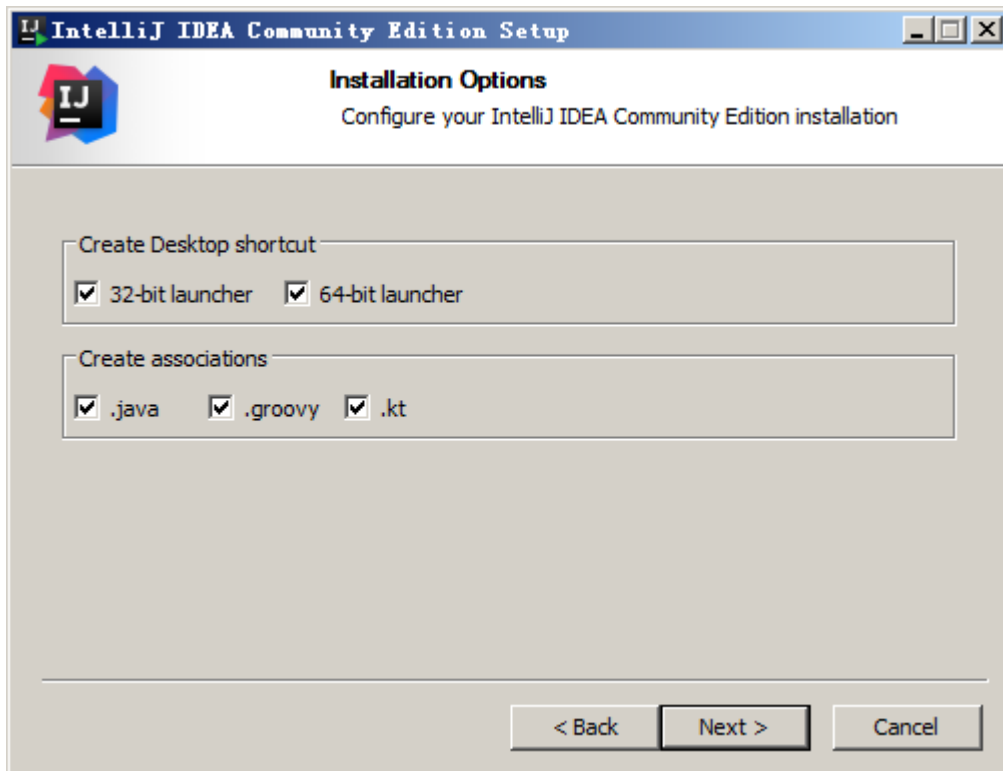
11. 运行命令：gradlew build 。会看到 N4 目录下 Module 目录下多出了新建的 Module 的 JAR 包，说明 gradle 运行良好。



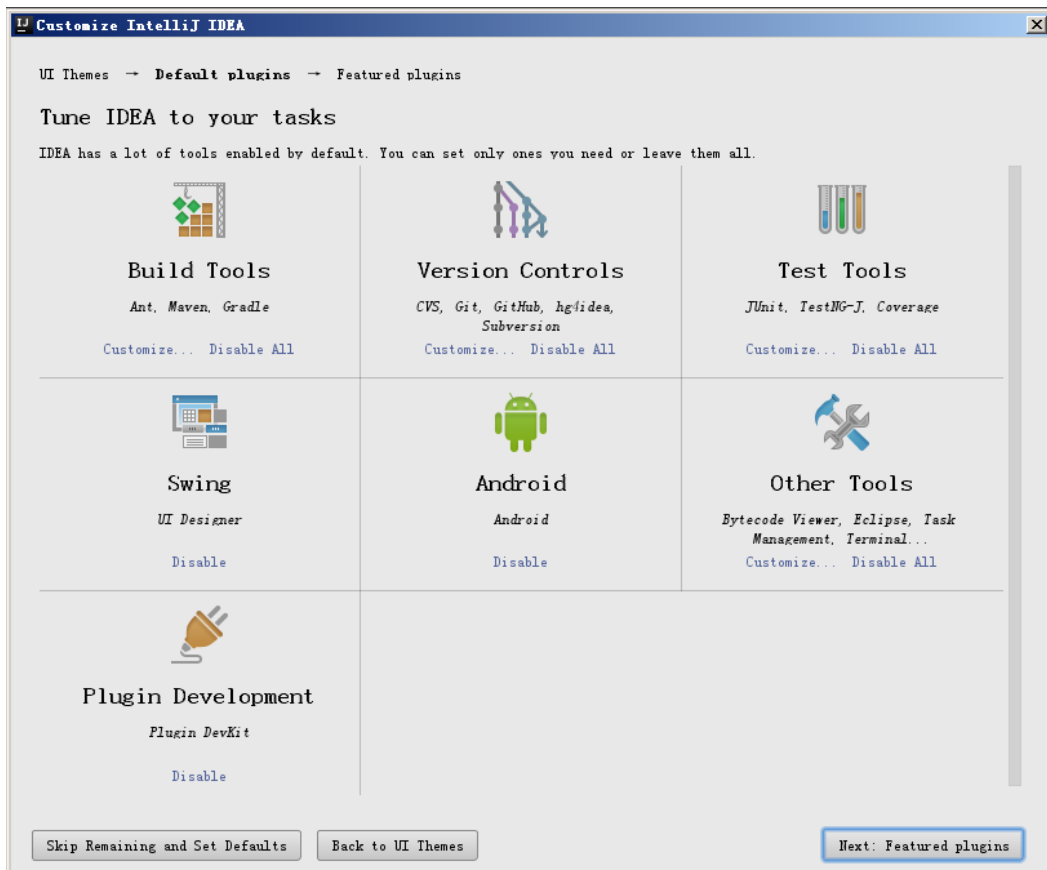
12. 如上就已经设置好最基本的开发环境了，如果你是纯文本编辑器及命令行的忠实爱好者的话，环境就已经设置完毕了，不过对我们一众普通人来说，还是需要 IDE 环境来开发代码的，N4 可以使用 Eclipse 或 IntelliJIDEA 来开发，官方说和 IntelliJIDEA 更搭配，下面我就 IntelliJIDEA 的安装进行图解。
13. 安装 GIT，参考 <https://git-scm.com/downloads>



14. 安装 IntelliJ IDEA, 参考 <http://www.jetbrains.com/idea/>, 有社区版（免费）和收费版，社区版可满足开发，此处使用社区版做演示。



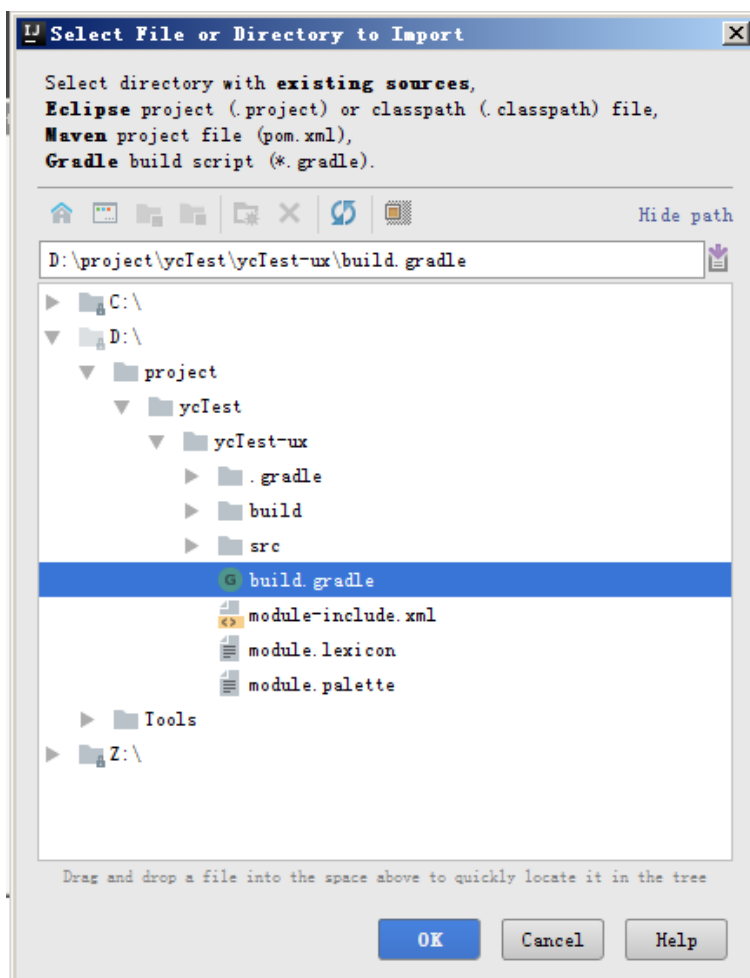
15. 第一次启动，同意默认设置。



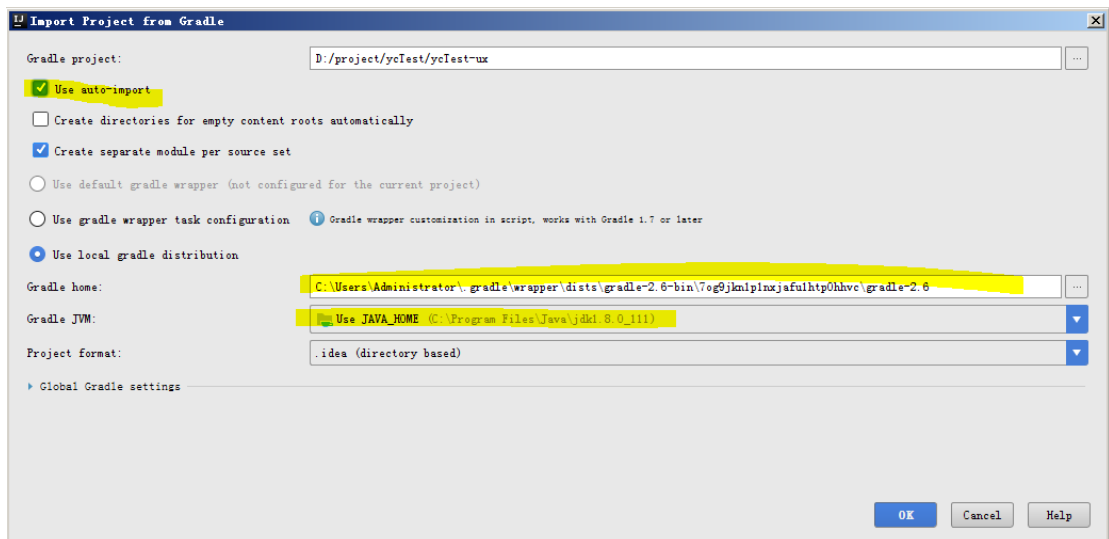
16. 选择引入项目，此处也建议开翻墙 VPN，因为引入好第一次启动 gradle 时 IntelliJIDEA 也会像第 9 条所述一样下载 N 多配件。



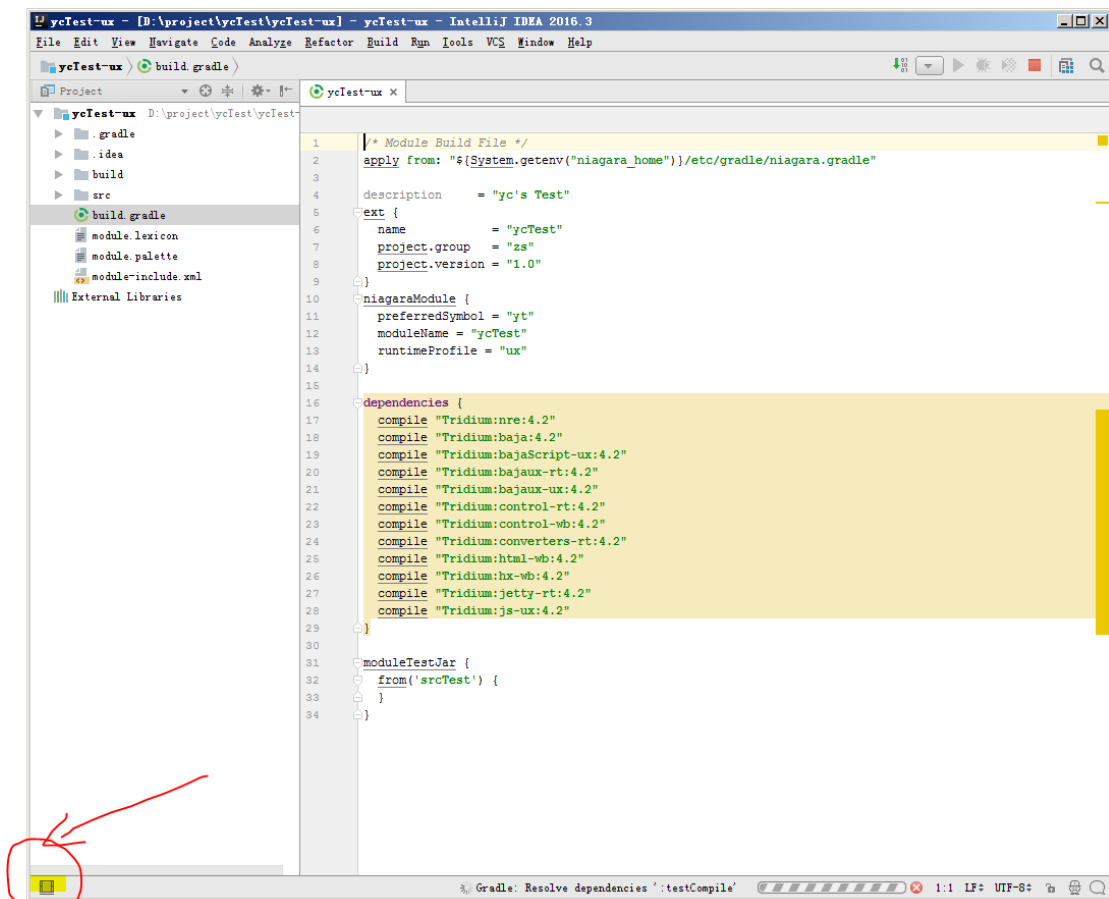
17. 选择模块项目下的 build.gradle 文件，则 IntelliJIDEA 会按照 gradle 项目的设置来引入项目。（官方推 IntelliJIDEA 的主要原因之一）



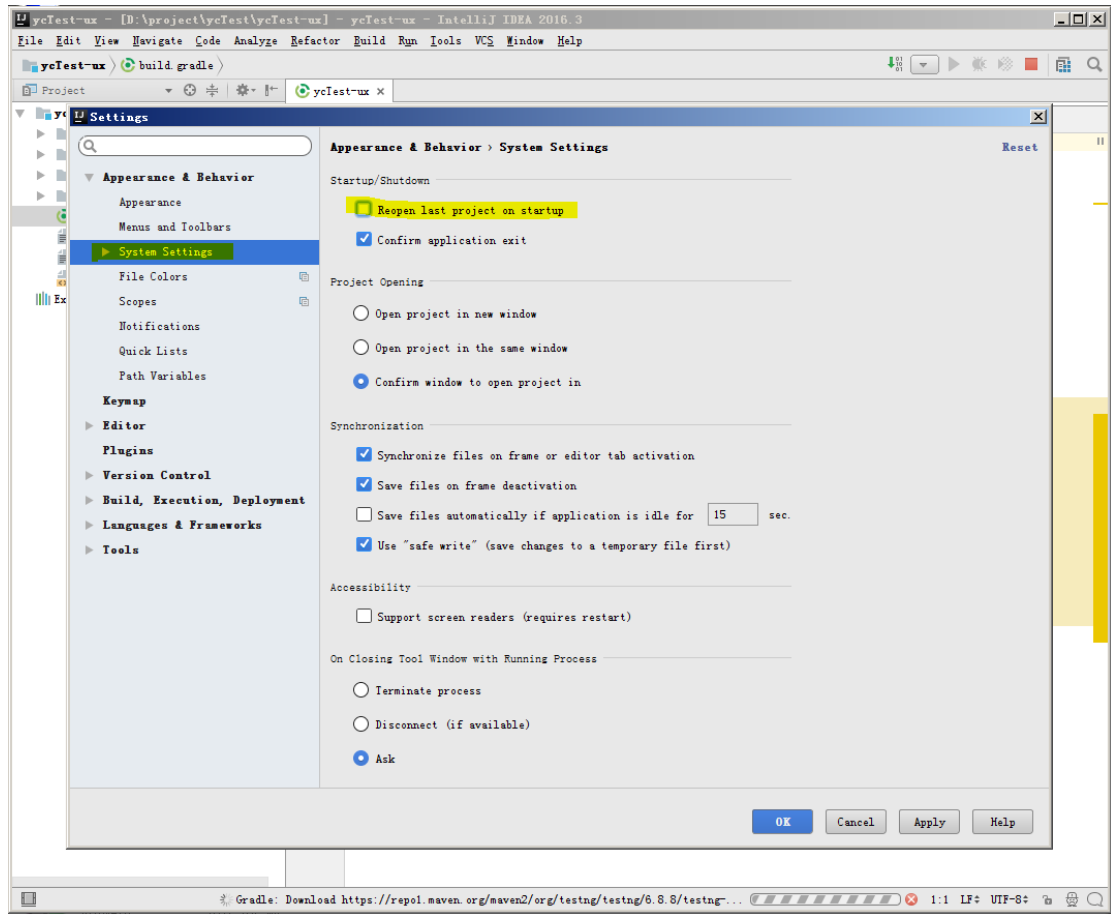
18. 引入参数设置，记得一定要勾上 use auto import 选项，这个和自动完成有非常大关系（用 IDE 最大的帮助就是自动完成功能）。还有，第一次引入 gradle 项目时，gradle 程序位置也需选择。还有，如果前述 4. d 的 JAVA_HOME 参数没设置时，此处也必须要设置。



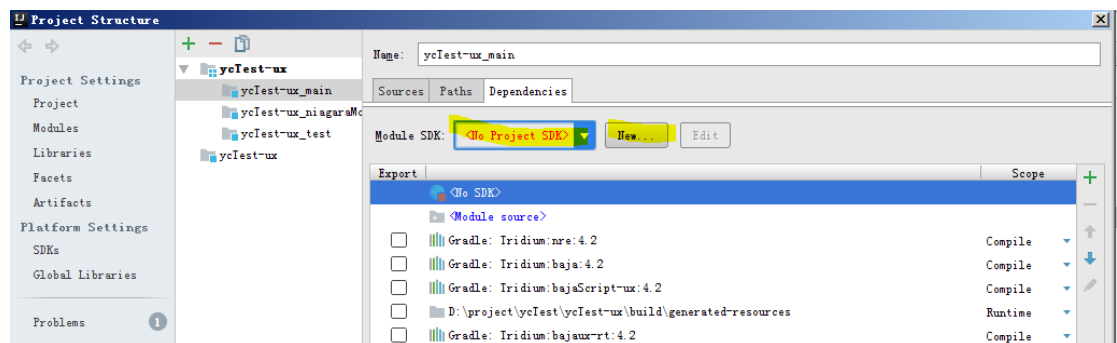
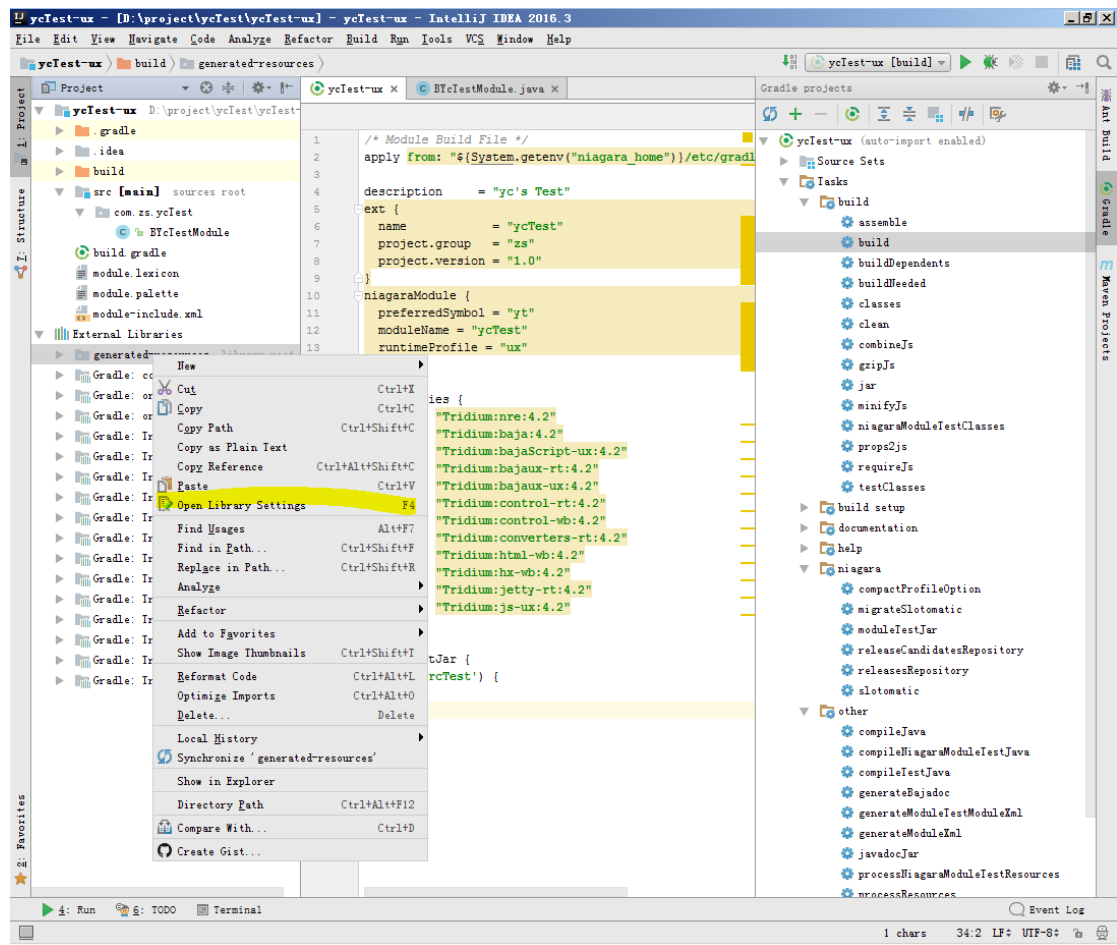
19. 开工具栏是要点击一下 IntelliJ IDEA 的左下角的一个按钮。

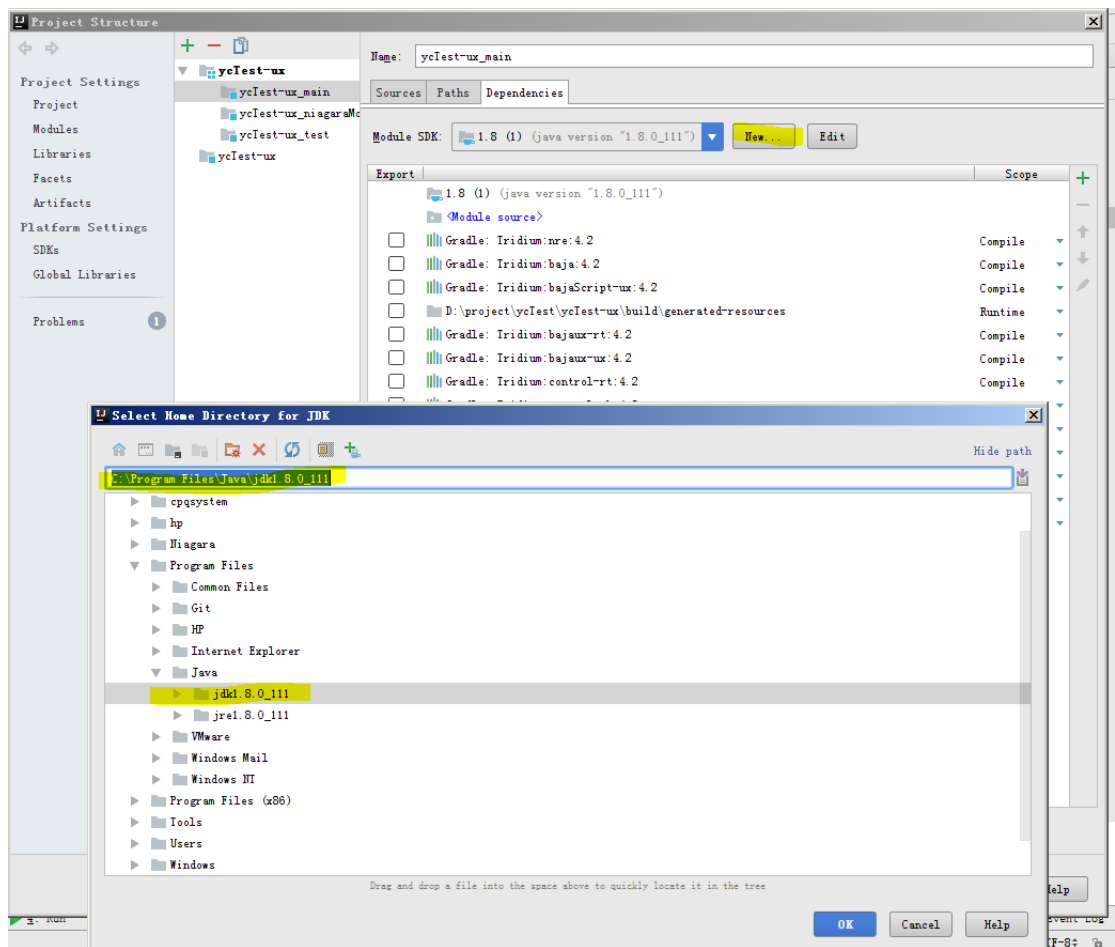


20. 如果有多个项目，则进去后选 File->setting 菜单，在 Sysem Settings 中关闭 reopen last ...选项，这样在每次开 IntelliJIDEA 时可以选项目打开。

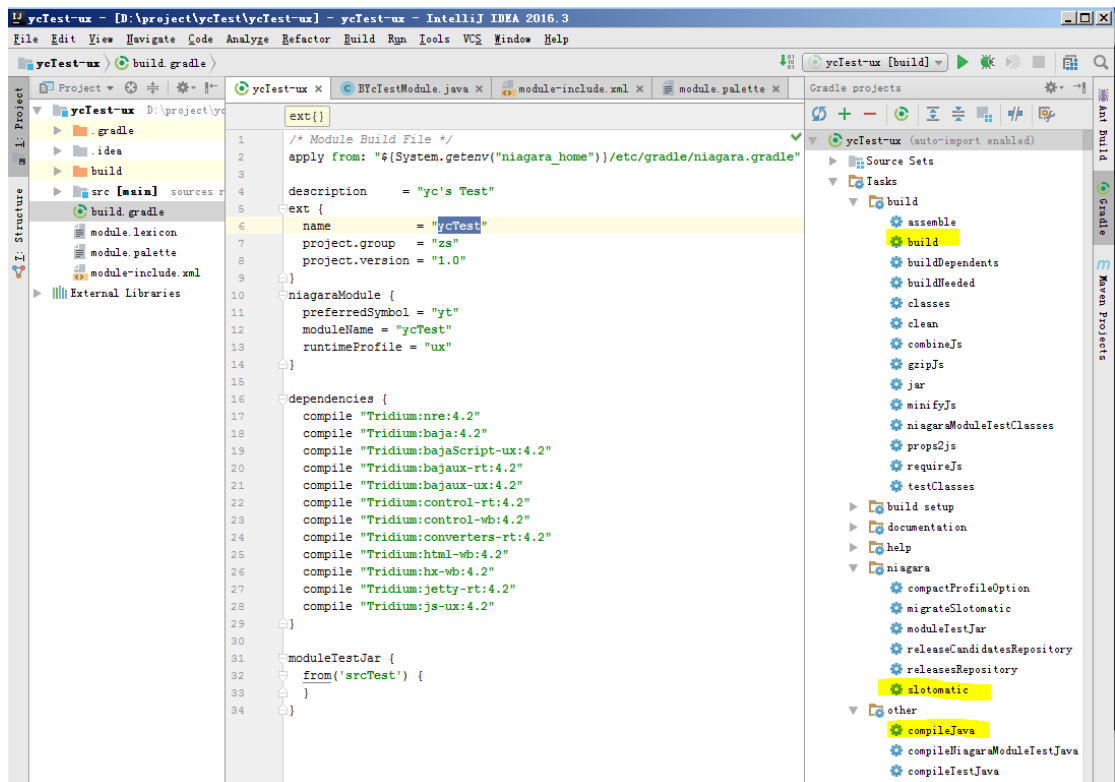


21. 因为是第一次运行，要设置 JDK 库，在左侧项目文件目录夹下的库目录中右击，选 open library settings 菜单，增加 JDK。增加后，build.gradle 下的黄色警告应该会消掉。

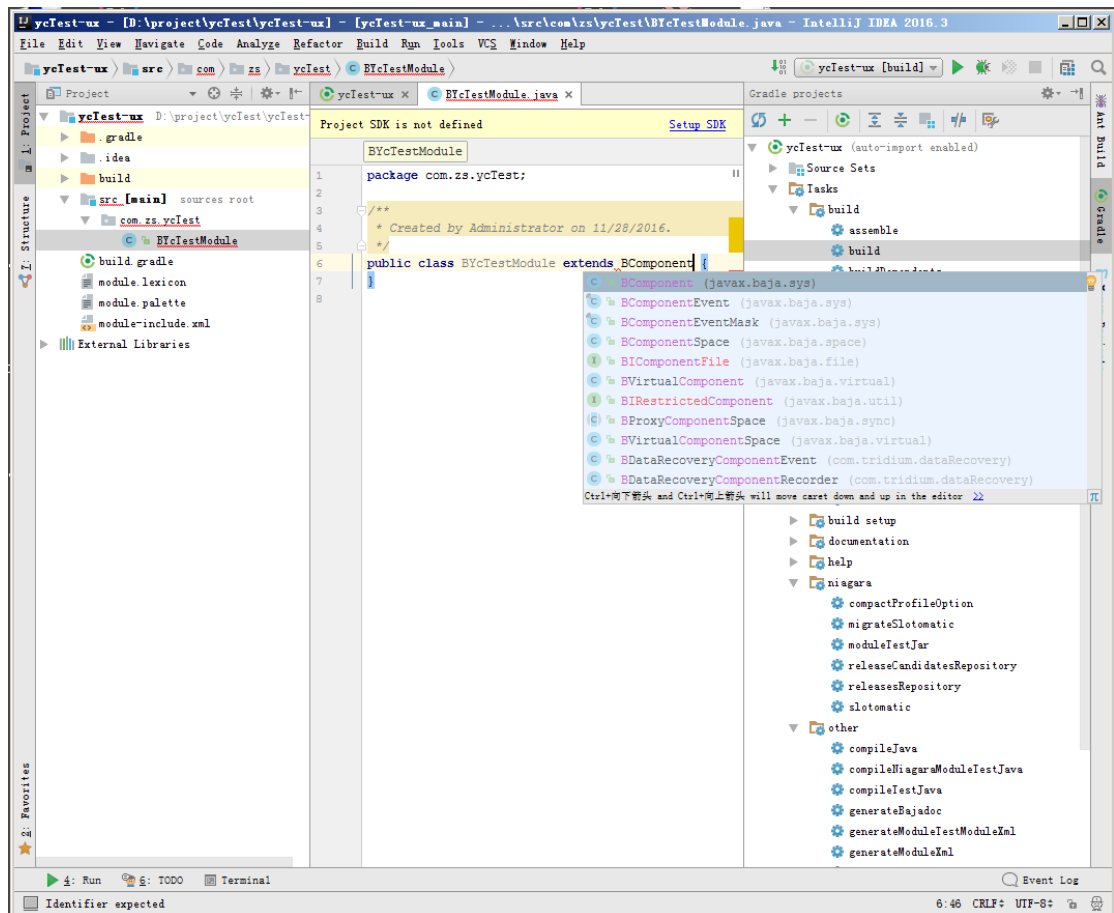




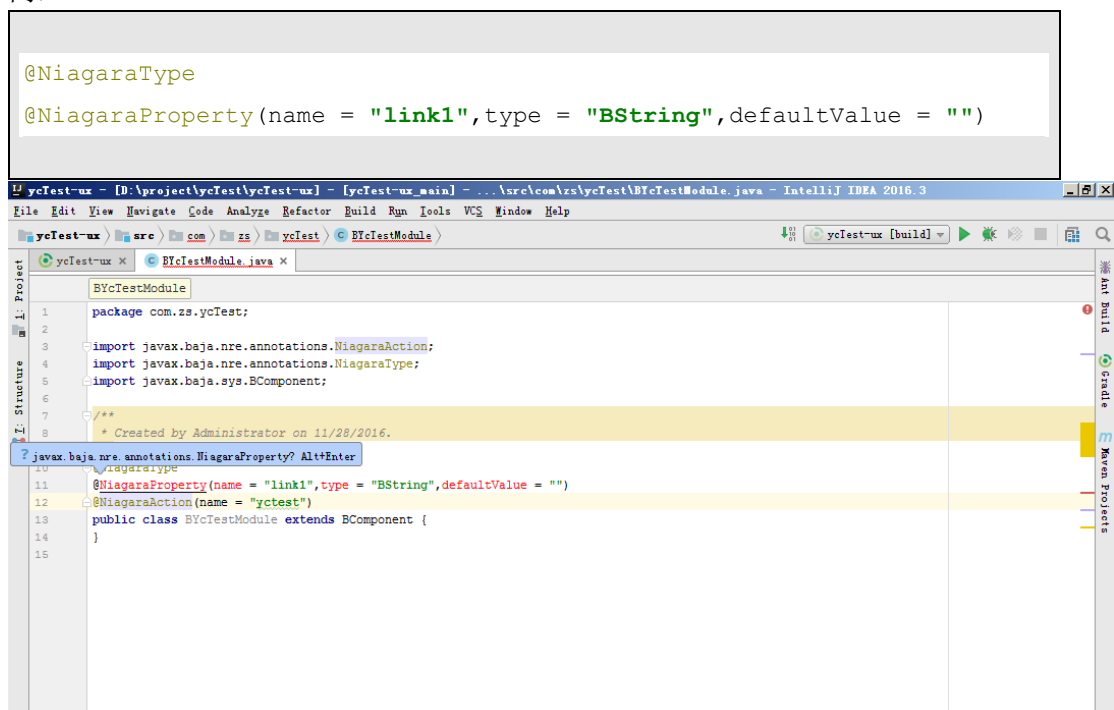
22. 打开右侧的 gradle 窗口，会看到此项目可用的 gradle 任务，最主要的就是 build、slotomatic、compileJava 三项命令。



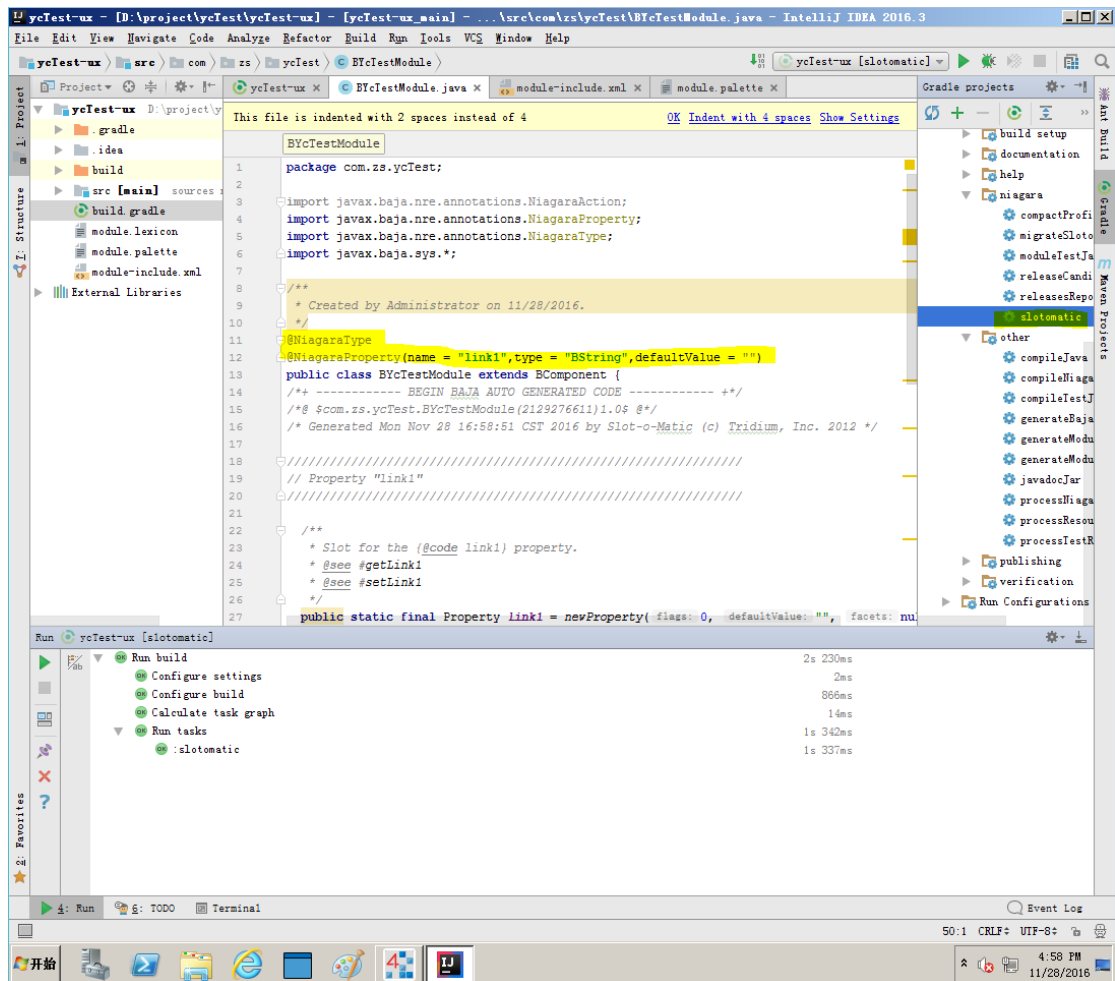
23. 建立一个新类，然后 extends 时就可以看到自动完成提示了^_^



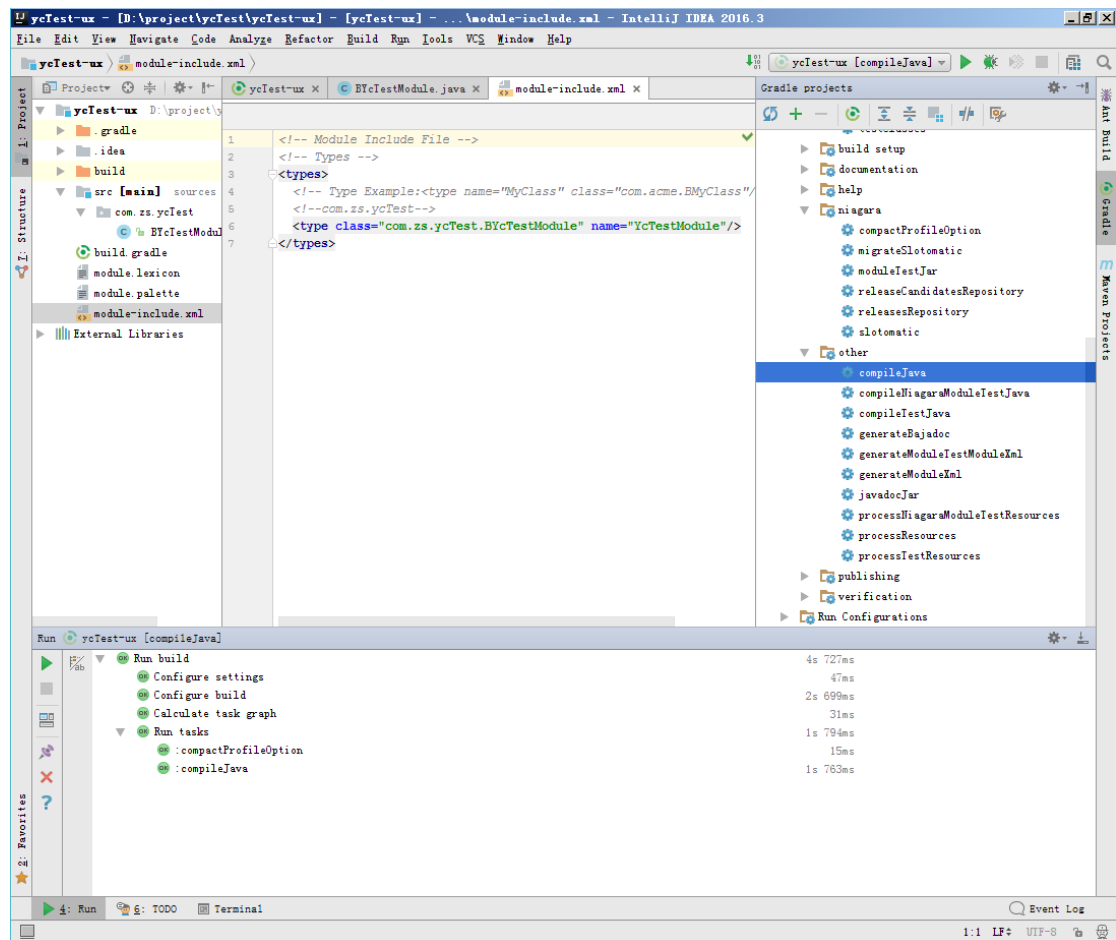
24. Slotomatic 在 N4 中发生很大变化，其格式可常见官方文档 slotomatic 章节，此处在此类上方添加两句。在提示红字的变量上点击 Alt+Enter，会自动 import 相关包（后同）。



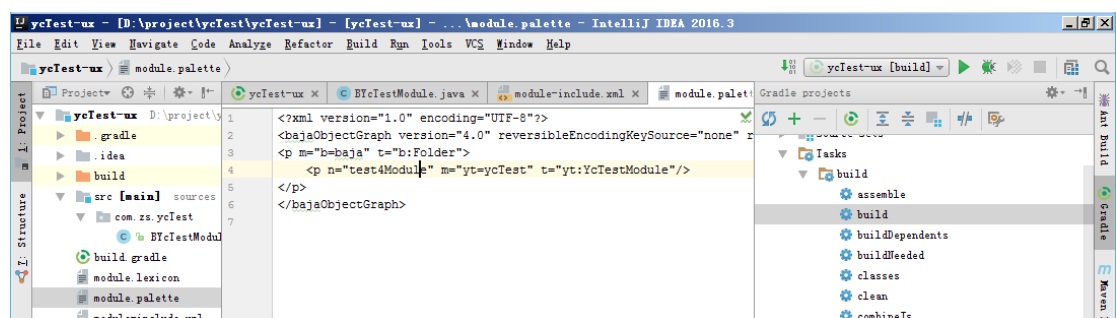
25. 运行 slotomatic, 你会发现自动生成代码出来了.



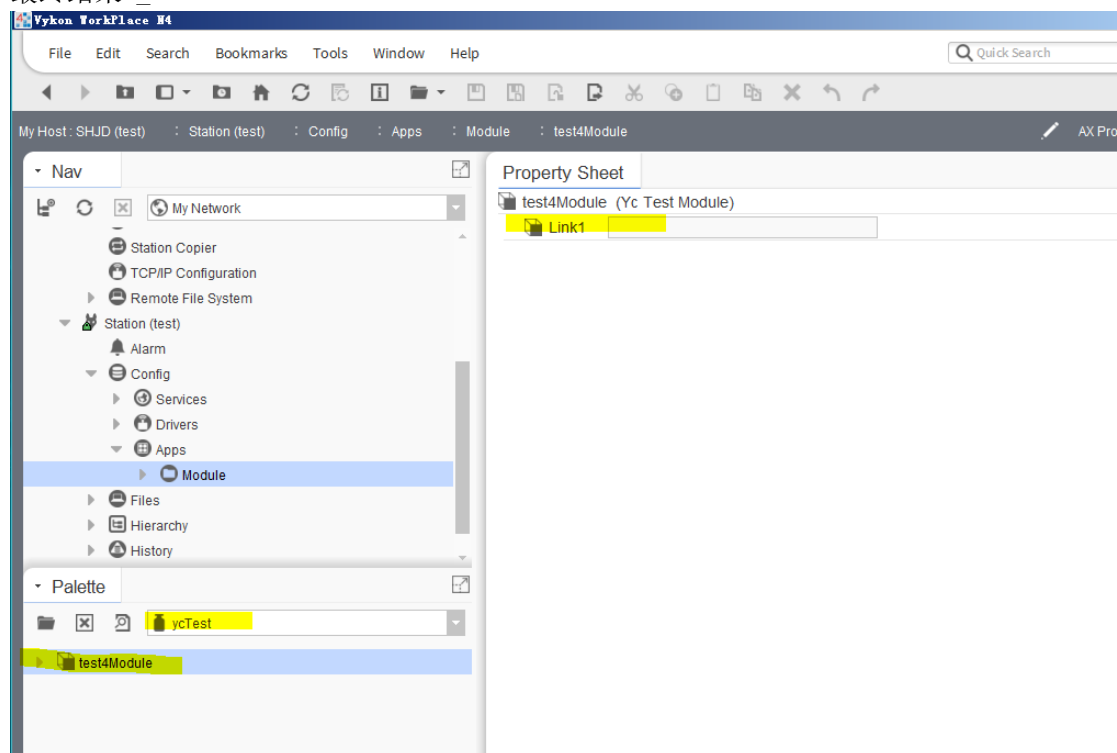
26. N4 中的 slotomatic 不会在 module-include.xml 中生成相关项，这个现在变成了 compileJava 任务的工作，当然你也可以直接 build，它中间包含了 compileJava 任务。



27. 添加 palte 文件中的项，build。



28. 最终结果^_^



以上就是 Tridium 的基础开发环境设置过程，后续还会有 N4 的 Javascript 开发环境设置过程详细文档，敬请期待^_^