您现在使用的CHM文件由Easy CHM的未注册版本制作而成。

注册版本不会在CHM里自动添加此页面。

制作时使用的Easy CHM程序版本号：v3.93
制作日期：2015/11/10

Easy CHM 软件

软件出品: 国华软件

截屏图片

中国北京

软件类型: 共享软件

免费下载最新的试用版

文件大小: 2989k - 2.84MB

购买注册:

这里

关键词: make chm, create chm, chm制作工具, chm电子书制作工具,chm帮助文件制作工具

描述:

Easy CHM是国华软件推出的一款强大的CHM电子书或CHM帮助文件的快速制作工具。

使用EasyCHM只需要三个步骤就可以完成CHM的制作：

1、用户指定一个目录，EasyCHM会自动导入全部目录及文件;
2、设置CHM编译选项;
3、开始制作。

EasyCHM非常适合个人和单位制作**高压缩比**的**带有全文检索**及**高亮显示搜索结果**的网页集锦、CHM帮助文件、产品说明、公司介绍、CHM电子书等等。

主要功能：

- 全自动的目录及文件导入（可以包括子目录）;
- EasyCHM支持导入任意的文件类型;
- EasyCHM操作速度快，性能稳定，EasyCHM因为上手容易深得广大用户好评;
- EasyCHM自动生成CHM的目录列表并自动生成所有目录项;
- 为CHM的目录列表自动添加多级编号;
- 在编辑目录项、索引项时用到的移动、拖拽、替换等操作中**Easy CHM完全支持多选及拖拽操作**，避免了一项一项地操作，极大的减少用户手工 – 非常适合企业维护大型CHM帮助文档;
- 支持批量查找替换多级目录各项的标题文字内容;
- 允许用户指定从文本文件的第N行自动截取标题;
- 易用的目录/索引编辑器;

- 丰富实用的CHM制作选项帮助用户制作更加个性化的专业CHM电子书或CHM帮助文件;
- EasyCHM自动生成输出Alias和Map头文件;
- EasyCHM自动生成上下文相关的帮助文件（ContextID），适合于软件公司制作专业的支持Help Context ID的专业帮助文件;
- EasyCHM可以保存工程文件，方便企业用户编辑维护文档。
- 批量更换CHM目录各项的图标;
- 内嵌CHM反编译工具
- 更多功能

EasyCHM未注册版本的主要限制：

1. EasyCHM的未注册版本可以制作带任意层级目录的CHM，但是每个CHM的工具栏上都会自动添加一个ABOUT按钮，以提醒用户注册。
   注册版本不会添加ABOUT按钮。
2. EasyCHM的未注册版本无法保存CHM的通用工程文件，如*.HHP、*.HHC、*.HHK等。

   未注册版本的EasyCHM会自动删除*.HHP、*.HHC、*.HHK这几个文件。
3. 未注册版本制作的CHM标题上会有Made by UNREGISTERED version of Easy CHM这样的提示您试用的是未注册版本的文字。

注册版本无任何限制，也不会再显示ABOUT按钮或者提示注册的文字。
如果EasyCHM可以为您的工作和学习提供帮助，请注册购买，感谢您给予软件作者的支持。

联系方式:

国华软件站点一: http://www.etextwizard.com

国华软件站点二: http://www.zipghost.com

技术支持信箱: webmaster@etextwizard.com

购买方式:

在线购买:（可提供发票，支持很多种支付方式，包括在线支付、银行电汇、邮政汇款等）
通过“共享软件注册中心”购买单用户企业版

通过“共享软件注册中心”购买单用户家庭版

直接向国华软件注册购买（共3种方式）：

1、招商银行划款:

开户行： 招商银行北京分行双榆树支行

帐 号： 001015570093

户 名： 冯国华

2、中国农业银行金穗卡转帐（注意是卡号）：

金穗卡卡号： 95599 8001 42061 34216

收 款 人： 冯国华

3、中国工商银行牡丹灵通卡转帐：

开户城市：北京
牡丹灵通卡卡号：9558 8202 0000 3186295 （注意是卡号9558820200003186295）

收 款 人： 冯国华

注意事项：

直接向作者注册的用户请在汇出足额的注册费后以EMAIL形式通知作者，同时请尽量提供

- 汇款日期
- 汇出城市
- 您汇到哪个银行帐号？
- 汇款人姓名

- 用于接收注册码的信箱（请确保可以接收邮件，最好多提供）
- 您希望使用的注册用户名（程序将授权给这个用户）
- 要注册的软件名称及套数

您提供的资料详细些会比较方便确认。
作者在确认支付已完成后将会马上发出注册码。
用户在收到注册信后请尽量给软件作者发个类似“已收到注册码”的确认信 - 感谢您的配合！

免费下载最新的试用版:
http://www.etextwizard.com/cn/download/ecdown.html

系统需求: Win98, WinME, WinXP, Win2000,WinVista, Win7

安装支持: 支持安装及卸载

重要说明，看三遍

介绍(Introduce)

CHM 制作者：Judas.n：http://www.YouMeek.com
CHM 制作时间：2015-11-10

本教程首发在 Github 上，如果有需要请进行 Fork：

- 地址：https://github.com/judasn/IntelliJ-IDEA-Tutorial/

本系列教程历时两个多月，第一个版本已经完成，目前在不定时进行调整和补充，所以本电子书会存在信息过时的可能，需要关注更新的请在 Github 上 `Watch`、`Star`、`Fork`。

特别需要友情提醒的是：请 `Fork` 之后，在我的基础上按你自己喜欢的方式整理一套属于你自己的快捷键列表，并导出为 PDF，以备不时查阅，对于提升你开发效率很有帮助！文章的图片建议在需要的时候 右键 - 查看图像（在新标签页打开图片） 进行原图查看。

同时也邀请您一起参与完善该教程，欢迎您反馈我的错误和意见！！！

本系列文章唯一授权的商业网站是：极客学院，其他商业网站一律禁止转载。个人博客、公众号等载体请在转载写明出处链接。

如果你只是单纯要阅读的话，建议移步极客学院上观看，访问速度快很多：

- 地址：http://wiki.jikexueyuan.com/project/intellij-idea-tutorial/

如果你想参与完善该教程，请移步到 Github 上进行 Fork：

- 地址：https://github.com/judasn/IntelliJ-IDEA-Tutorial/

联系(Contact)

Judas.n

- Email：judas.n@qq.com（常用） Or jn3.141592654@gmail.com（备用）
- Blog：http://www.YouMeek.com
- IntelliJ IDEA QQ 群：244908708，入群请看：http://www.youmeek.com/idea
- 欢迎捐赠^_^：http://www.youmeek.com/donate

参与完善教程(Participate)

如果您不会使用 Git 或是 Github 也没关系，请认真学习下面三个视频教程：

- @Markdown 介绍+语法+笔记+常用工具推荐(视频教程)
- @IntelliJ IDEA：Git 和 Github 专讲
- @IntelliJ IDEA：Github 协同合作全流程(视频教程)

Github 常用按钮说明

- Watch：关注该项目，作者有更新的时候，会在你的 Github 主页有通知消息。
- Star：收藏该项目，在你的头像上有一个“Your stars”链接，可以看到你的收藏列表。
- Fork：复制一份项目到的Github空间上，你可以自己开发自己的这个地址项目，然后 Pull Request 给项目原主人。

参与作者汇总

作者(按参与时间排序)	地址
Judas.n	http://www.YouMeek.com
温泉	https://github.com/wenquan0hf
zhenhappy	https://github.com/zhenhappy

关于

学习前提

由于 IntelliJ IDEA 官网在亚洲没有设服务器，且官网用到一些类似 Twitter、Facebook 等站的脚本会使得你在国内出现访问巨慢或是不允许访问的特殊情况，所以建议你在访问官网、访问插件库、小版本本地迭代更新等操作的时候出现奇怪问题的时候，请自备VPN等网络加速工具。

很多用户都是先学习了 Eclipse、MyEclipse 再转到 IntelliJ IDEA 的，这里需要先说明的是，在学习 IntelliJ IDEA 过程中，你暂且要放下 Eclipse 下的开发思维方式，不能按 Eclipse 的软件思想或是结构去要求 IntelliJ IDEA，这样对你学习 IntelliJ IDEA 非常不利。

适用人群

用 IntelliJ IDEA 进行开发语言的学习者。

用 IntelliJ IDEA 进行开发语言的开发工作者。

其中对于语言开发学习者我是非常建议你使用 IntelliJ IDEA，因为一些代码格式、命名规范在 IntelliJ IDEA 下都是有良好的提示，对于我们所处的输入法下的中文全角符号也可以得到快速发现。特别是学习 Python 的学习者，当你在用 Pycharm 进行学习的时候，Pycharm 会时刻告诉你什么时候要注意空格、换行，提醒你有 PEP8 编码规范，你也可以通过快捷键快速格式化出适合 Python 要求的代码，这对于学习者来讲，真的很重要，它可以让你更专注于自己的代码。

教程演示的 IntelliJ IDEA 版本

IntelliJ IDEA 13 版本和 14 版本，在设置上差异很大，14 版本 IntelliJ IDEA 对整个 IDE 的设置进行了重新编排、归类，但是细节设置上所沿用的介绍是没有多大改变的。

目前（2015 年 06 月）IntelliJ IDEA 官网最新版本信息为：**Version:14.1.4 Build:141.1532.4 Released:June 19th, 2015**。

IntelliJ IDEA 有旗舰版和社区版本之分，本系列教程将以 14.1.4 的旗舰版进行演示和讲解。

其中旗舰版（Ultimate Edition）为收费版本，有 30 天试用期。如果你是学生、老师、开源项目参与者都可以向官网免费试用旗舰版，具体你可以查看下面链接。社区版（Community Edition）为免费版本，功能较旗舰版少了很多。

本教程使用的 IntelliJ IDEA 主题为较受欢迎的黑色：**Darcula**。

- 申请免费版本：https://www.jetbrains.com/idea/buy/
- 旗舰版和社区版差异细节：https://www.jetbrains.com/idea/features/editions_comparison_matrix.html

教程演示的系统环境

- 系统：Windows 8.1 64 位 简体中文版
- JDK 版本：1.8.0_05 64 位
- 建议使用 JDK 版本为：1.6 及 1.6 以上，更加详细的系统要求会在安装教程篇中进行讲解。

IntelliJ IDEA 版本迭代习惯

2015 年 IntelliJ IDEA 主版本是 14，目前（2015 年 06 月）最新版本是 14.1.4。与此同时，2015 年 06 月 17 日，官网开始提供 15 EAP 版本（Early Access Program 早期预览版）。如果你对 IntelliJ IDEA 下个大版本的新特性很感兴趣，你可以随时关注官网博客最新动态。

按正常情况来讲，IntelliJ IDEA 大版本是一年迭代一次。大版本下的小版本迭代时间没有固定，快的是一个月不到就迭代一次，慢的话基本在两到三个月迭代一次。相对其他 IDE 来讲迭代周期还是比较紧凑，但是作为用户你不用担心因为频繁迭代更新而引起的项目配置问题或是软件配置问题，后面有课程会专门对此进行说明。

- IntelliJ IDEA 官网博客：http://eap.jetbrains.com/idea

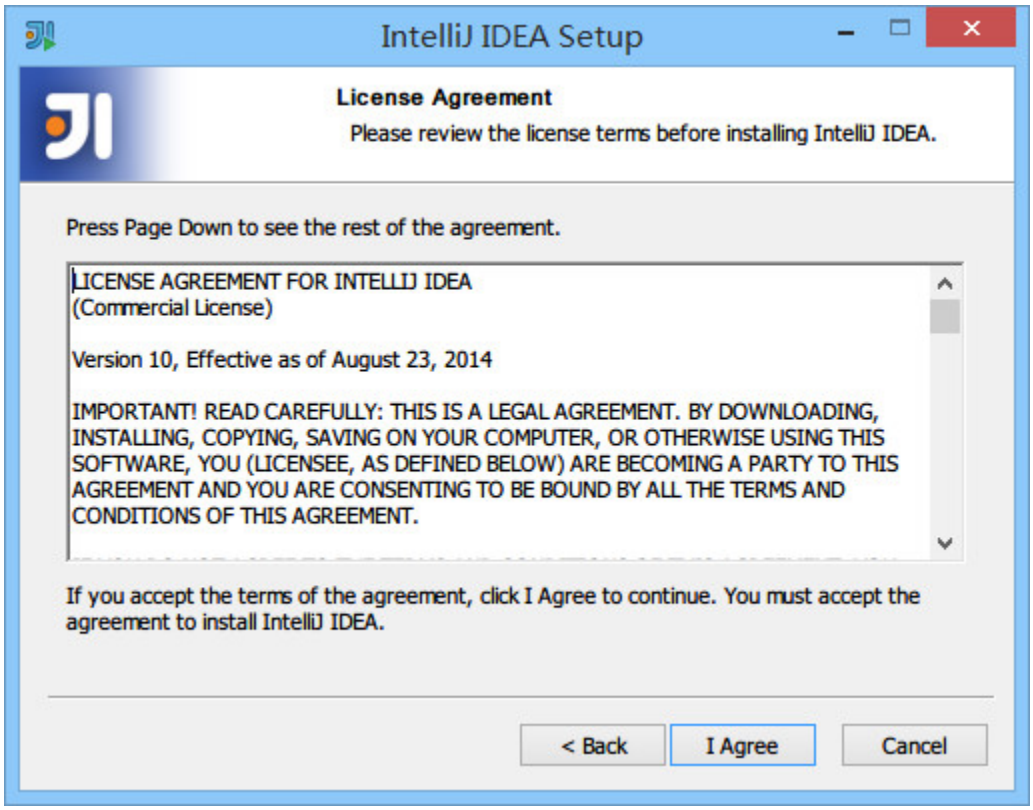
Windows 系统下安装 IntelliJ IDEA

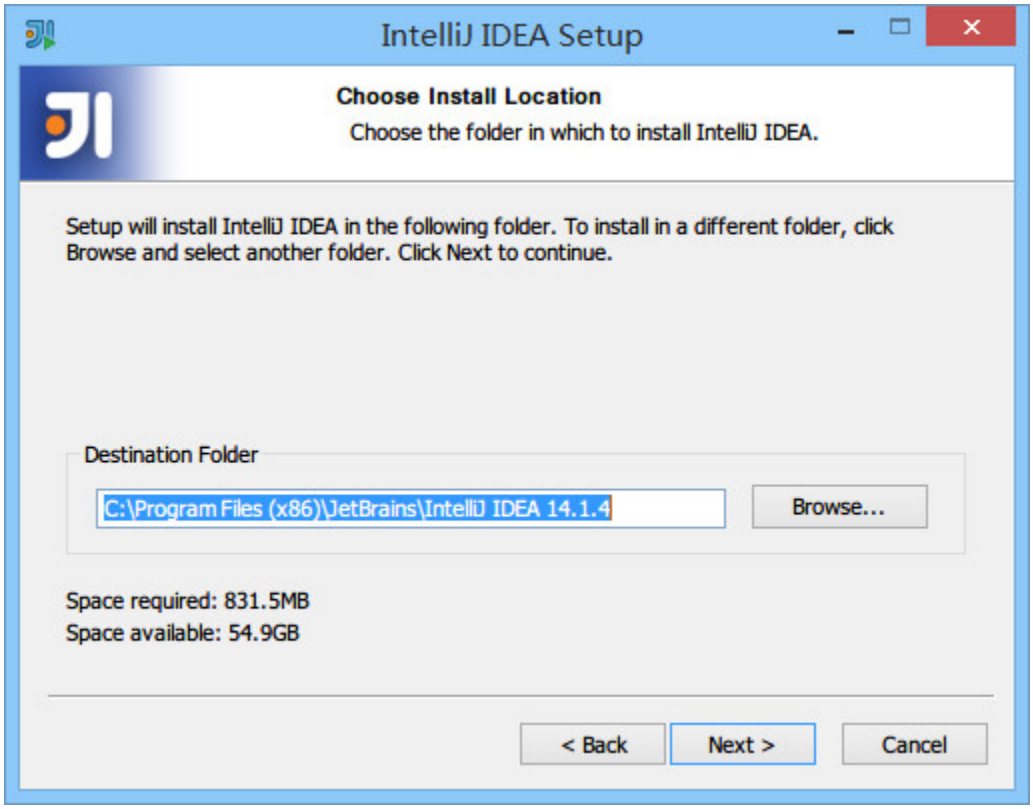
系统要求

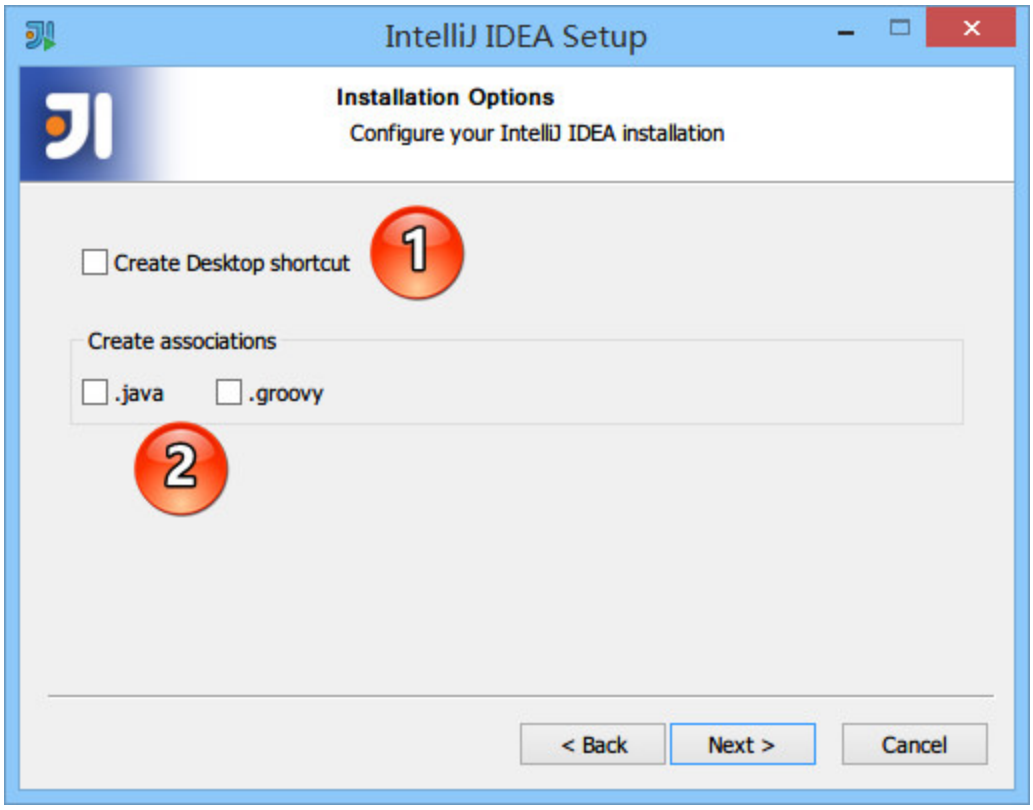
- 系统支持：Microsoft Windows 8 / 7 / Vista / 2003 / XP（每个系统版本的 32 位和 64 位都可以）
- JDK 版本：Oracle JDK 1.6 或以上
- 内存：最低要求 1 GB，推荐 2 GB 以上
- 硬盘：最低要求 2 GB
- 显示器：最低要求 1024 X 768 分辨率
- 更多信息可以阅读：https://www.jetbrains.com/idea/download/system_requirements.jsp?os=win

首次安装

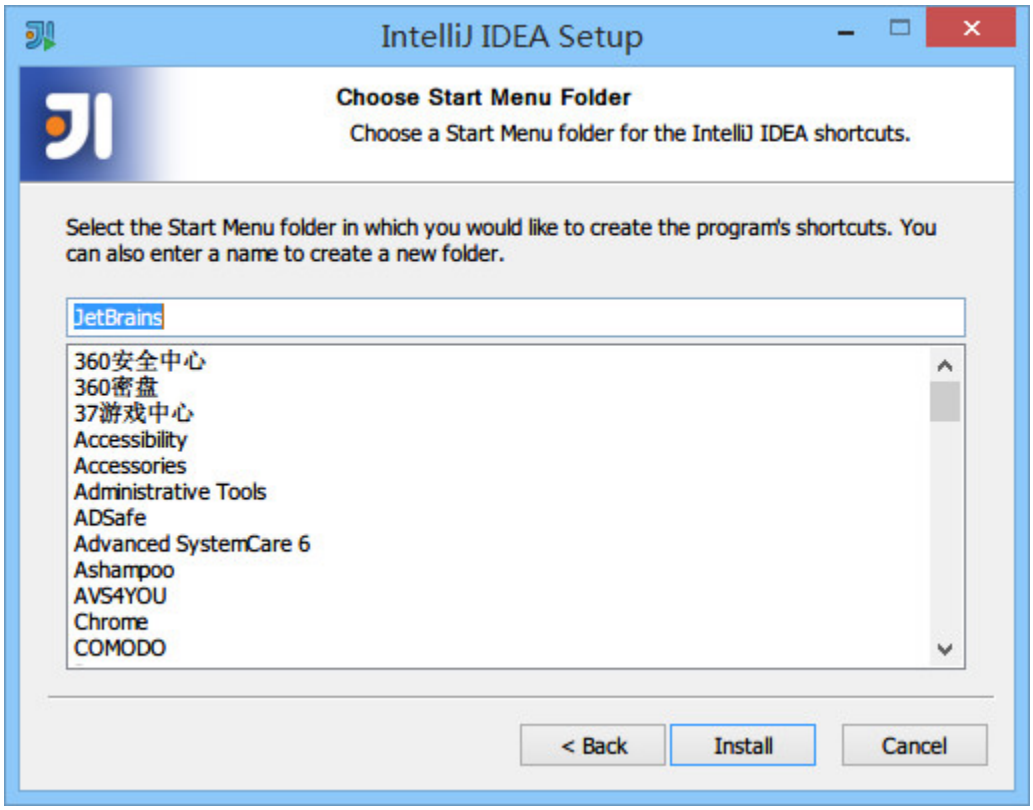
- IntelliJ IDEA 的安装是非常简单的，不需要做过多的选择，可以说简单到都是 `Next` 即可。

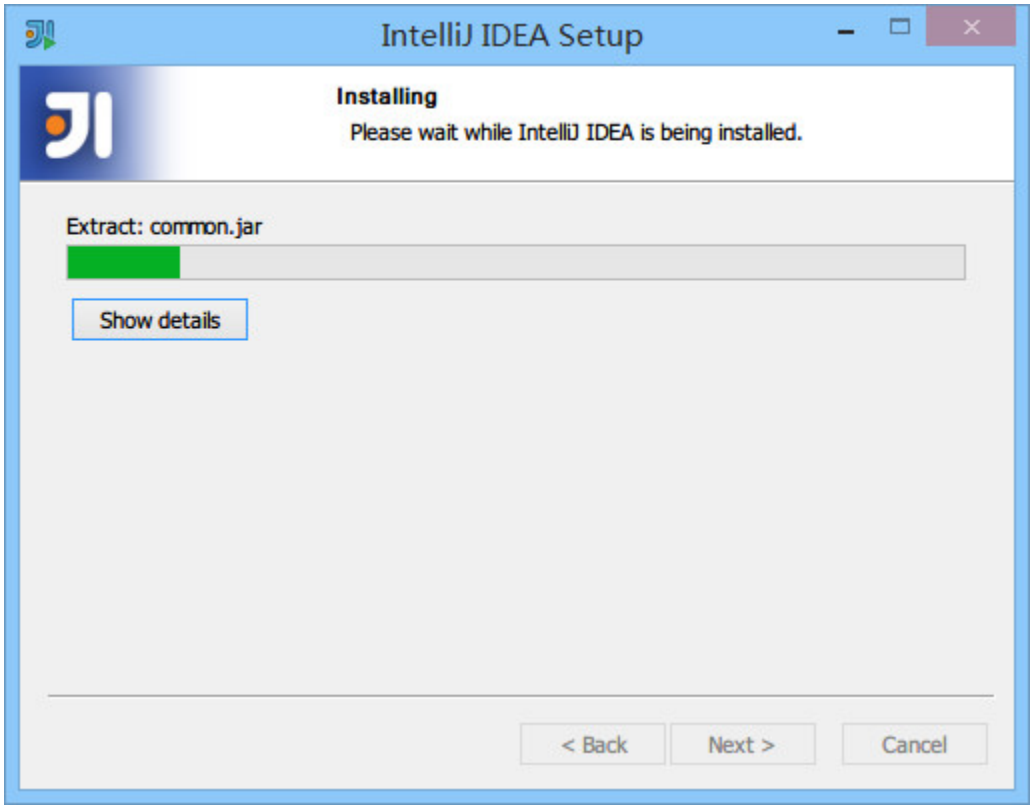

IntelliJ IDEA Setup
License Agreement
Please review the license terms before installing IntelliJ IDEA.
Press Page Down to see the rest of the agreement.
LICENSE AGREEMENT FOR INTELLIJ IDEA
(Commercial License)
Version 10, Effective as of August 23, 2014
IMPORTANT! READ CAREFULLY: THIS IS A LEGAL AGREEMENT. BY DOWNLOADING, INSTALLING, COPYING, SAVING ON YOUR COMPUTER, OR OTHERWISE USING THIS SOFTWARE, YOU (LICENSEE, AS DEFINED BELOW) ARE BECOMING A PARTY TO THIS AGREEMENT AND YOU ARE CONSENTING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install IntelliJ IDEA.
< Back
I Agree
Cancel

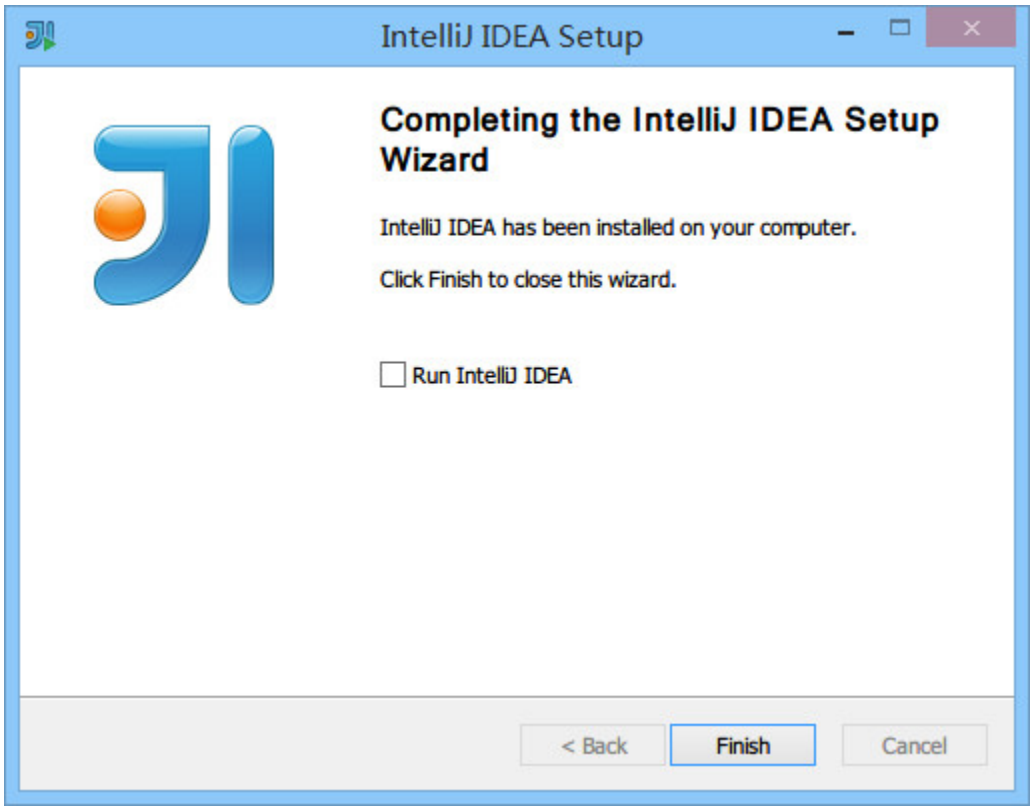

IntelliJ IDEA Setup
Choose Install Location
Choose the folder in which to install IntelliJ IDEA.
Setup will install IntelliJ IDEA in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Program Files (x86)\JetBrains\IntelliJ IDEA 14.1.4
Browse...
Space required: 831.5MB
Space available: 54.9GB
< Back
Next >
Cancel



- 上图标记 1 表示在桌面上创建一个快捷图标，建议勾选上，方便我们在安装后定位 IntelliJ IDEA 安装目录。
- 上图标记 2 表示关联 Java 和 Groovy 文件，建议都不要勾选，正常我们会在 Windows 的文件系统上打开这类文件都是为了快速查阅文件里面的内容，如果用 IntelliJ IDEA 关联上之后，由于 IntelliJ IDEA 打开速度缓慢，这并不能方便我们查看。
- 建议在 Windows 系统上关联此类文件可以用 EmEditor、Notepad++ 这类轻便的编辑器。


IntelliJ IDEA Setup
Choose Start Menu Folder
Choose a Start Menu folder for the IntelliJ IDEA shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
JetBrains
360安全中心
360密盘
37游戏中心
Accessibility
Accessories
Administrative Tools
ADSafe
Advanced SystemCare 6
Ashampoo
AVS4YOU
Chrome
COMODO
< Back
Install
Cancel

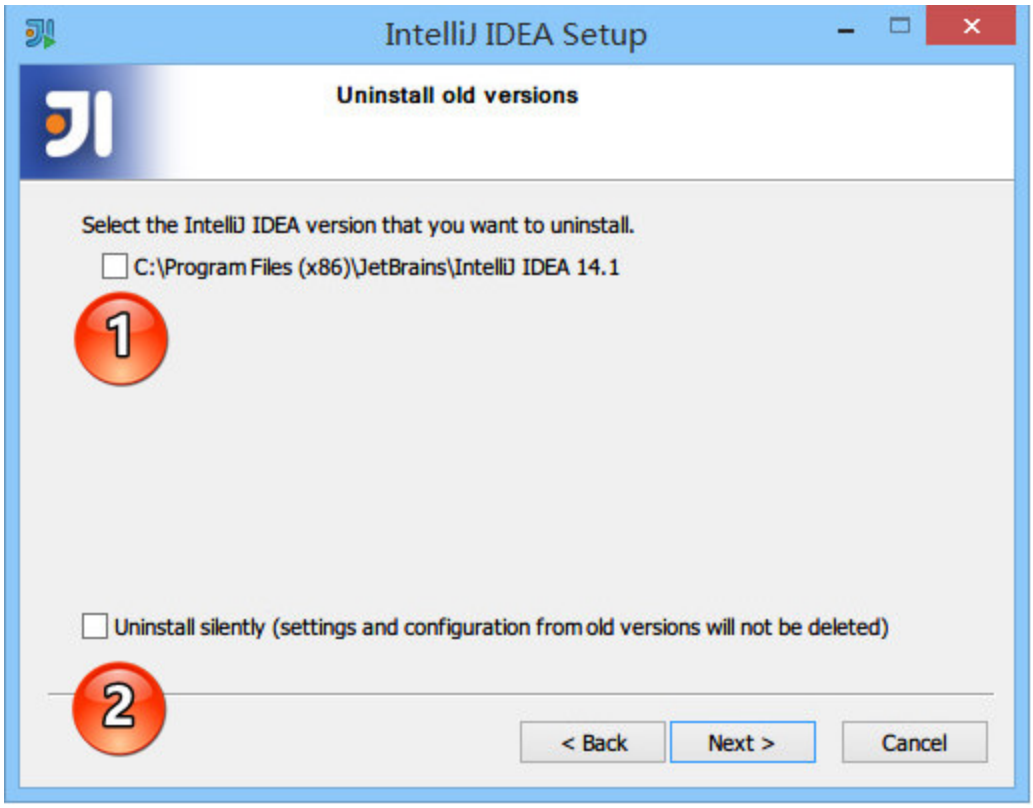

IntelliJ IDEA Setup
Installing
Please wait while IntelliJ IDEA is being installed.
Extract: common.jar
Show details
< Back
Next >
Cancel



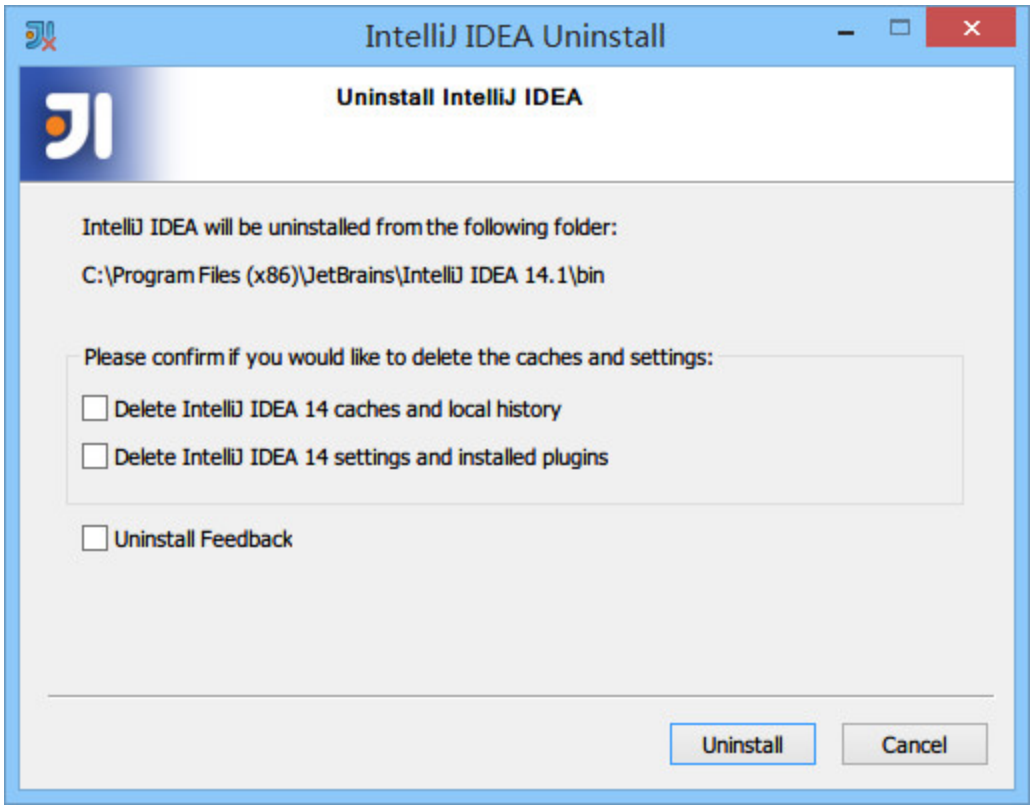
- 整个安装过程，一般的配置电脑安装所需的时间大约是 1 ~ 5 分钟。

已有旧版本安装新版本


IntelliJ IDEA Setup
Welcome to the IntelliJ IDEA Setup Wizard
This wizard will guide you through the installation of IntelliJ IDEA.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel



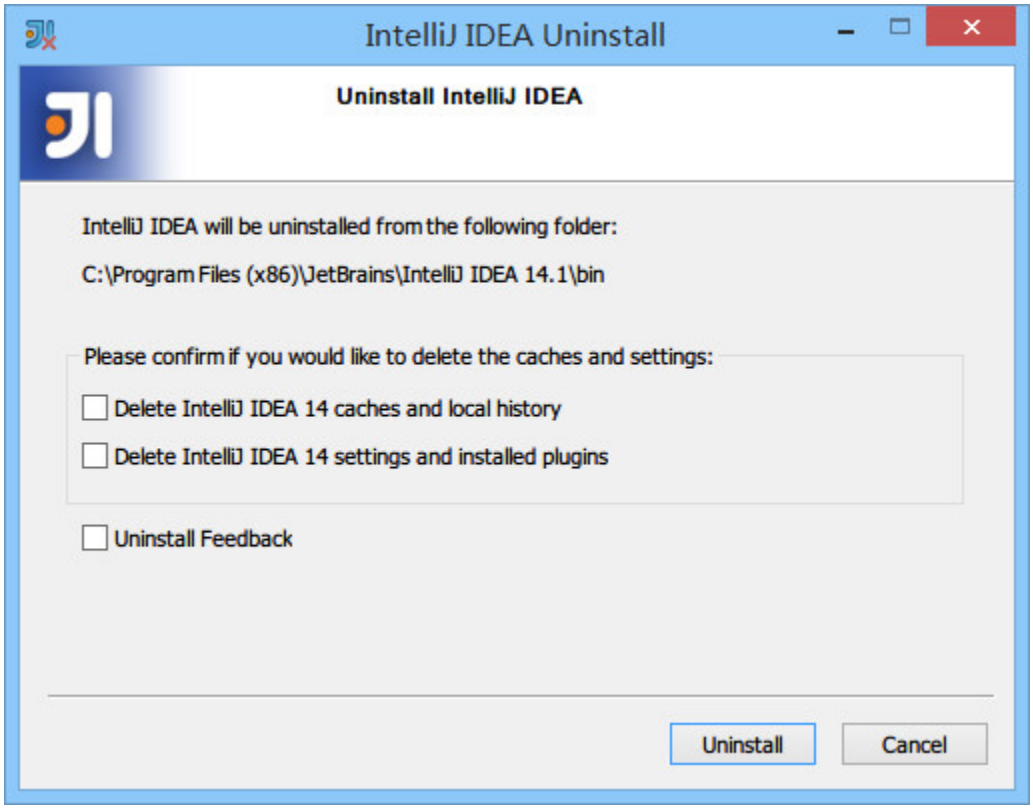
- 上图，显示我目前电脑中已经有一个 IntelliJ IDEA 版本，如果我勾选了标记 1，则表示安装之前会先卸载掉电脑上的旧版本。
- 上图标记 2，如果勾选了，则 IntelliJ IDEA 在卸载旧版本的时候直接删除掉你旧版本的个性化设置，这里不推荐勾选。
- 在小版本迭代中建议是卸载掉旧版本的，然后再进行新版本安装，因为小版本迭代一般都是 Bug 的修复，保留旧版本没有多大意义。
- 在大版本迭代中建议是保留旧版本，也就是不勾选上图标注 1，IntelliJ IDEA 是支持一台电脑装多个版本的。
- 接下来的步骤我们假设勾选了标注 1 再进行安装。

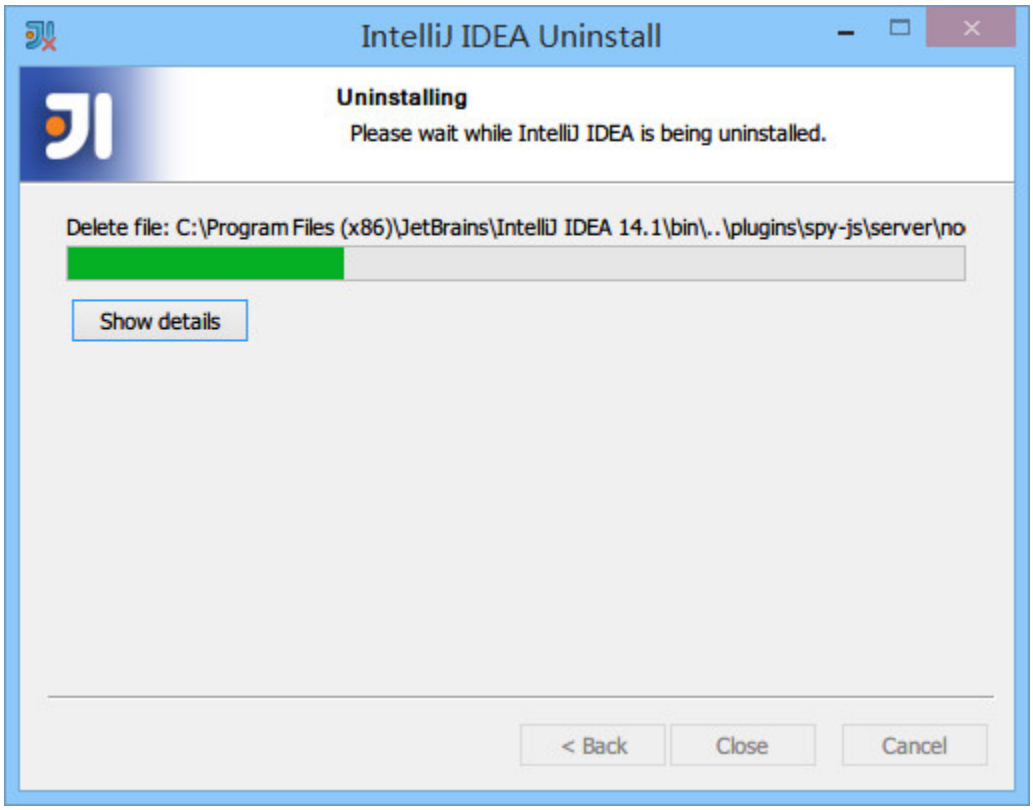


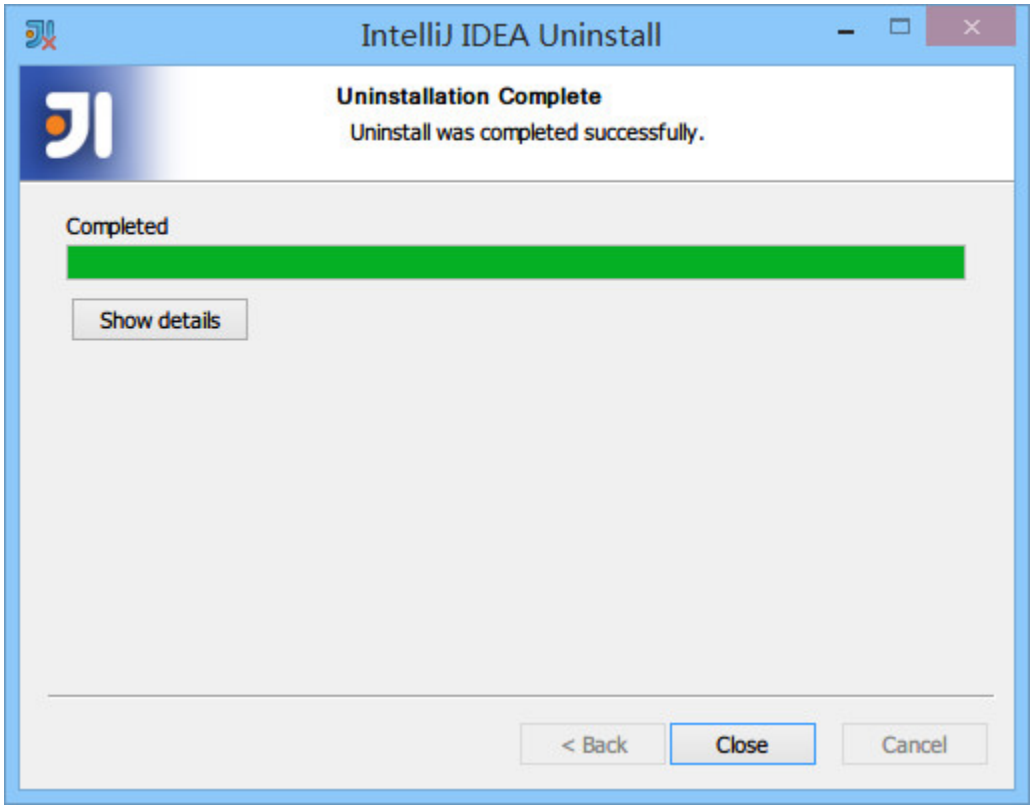
- 上图，由于上一步勾选了卸载旧版本选项，所以出现了选择删除旧版本的配置选项。
- 第一个选项：删除旧版本的缓存和本地历史记录。
- 第二个选项：删除旧版本的个人个性化设置。
- 建议两个都不要勾选。
- 点击 uninstall，进入全自动的卸载过程，卸载完成接下来的步骤跟上文“首次安装”一致，这里不再进行说明。

卸载

- 卸载过程在第 3 点已经有涉及到了，专门对 IntelliJ IDEA 进行卸载也是一样的流程。




IntelliJ IDEA Uninstall
Uninstalling
Please wait while IntelliJ IDEA is being uninstalled.
Delete file: C:\Program Files (x86)\JetBrains\IntelliJ IDEA 14.1\bin\..\plugins\spy-js\server\no
Show details
< Back
Close
Cancel


IntelliJ IDEA Uninstall
Uninstallation Complete
Uninstall was completed successfully.
Completed
Show details
< Back
Close
Cancel

Ubuntu 系统下安装 IntelliJ IDEA

系统要求

- 系统支持：只要是支持 GNOME 或 KDE 桌面系统，建议是 Ubuntu（32位和64位都可以）
- JDK 版本：Oracle JDK 1.6 或以上
- 内存：最低要求 1 GB，推荐 2 GB 以上
- 硬盘：最低要求 2 GB
- 显示器：最低要求 1024 X 768 分辨率
- 更多信息可以阅读：https://www.jetbrains.com/idea/download/system_requirements.jsp?os=linux

重要说明

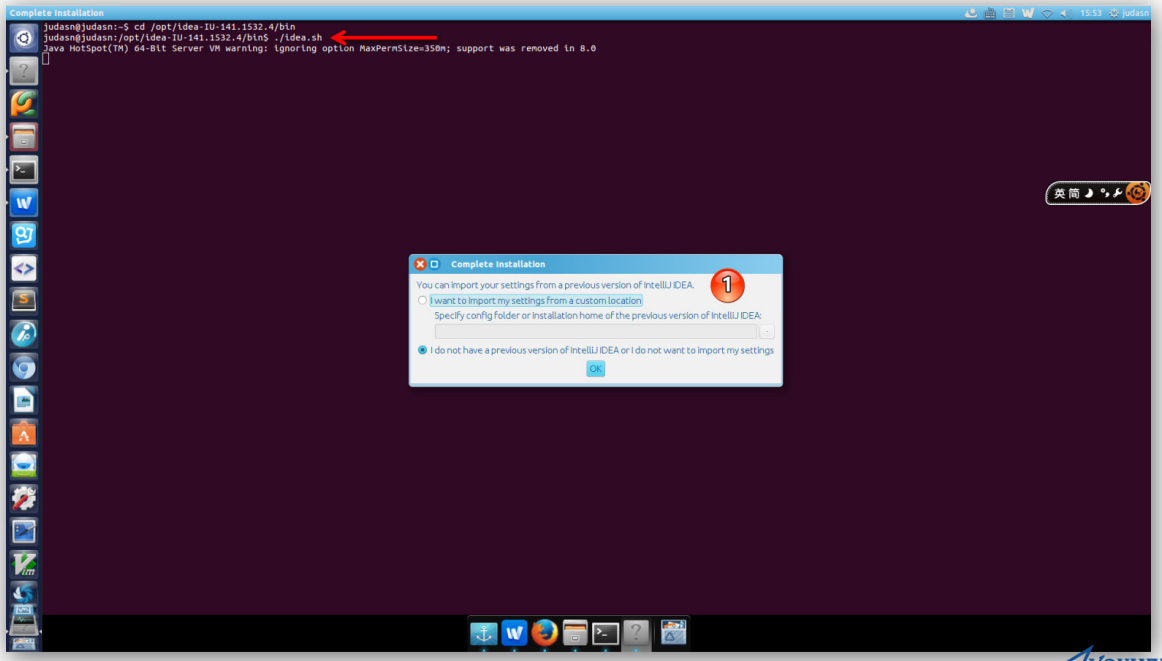
我这里以 Ubuntu 系统为例进行讲解。但是，在学习下面内容之前请先看下章节：Windows 下安装

因为它们配置流程是基本一样的，只是系统不同，开始的步骤不太一样而已，因此相同部分我这里是不会再讲的，我只讲 IntelliJ IDEA 在 Linux 安装特殊的地方。

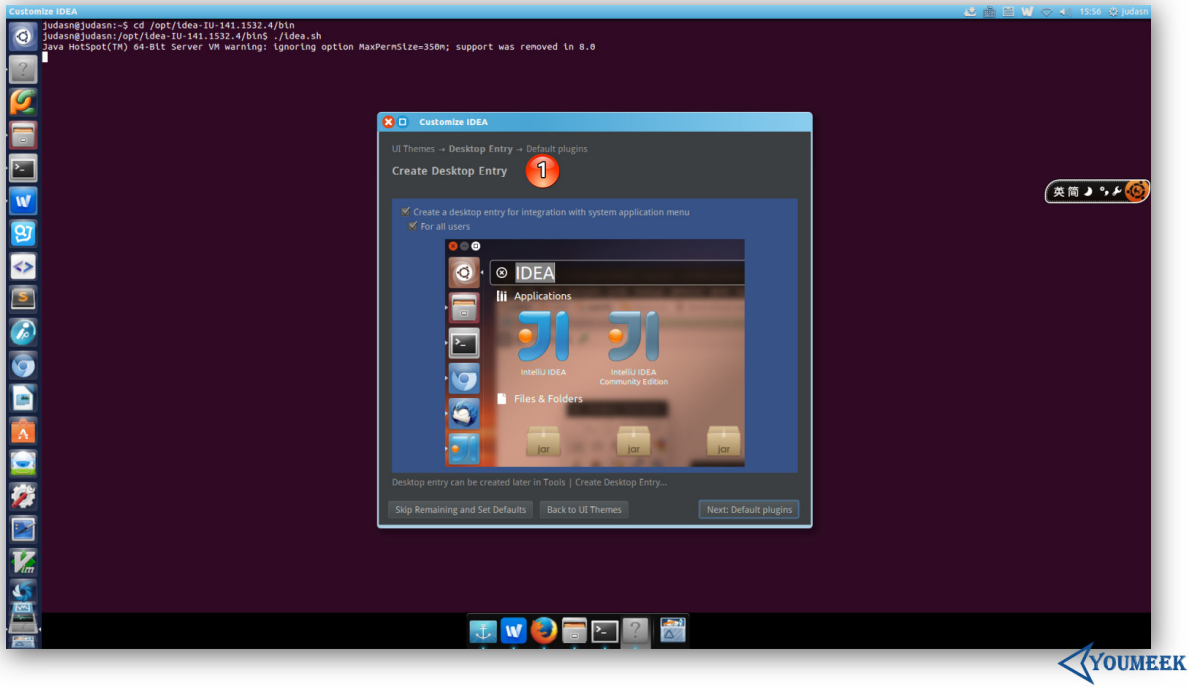
Ubuntu 下安装过程

先把你下载到的 `ideaIU-14.1.4.tar.gz` 移动到你平时存放软件的目录下，然后进行解压，我电脑是放在 /opt 下。

- 终端下解压命令：`tar xfz ideaIU-14.1.4.tar.gz`，解压出来的目录名称是：`idea-IU-141.1532.4`
- 可能在解压过程中你需要 `sudo` 命令权限，或者是切换到 root 账号下。如果你是切换到 root 用户下就一定要注意，解压完记得再切回来你常用的账户，不然等下生成的 IntelliJ IDEA 配置文件是放在 `/home/root` 下，这样就跟你常用的那个用户没啥关系了。



- 在假设你已经通过终端切换到了你常用的用户下之后，现在用终端进入解压目录下的 `bin` 子目录 下，然后在终端下运行启动命令：`./idea.sh`，运行的效果如上图箭头所示。剩下的配置步骤就跟 Windows 基本一样了，如标注 1 所示，所以这里不多讲。



- 其中，在整个首次启动的配置过程中，唯一跟 Windows 不太一样的就是上图标注 1 这个地方。原因是 Linux 下创建启动图标是非常非常非常的麻烦，所以 IntelliJ IDEA 帮我们考虑到了，所以只要勾选下即可解决这种麻烦事。

- 创建完启动图标之后，我们可以在如上图标注 1 所示的 Dash 这个地方找到 IntelliJ IDEA 图标。但是图标我们一般是放在启动栏上的，所以这里你可以按着箭头的方向拖动 IntelliJ IDEA 图标到启动栏上即可。

- 启动的最后效果如上图，是不是有点过于简单了？！

卸载

Linux 的卸载是不需要执行程序的，只需要：删除对应目录。

- 删除主程序目录，也就是我们本文上面讲的解压出来的 `idea-IU-141.1532.4`。
- 如果不想保留你的配置文件，还可以删除配置目录，目录所在位置：./home/你用登录名/.IntelliJIdea14

Mac 系统下安装 IntelliJ IDEA

系统要求

- 系统支持：Mac OS X 10.5 以上
- JDK 版本：Apple Java 6 或 Oracle Java 7 以上
- 内存：最低要求 1 G，推荐 2 G 以上
- 硬盘：最低要求 2 G
- 显示器：最低要求 1024 X 768 分辨率
- 更多信息可以阅读：https://www.jetbrains.com/idea/download/system_requirements.jsp?os=mac

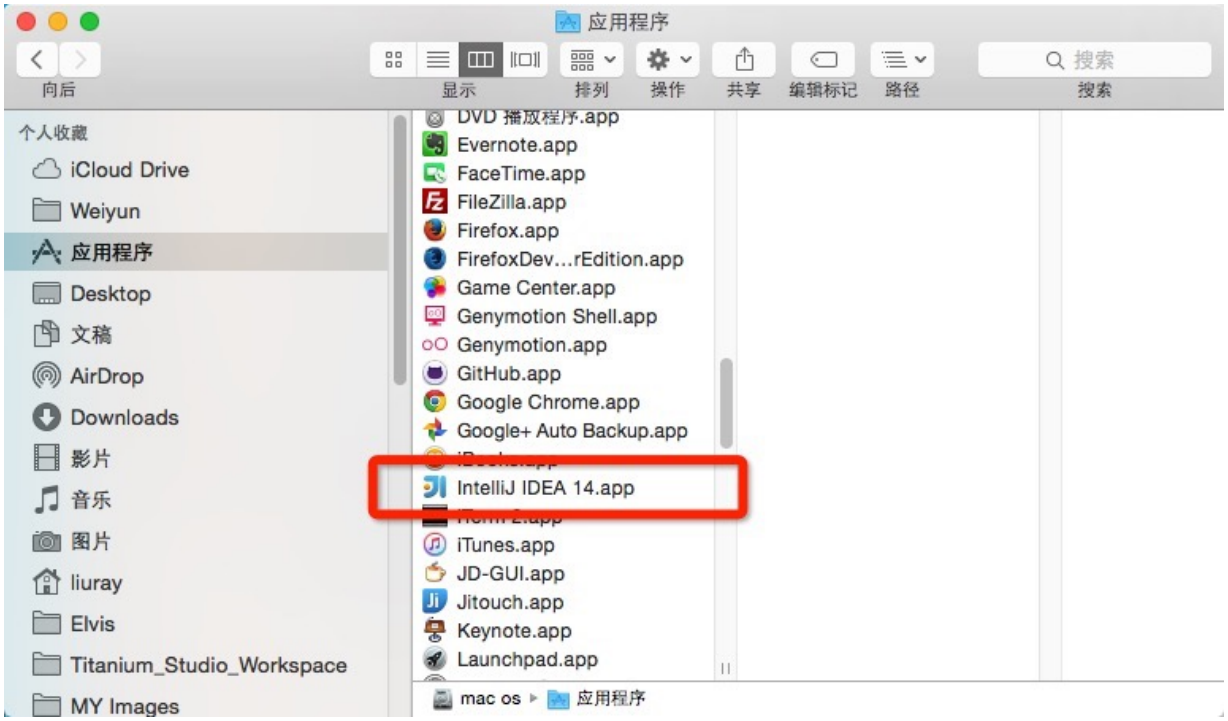
重要说明

我这里以 Mac 系统为例进行讲解。但是，在学习下面内容之前请先看下章节：Windows 下安装

因为它们配置流程是基本一样的，只是系统不同，开始的步骤不太一样而已，因此相同部分我这里是不会再讲的，我只讲 IntelliJ IDEA 在 Mac 安装特殊的地方。

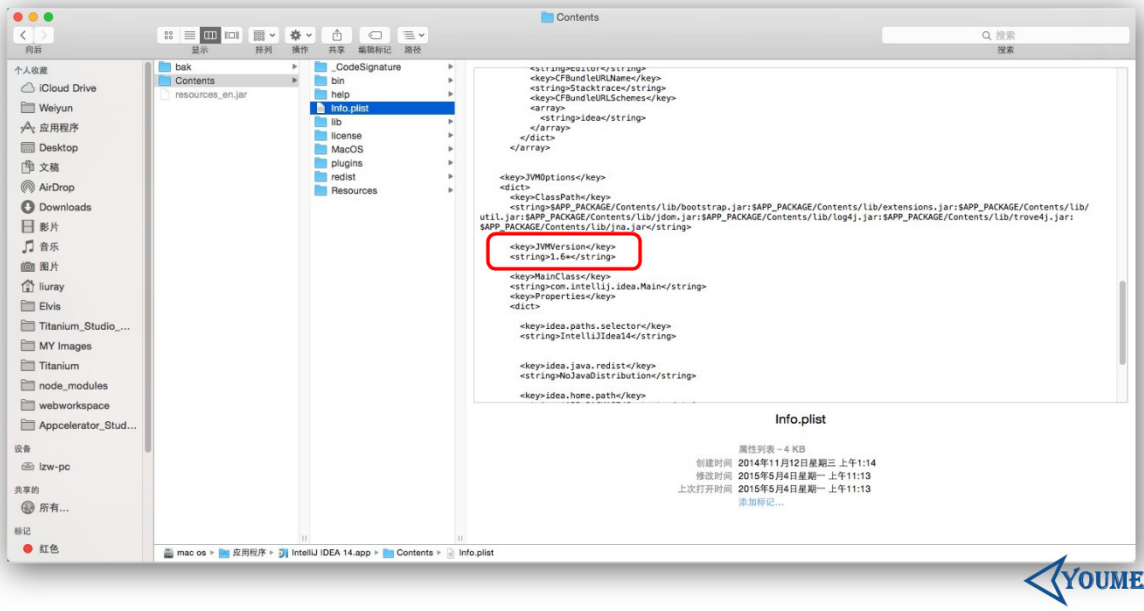
Mac 下安装过程

- 运行下载到 `ideaIU-14.1.4.dmg`，效果如上图所示。
- 根据提示把左侧的 IntelliJ IDEA 图标拖动到右侧目录图



- 拖动完成之后即可在 应用程序 中看到 IntelliJ IDEA 的启动图标，点击运行即可。

Mac 修改运行 JDK 版本



- 如果你的 Mac 安装有多个 JDK，你想使用高版本的 JDK 运行 IntelliJ IDEA 可以按如下方式进行修改：
- 在 应用程序 中找到 `IntelliJ IDEA.app` 然后对此进行 `右键 > 显示包内容 > Contents > Info.plist`，效果如上图所示。
- 找到上图红圈标注的代码，修改 `JVMVersion` 的属性值，如果是 JDK 7，则改为 `1.7*`。如果是 JDK 8，则改为 `1.8*`。

安装总结

硬件建议

从上一讲的安装教程来看，IntelliJ IDEA 对硬件的要求看上去不是很高。可是实际在开发中其实并不是这样的，特别是开发 Java Web 的项目的计算机，2G 内存是基本不够用的。

我们现在来假设一种国内常见的开发环境：

> 有一个在开发的 Java Web 项目，它使用的框架为主流的：Struts + Spring + Hibernate，使用者三个框架的过程中，我们要引入大量的框架架包，在我们的 Web 容器启动时，这些框架架包就要占用大量的内存，而且 IntelliJ IDEA 本身功能繁多，占用的内存也不算低，再加上我们这里还没计算计算机上的其他软件应用。所以基本上 2G 内存的计算机只适合写小程序、小项目或是开发静态页面。

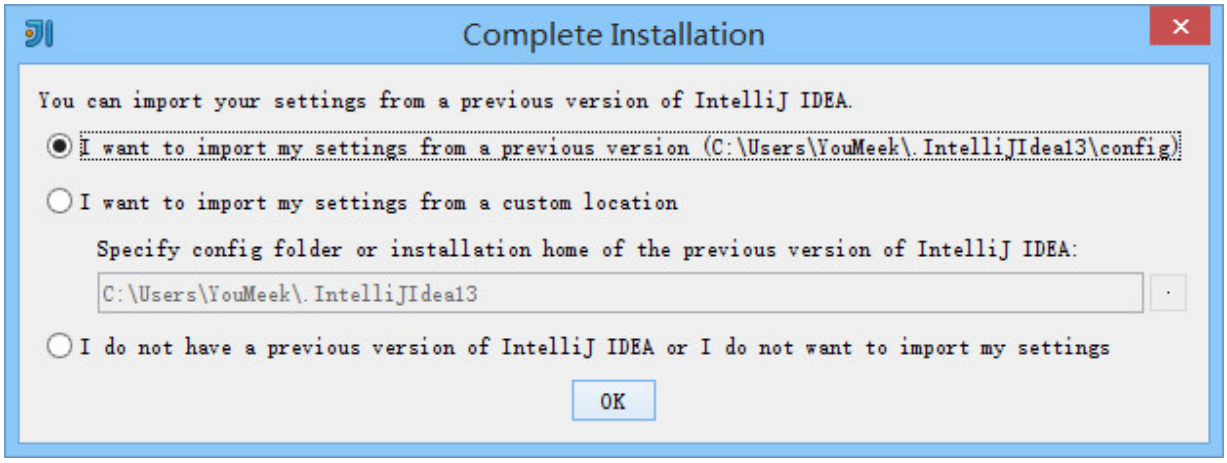
我个人建议，如果你是开发 Java Web 项目最好的方案是 8G 内存或是以上，硬盘能在用上固态是最好的，因为 IntelliJ IDEA 有大量的缓存、索引文件，把 IntelliJ IDEA 的缓存、索引文件放在固态上，IntelliJ IDEA 流畅度也会加快很多。

如果你正在使用 Eclipse / MyEclipse，想通过 IntelliJ IDEA 来解决计算机的卡、慢等问题，我这里可以直接明白地告诉你：这基本上是不可能的，本质上你应该对自己的硬件设备进行升级。

首次运行

向导功能

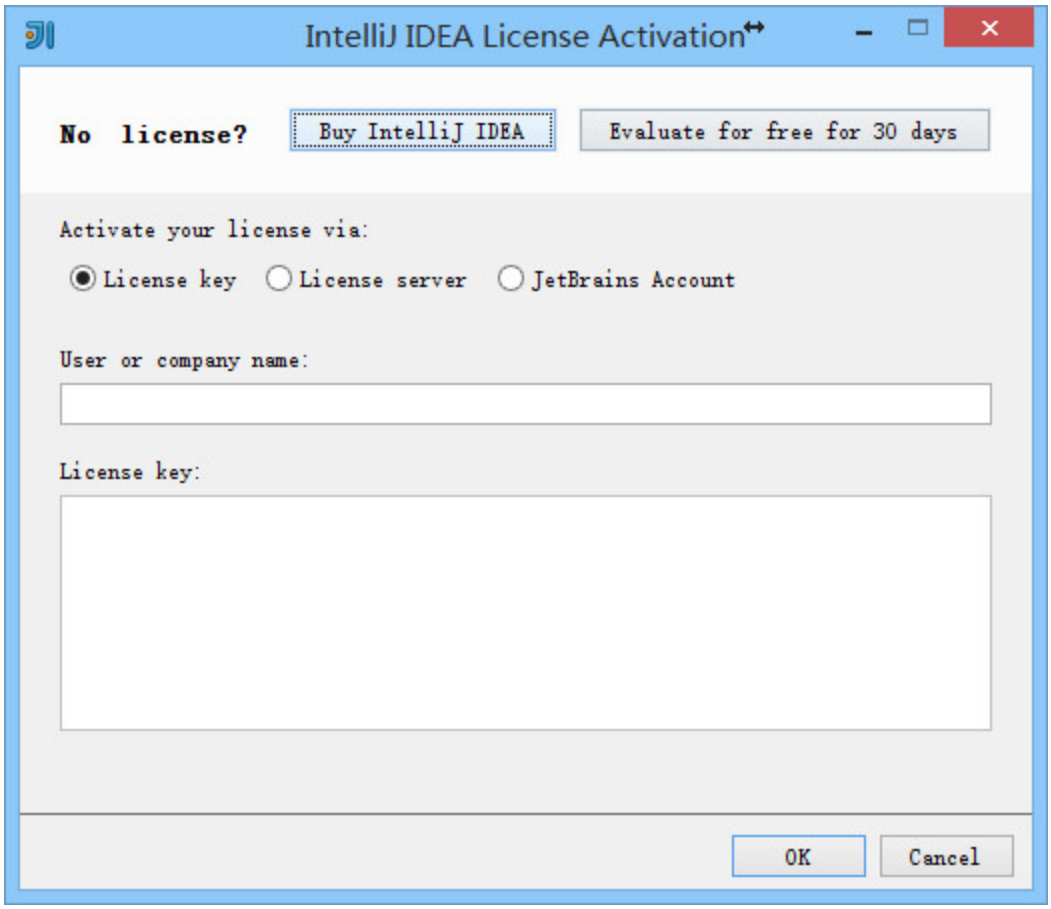
- 假如你计算机上在过去已经有安装过 IntelliJ IDEA 14 的版本，且你在卸载 IntelliJ IDEA 的过程中，IntelliJ IDEA 的配置文件目录都没有删除，那安装新版本之后是不会有首次运行的功能向导的。
- 假如你计算机上没有安装过 IntelliJ IDEA，或是 卸载 IntelliJ IDEA 过程中你删除了 IntelliJ IDEA 的配置文件目录，则当你双击运行桌面上的 IntelliJ IDEA 快捷图标，将进入下面介绍的向导过程。



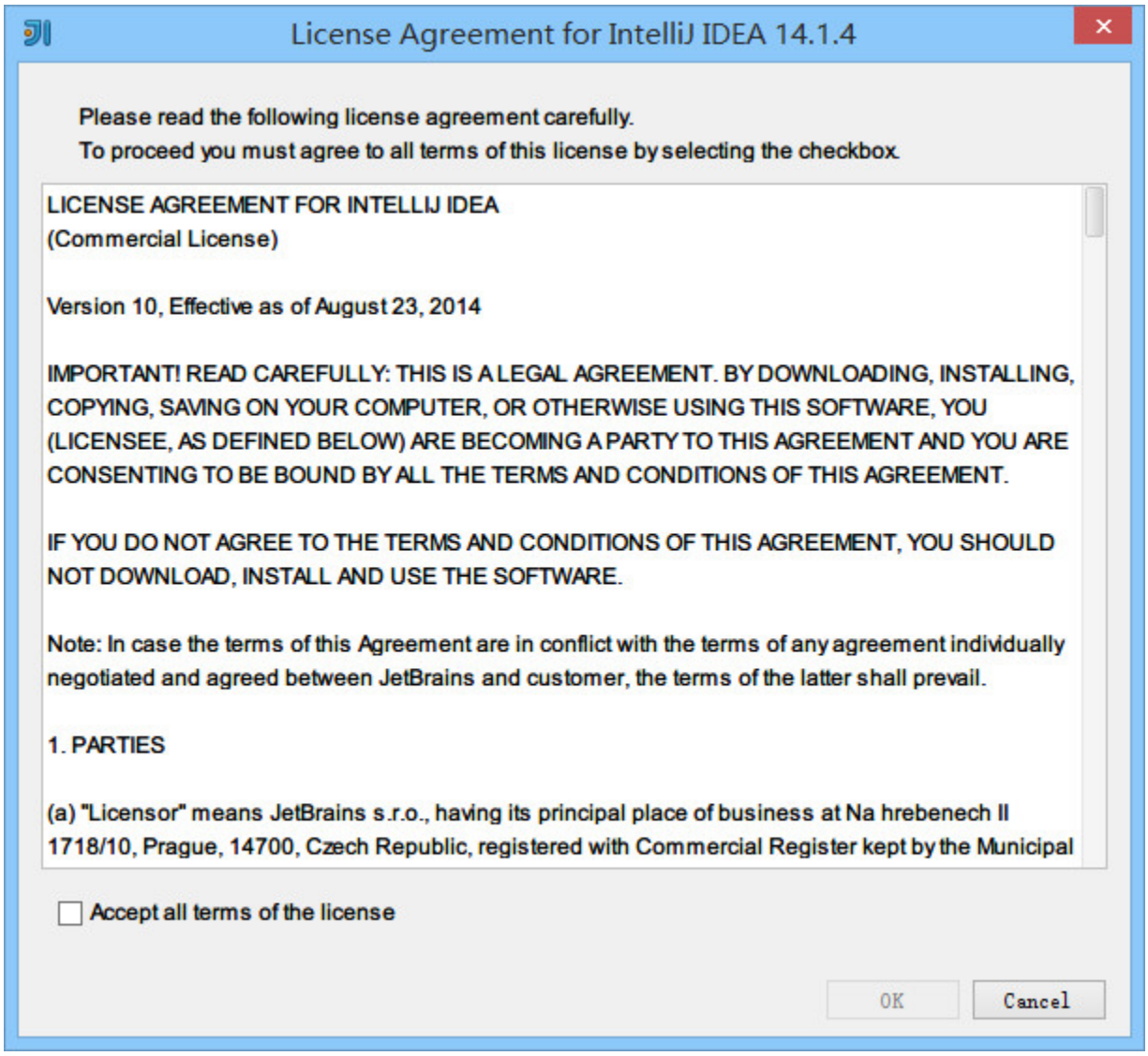
- 上图第一个单选按钮表示 IntelliJ IDEA 识别到我计算机上有 IntelliJ IDEA 13 版本的旧配置，如果我选择了该选项，则 IntelliJ IDEA 将自动把旧版本的配置文件转移到新版本的配置文件目录上。如果你计算机上首次安装一般是没有该选项的。
- 上图第二个单选按钮表示你可以指定 IntelliJ IDEA 导入你计算机上存在其他目录的 IntelliJ IDEA 配置文件目录，如果你

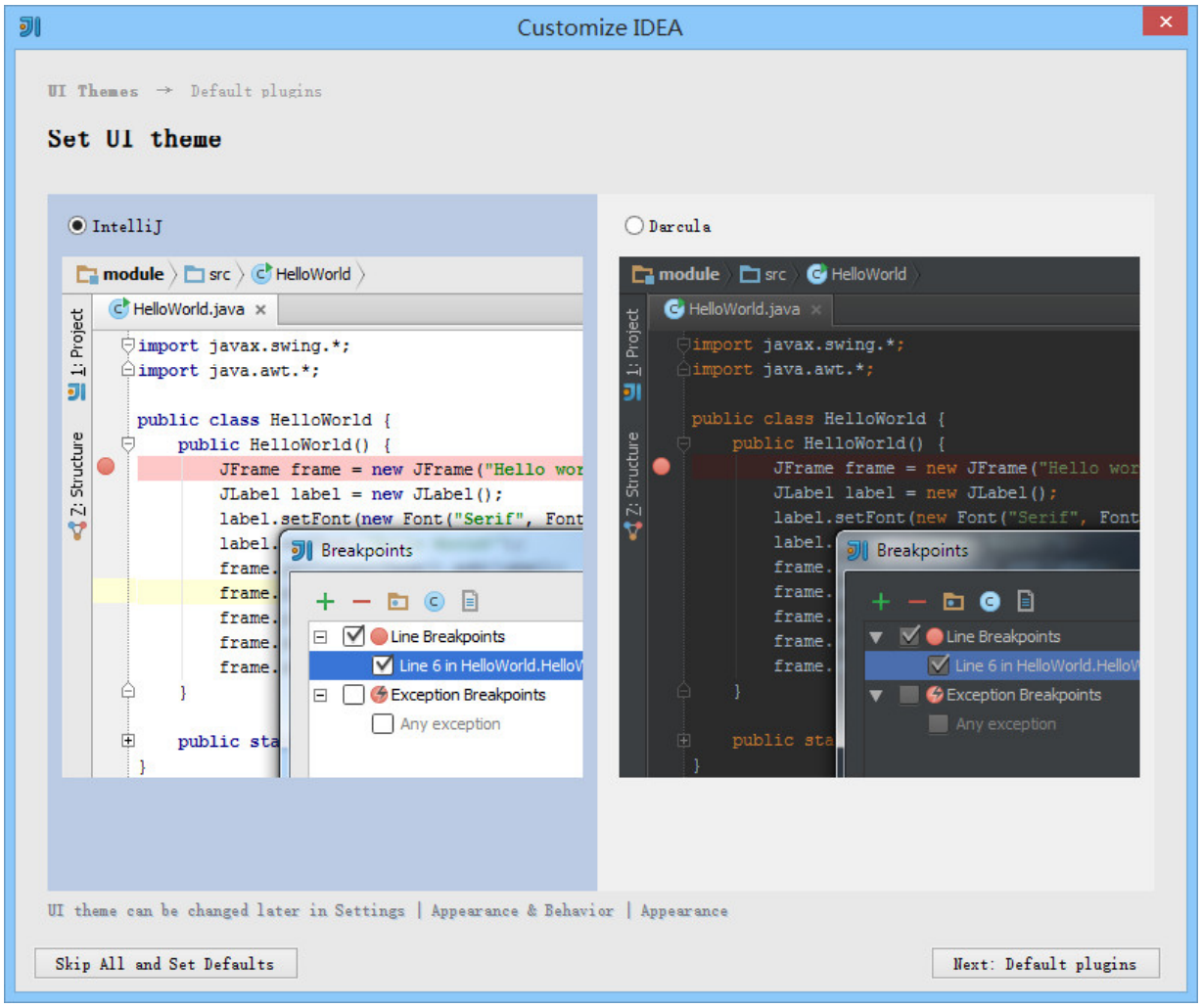
有的话。

- 上图第三个单选按钮表示你没有任何早期版本的 IntelliJ IDEA 配置，你不导入任何配置，让 IntelliJ IDEA 生成一份新的配置。



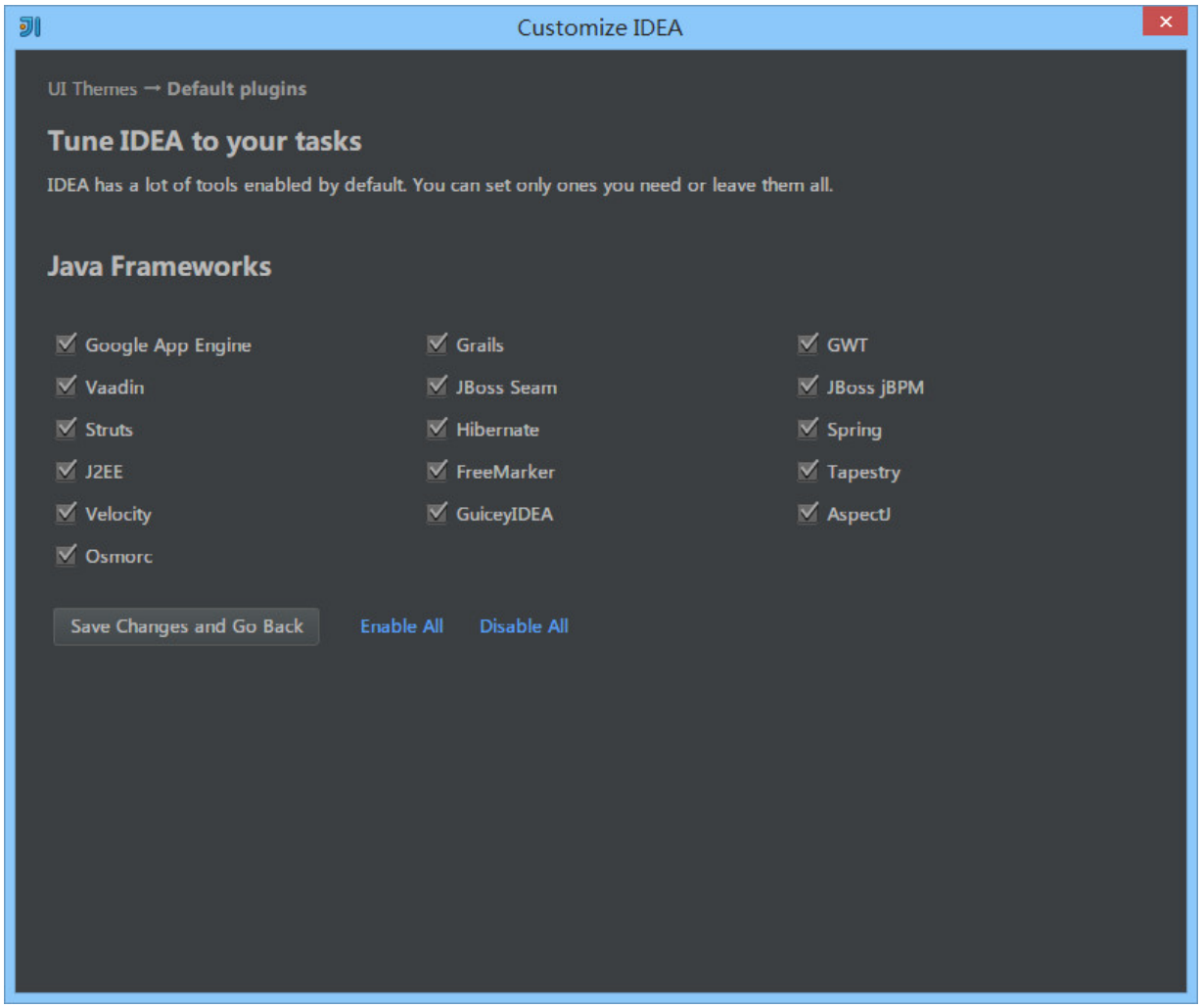
- 上图默认选择中的是 `Buy IntelliJ IDEA`，验证 IntelliJ IDEA 的许可有如图三种方式，我们这里使用的是 30 天试用版本进行演示，顾单击 `Evaluate for free for 30 days` 进行下一步。


License Agreement for IntelliJ IDEA 14.1.4
Please read the following license agreement carefully.
To proceed you must agree to all terms of this license by selecting the checkbox.
LICENSE AGREEMENT FOR INTELLIJ IDEA
(Commercial License)
Version 10, Effective as of August 23, 2014
IMPORTANT! READ CAREFULLY: THIS IS A LEGAL AGREEMENT. BY DOWNLOADING, INSTALLING, COPYING, SAVING ON YOUR COMPUTER, OR OTHERWISE USING THIS SOFTWARE, YOU (LICENSEE, AS DEFINED BELOW) ARE BECOMING A PARTY TO THIS AGREEMENT AND YOU ARE CONSENTING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.
IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU SHOULD NOT DOWNLOAD, INSTALL AND USE THE SOFTWARE.
Note: In case the terms of this Agreement are in conflict with the terms of any agreement individually negotiated and agreed between JetBrains and customer, the terms of the latter shall prevail.
1. PARTIES
(a) "Licensor" means JetBrains s.r.o., having its principal place of business at Na hrebenech II 1718/10, Prague, 14700, Czech Republic, registered with Commercial Register kept by the Municipal
Accept all terms of the license
OK
Cancel



- 上图选择的时候 IntelliJ IDEA 主题 UI，在 Windows 系统版本中 IntelliJ IDEA 自带了 4 个主题，但是用的最多的就是上图这 2 种，其中大家基本偏爱黑色的 `Darcula`。这个没有好坏之分，根据你的喜好来进行选择，我们演示的版本就是用 `Darcula`。

- 上图显示了 IntelliJ IDEA 支持的主要的一些扩展功能或者说是工具、插件也可以。你可以根据自己开发的需求进行禁用一些扩展，这样可以稍微减轻 IntelliJ IDEA 运行时所占内存，加快运行速度，但是效果并不会很明显就是。
- 我们这里点击 `Java Frameworks` 的 `Customize` 进行下一步操作。

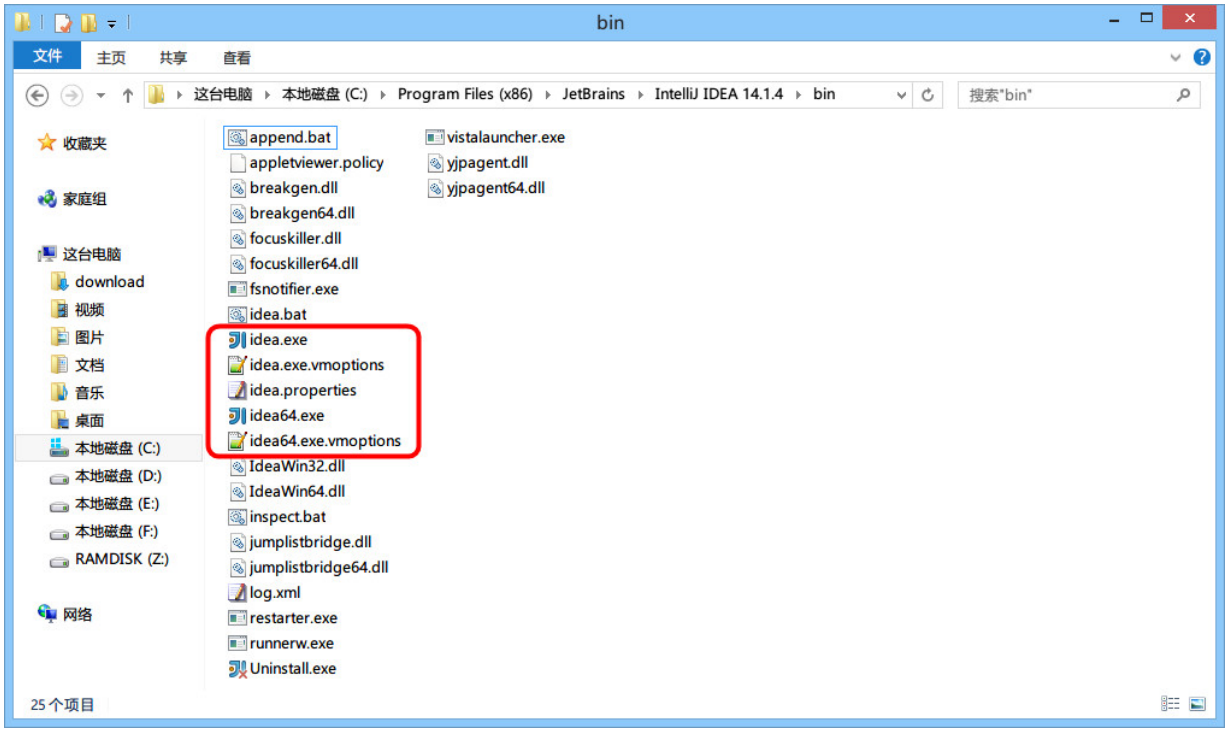


- 上图显示了 IntelliJ IDEA 所以支持的 `Java Frameworks`。我们可以根据自己的开发需求不启用指定框架的。去掉框架前面的勾选框就表示不启用该框架功能支持。
- 对于不启用的框架，我们也可以在后期进行重新勾选，这会在 IntelliJ IDEA 插件那一讲进行专门讲解。

- 选择好自己所需的扩展功能后，按 `Start using IntelliJ IDEA` 显示上图启动界面，金黄色进度条走完之后，欢迎真正进入 IntelliJ IDEA 的编码世界！

IntelliJ IDEA 相关核心文件和目录介绍

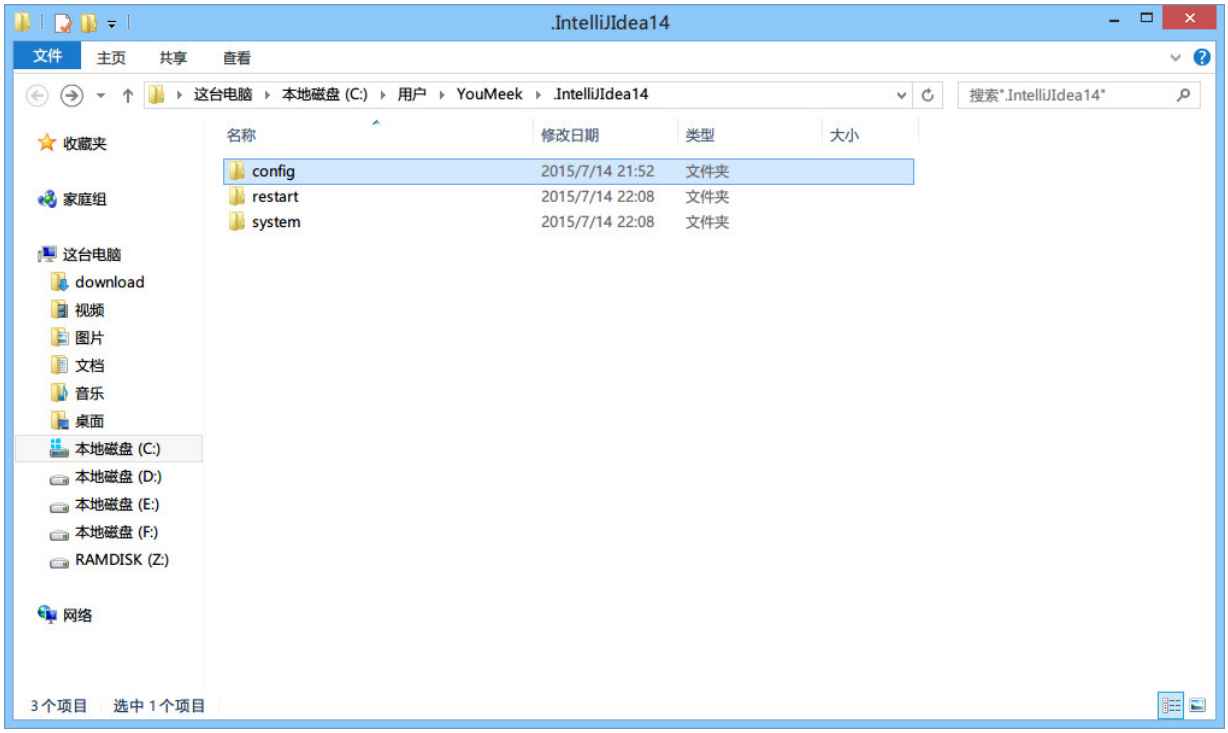
安装目录介绍



- IntelliJ IDEA 的安装目录并不复杂，上图为最常改动的 bin 目录，经常会改动的文件或是必须介绍就是如图红色框中的几个。
- `idea.exe` 文件是 IntelliJ IDEA 32 位的可行执行文件，IntelliJ IDEA 安装完默认发送到桌面的也就是这个执行文件的快捷方式。
- `idea.exe.vmoptions` 文件是 IntelliJ IDEA 32 位的可执行文件的 VM 配置文件，具体配置修改会下面进行专门讲解。

- `idea64.exe` 文件是 IntelliJ IDEA 64 位的可行执行文件，要求必须电脑上装有 JDK 64 位版本。64 位的系统也是建议使用该文件。
- `idea64.exe.vmoptions` 文件是 IntelliJ IDEA 64 位的可执行文件的 VM 配置文件，具体配置修改会下面进行专门讲解。
- `idea.properties` 文件是 IntelliJ IDEA 的一些属性配置文件，具体配置修改会下面进行专门讲解。

设置目录介绍

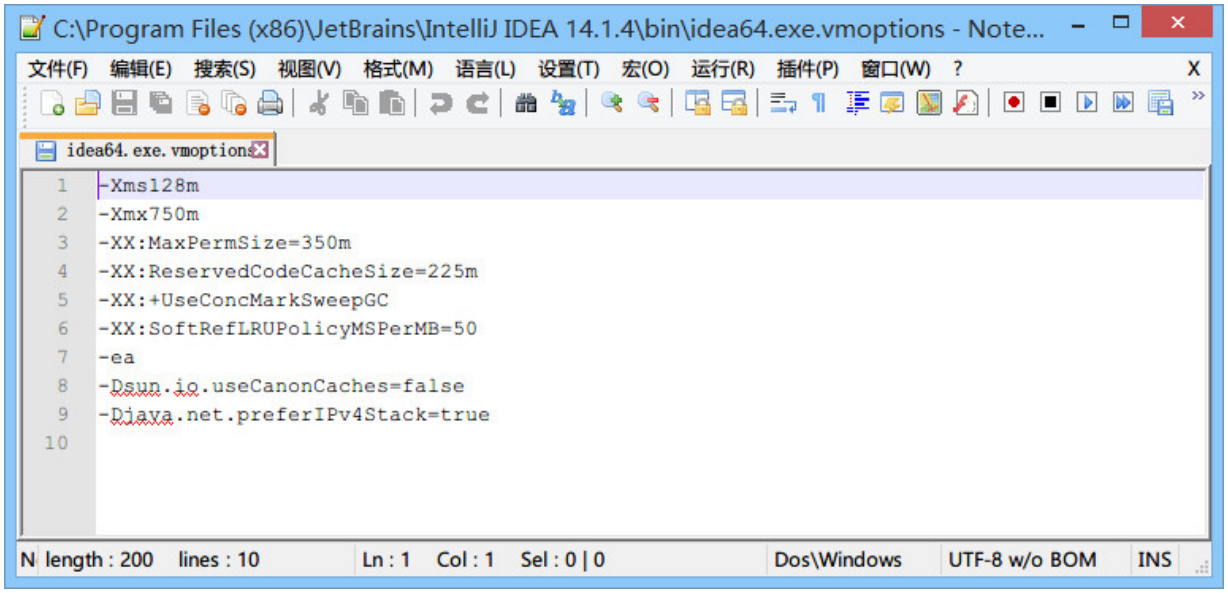


- 不管你使用的是哪个操作系统，IntelliJ IDEA 的设置目录命名是统一的、有规律：.IntelliJ IDEA14。其中 14 表示大版本号，如果你电脑上还同时装有 13 的版本，那则还应该会有一个：`.IntelliJ IDEA13` 的设置目录，其他版本道理一样。
- 在三大主流的操作系统上，你只要对整个硬盘进行搜索：`.IntelliJ IDEA14`，即可找到，无需可以去记忆到底生成在哪个目录下。
- 对于这个设置目录有一个特性，就是你删除掉整个目录之后，重新启动 IntelliJ IDEA 会再自动帮你再生成一个全新的默认配置，所以很多时候如果你把 IntelliJ IDEA 配置改坏了，没关系，删掉该目录，一切都会还原到默认，我是很建

议新人可以多自己摸索 IntelliJ IDEA 的配置，多几次还原，有助于加深对 IntelliJ IDEA 的了解。

- `config` 目录是 IntelliJ IDEA 个性化化配置目录，或者说是整个 IDE 设置目录。也是我个人认为最重要的目录，没有之一，如果你还记得安装篇的介绍的时候，安装新版本的 IntelliJ IDEA 会自动扫描硬盘上的旧配置目录，指的就是该目录。这个目录主要记录了：IDE 主要配置功能、自定义的代码模板、自定义的文件模板、自定义的快捷键、Project 的 tasks 记录等等个性化的设置。
- `system` 目录是 IntelliJ IDEA 系统文件目录，是 IntelliJ IDEA 与开发项目一个桥梁目录，里面主要有：缓存、索引、容器文件输出等等，虽然不是最重要目录，但是也是最不可或缺目录之一。

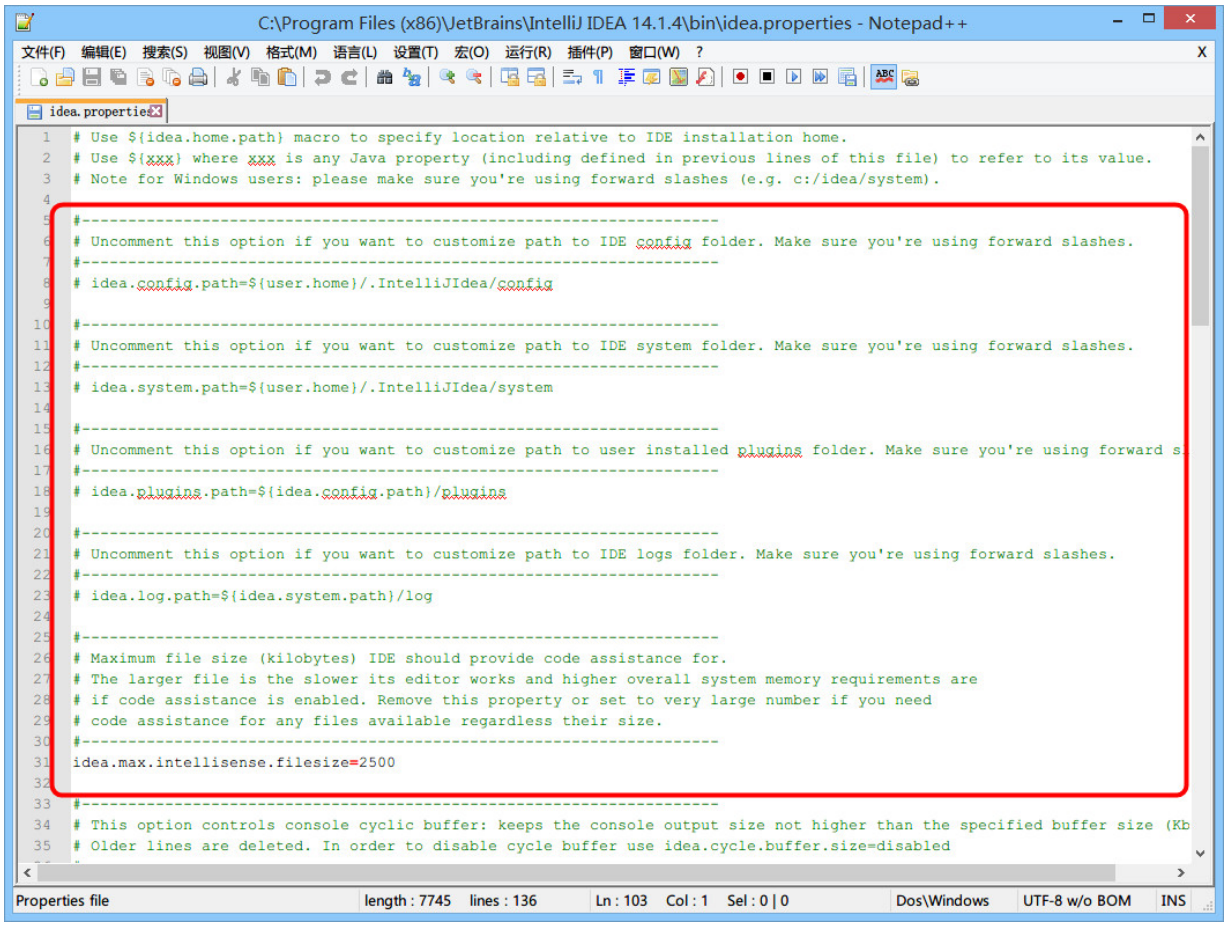
配置文件常见修改内容说明



- 上图是 64 位可执行文件的 JVM 配置文件内容，如果你是 32 位的系统你应该修改的是 `idea.exe.vmoptions` 文件里面的内容，但是由于 32 位系统内存一般都是 2G 左右的，所以也没有多大空间可以调整，所以一般无需调整的。
- 修改的原则主要是根据自己机器的内存情况来判断的，我个人是建议 8G 以下的机子或是静态页面开发者都是无需修改的。如果你是开发大型项目、Java 项目或是 Android 项目，并且内存大于 8G，建议进行修改，常修改的就是下面 4 个参数，我这里主要以我的机子会例进行建议，每个人机子情况不一，这里也只是做一个引子，最好的调整方式是你可以根据 jconsole 这类工具进行观察后个性化调整。

  - `-Xms128m`，16 G 内存的机器可尝试设置为 `-Xms512m`
  - `-Xmx750m`，16 G 内存的机器可尝试设置为 `-Xmx1500m`
  - `-XX:MaxPermSize=350m`，16G 内存的机器可尝试设置为 `-XX:MaxPermSize=500m`

- `-XX:ReservedCodeCacheSize=225m`，16G 内存的机器可尝试设置为 `-XX:ReservedCodeCacheSize=500m`



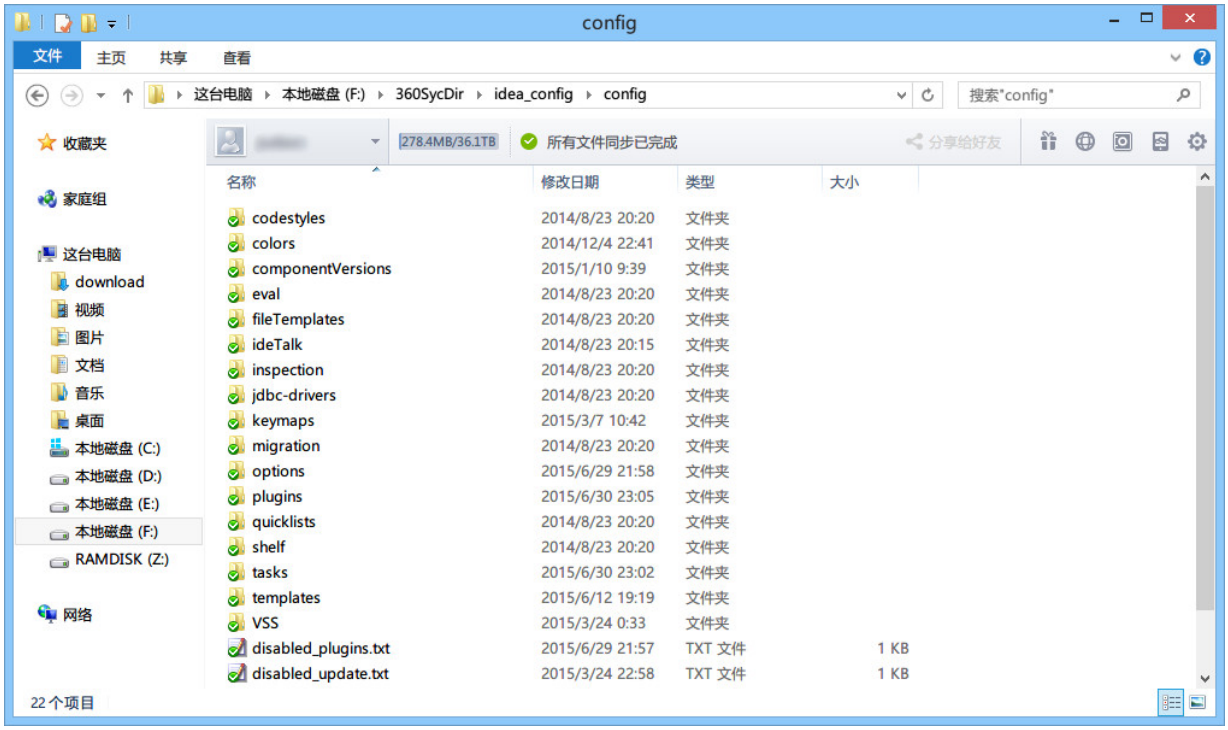
- 上图是 IntelliJ IDEA 一些属性配置，没有 32 位和 64 位之分，修改原则主要根据个人对 IntelliJ IDEA 的个性化配置情况来分析。常修改的就是下面 4 个参数：

  - `idea.config.path=${user.home}/.IntelliJIdea/config`，该属性主要用于指向 IntelliJ IDEA 的个性化配置目录，默认是被注释，打开注释之后才算启用该属性，这里需要特别注意的是斜杠方向，这里用的是正斜杠。
  - `idea.system.path=${user.home}/.IntelliJIdea/system`，该属性主要用于指向 IntelliJ IDEA 的系统文件目录，

默认是被注释，打开注释之后才算启用该属性，这里需要特别注意的是斜杠方向，这里用的是正斜杠。如果你的项目很多，则该目录会很大，如果你的 C 盘空间不够的时候，还是建议把该目录转移到其他盘符下。

- `idea.max.intellisense.filesize=2500`，该属性主要用于提高在编辑大文件时候的代码帮助。IntelliJ IDEA 在编辑大文件的时候还是很容易卡顿的。
- `idea.cycle.buffer.size=1024`，该属性主要用于控制控制台输出缓存。有遇到一些项目开启很多输出，控制台很快就被刷满了没办法再自动输出后面内容，这种项目建议增大该值或是直接禁用掉，禁用语句 `idea.cycle.buffer.size=disabled`。

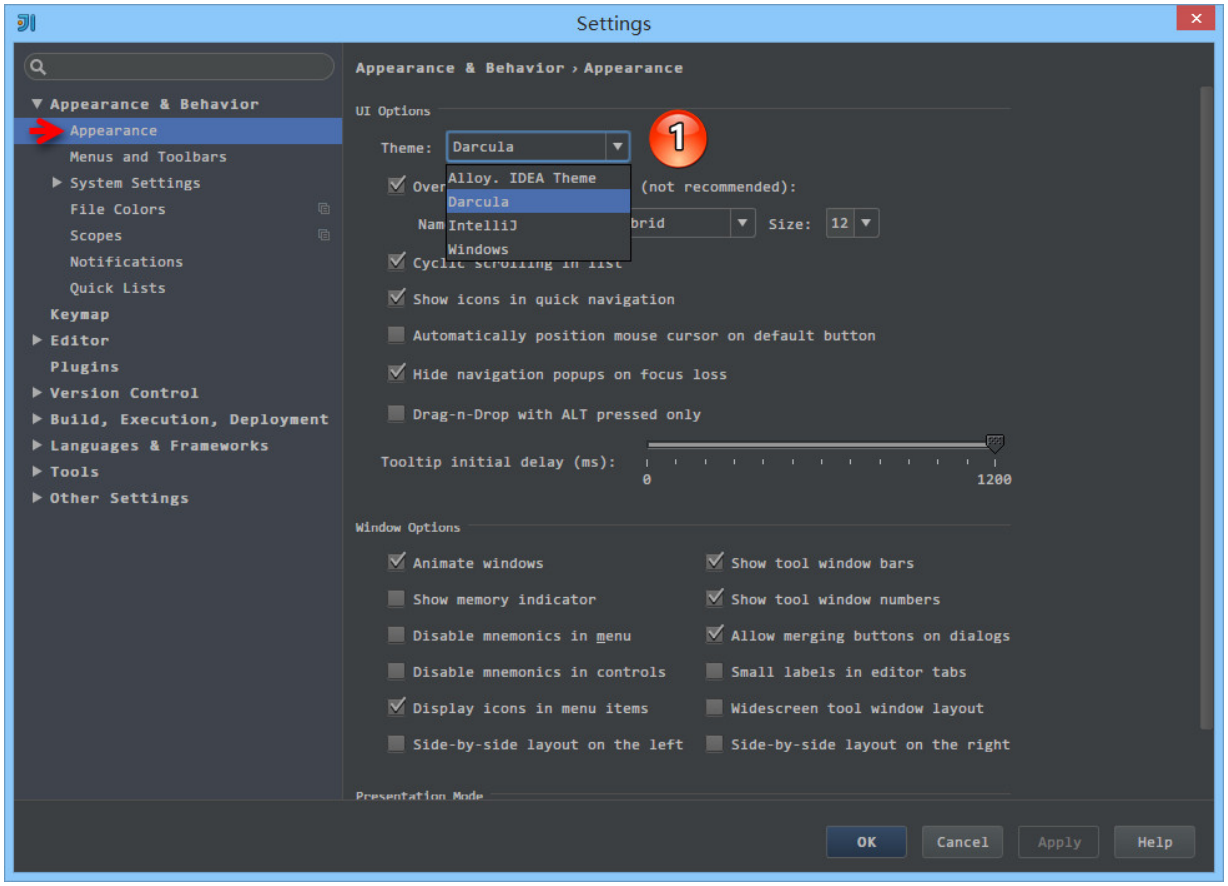
设置目录进行多台设置同步化处理



- 上图是我的个性化配置目录，我是存放在 F 盘，同时该目录也是在 360 同步盘中。这样做主要是为了让我的多台设置可以同时使用一个个性化配置，保证个人开发习惯，额外作用就是在服务器上一个备份作用。
- 设置方式很简单，修改 `idea.properties` 属性文件中的 `idea.config.path` 值，我的机器为：`idea.config.path=F:/360SycDir/idea_config/config`

IntelliJ IDEA 主题、字体、编辑区主题、文件编码修改

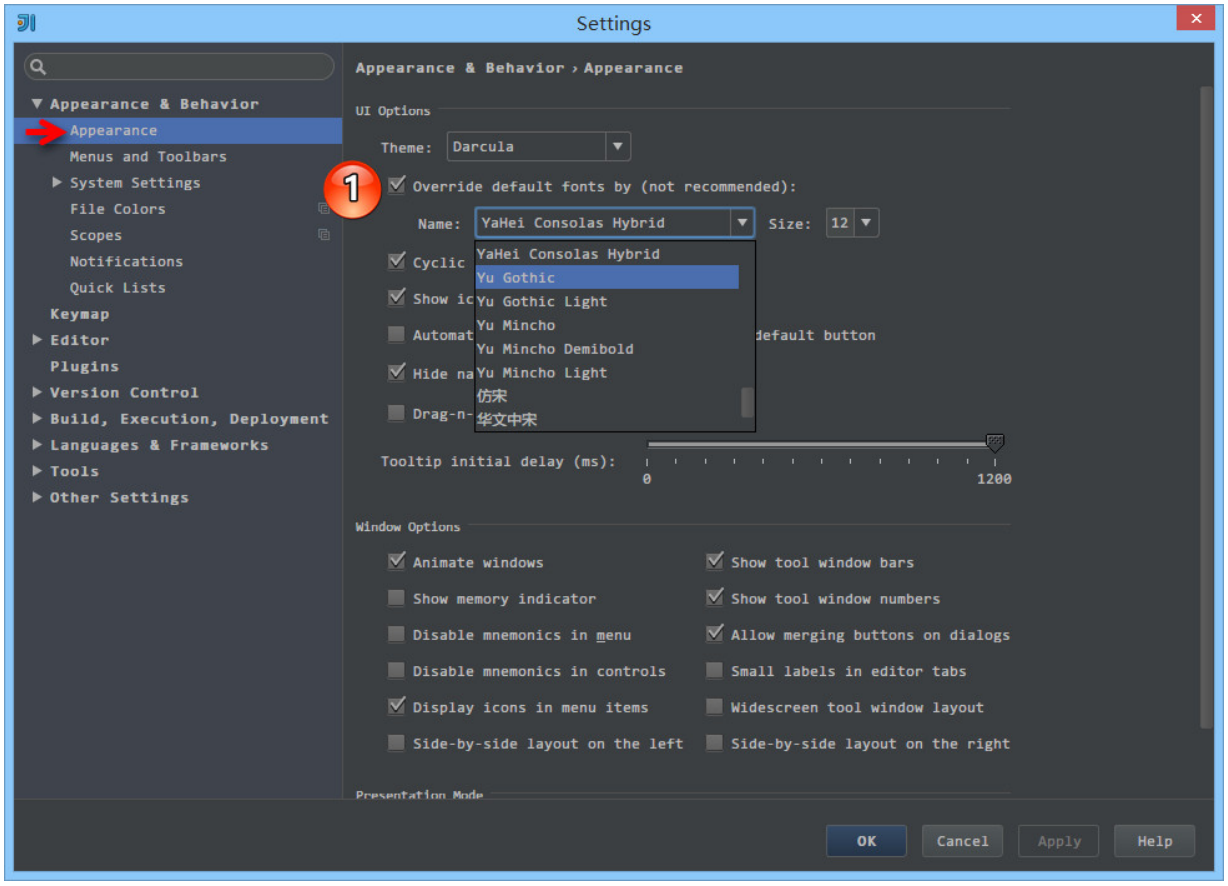
主题修改



- 上图标注 1 所示为 IntelliJ IDEA 修改主题的地方，在 Windows 系统上 IntelliJ IDEA 默认提供的主题有四套：`Darcula`、`IntelliJ`、`Windows`、`Alloy. IDEA Theme`。除了 `Darcula` 是黑色主题，其他三套都是以白色为背景的。
- 其他操作系统上不一定会也有四套主题的，主题的选择上大家根据自己喜好即可。改变主题需要重启 IntelliJ IDEA 方可看到效果。

字体修改

主题字体修改

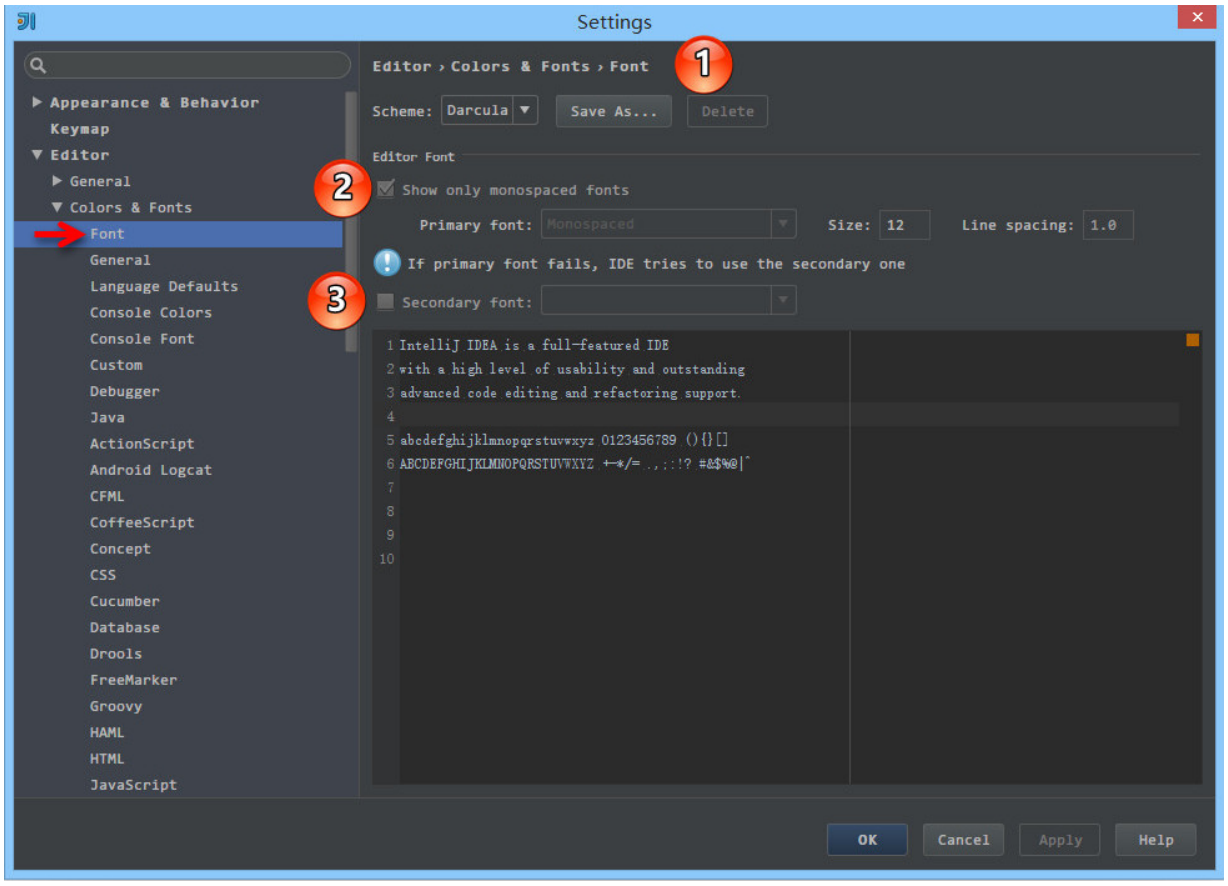


- 如上图标注 1 所示，IntelliJ IDEA 主题字体的修改要先勾选 `Override default fonts by` 。默认 IntelliJ IDEA 是不推荐修改的，但是由于字体是有分包含中文和不包含中文之分的，一般使用英文的国家是不需要额外担心乱码问题的，而我们需要。
- 字体的审美上每个人不一样，但是如上一段说的，这里的字体修改是需要知道一个前提的，那就是你选择的那个字体必

须含有中文，比如微软雅黑和宋体这类是包含中文的，而 `Courier New` 和 `Monaco` 这类只是单纯的英文字体。

- 如果你选择的字体不包含中文，那可能会在很多位置上出现类似 □□□□□ 这样的乱码问题，比如文件名含有中文、字体是中文名字的都会变成 □□□□□。
- 在修改 IntelliJ IDEA 的主题字体的时候，不建议把字体调成很大，因为很多人遇到这样一种情况：显示器分辨率低，主题字体又大，在 IntelliJ IDEA 的某些操作的工具菜单、右键菜单选项中部分选项超出了分辨率显示范围，没办法被选中。当然了，如果你一定要把字体改大，又不用大分辨率显示器，那可以通过 IntelliJ IDEA 的 `Menus and Toolbars` 删除部分你认为用不到的菜单，但是一般不建议这样做。
- 还需要特别注意的时候，如果你是开着 IntelliJ IDEA 的时候，新装了一个字体的话，那必须重启IntelliJ IDEA 之后才能在下拉列表找到新装的字体。

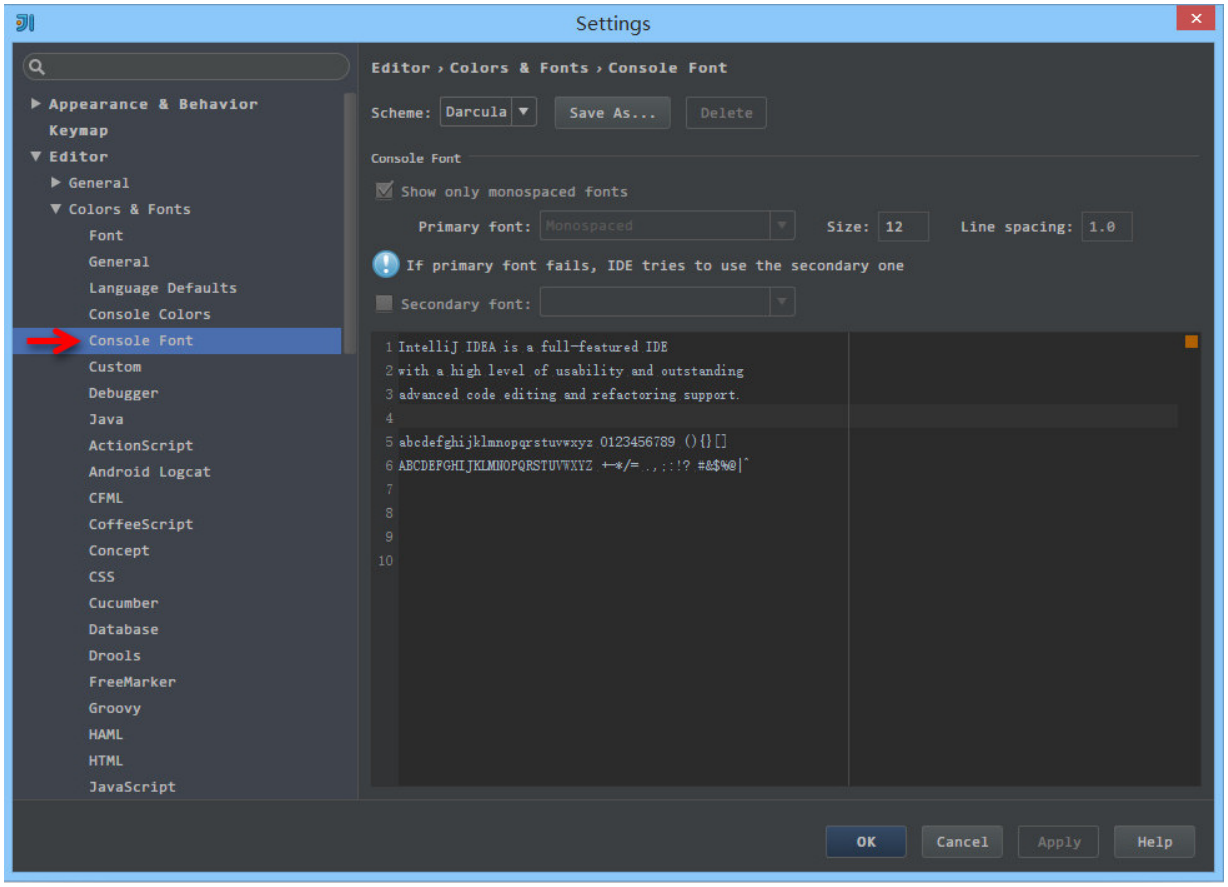
代码编辑字体修改



- 如上图标注 1 所示，默认 IntelliJ IDEA 是不能直接在默认的代码模板上修改字体的，需要先 Save As 一份出来，然后才可以修改。这种设计在 IntelliJ IDEA 其他很多设置也是如此的，所以如果你还看到类似有 Copy、Save As 这类选项的按钮就要想到是此设计思想。
- 如上图标注 2 所示，勾选的 Show only monospaced fonts 表示筛选显示系统上的等宽字体。由于 Windows 系统上等宽字体并不多，勾选此选项出现的下拉字体可选择就很少。取消勾选之后，就可以显示系统上所有已安装的字体。
- 如上图标注 3 所示，其中编码字体有第一字体（Primary font） 和 第二字体（Secondary font）之分。当有些字符在第一字体支持不了的时候，会去使用第二字体进行支持。

- 我个人习惯上：英文字体使用 Monaco，由于此字体不支持中文，所以我把这个设置为第一字体，第二字体使用 Yahei Consolas Hybrid 进行支持，该字体含有中文。这两个字体都不是系统自带的，需要自行下载安装。
- 如果你的第一字体不包含中文的话，第二字体包含中文，那在有些地方也还是会出现 口口口口口 这类问题，比如 Ctrl + Shift + N 进行查找文件的时候，如果你输入中文也会变成 口口口口口 ，我个人文件名为中文的不多，所以就容忍了这种情况。如果你不愿意容忍这种情况，那还是回到最开始的要求：第一字体包含中文。

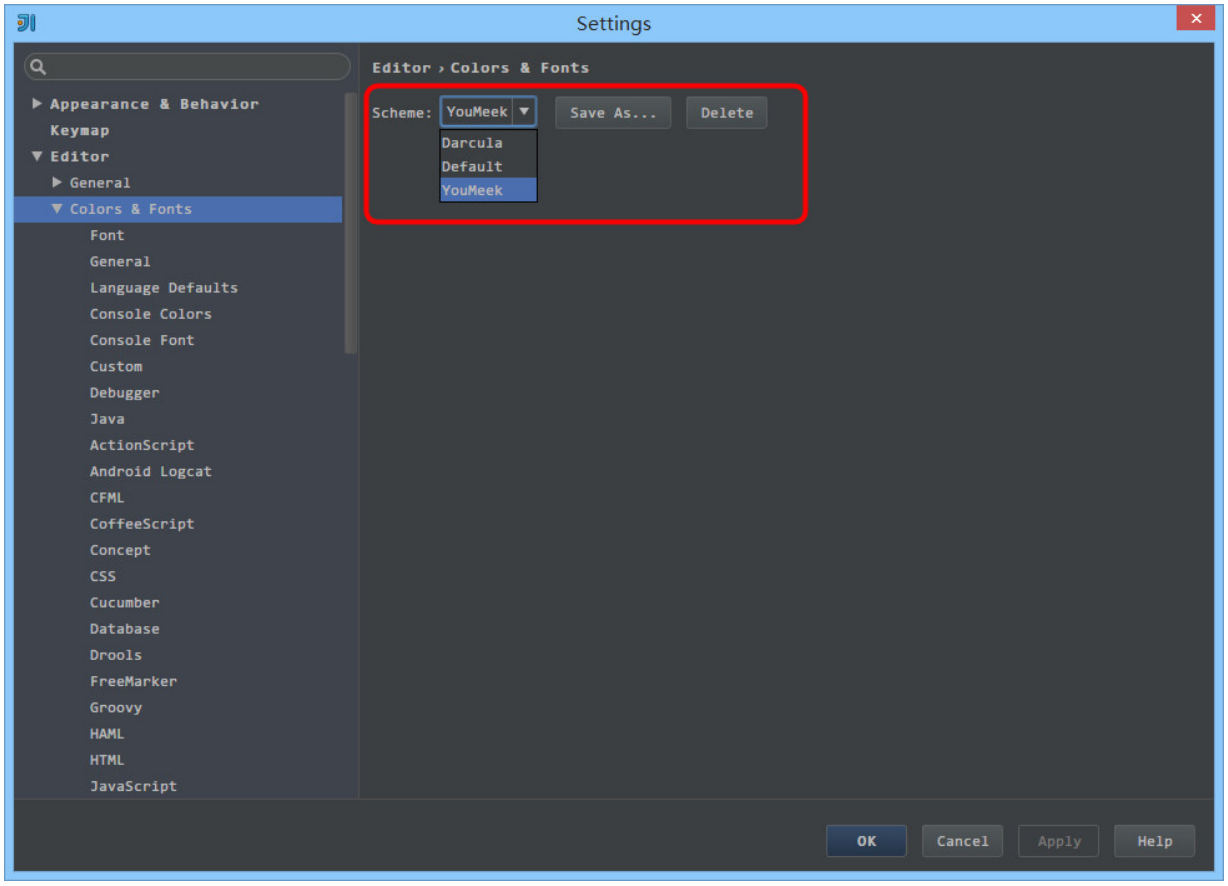
控制台输出字体修改



- 如上图为控制台输出内容字体修改，有很多 IntelliJ IDEA 新人在做输出的时候出现乱码原因就是因为没有在这里进行设置。
- *控制台输出字体* 修改的原理跟 *代码编辑字体修改* 是一样的，所以这里不进行讲解。

编辑区主题修改

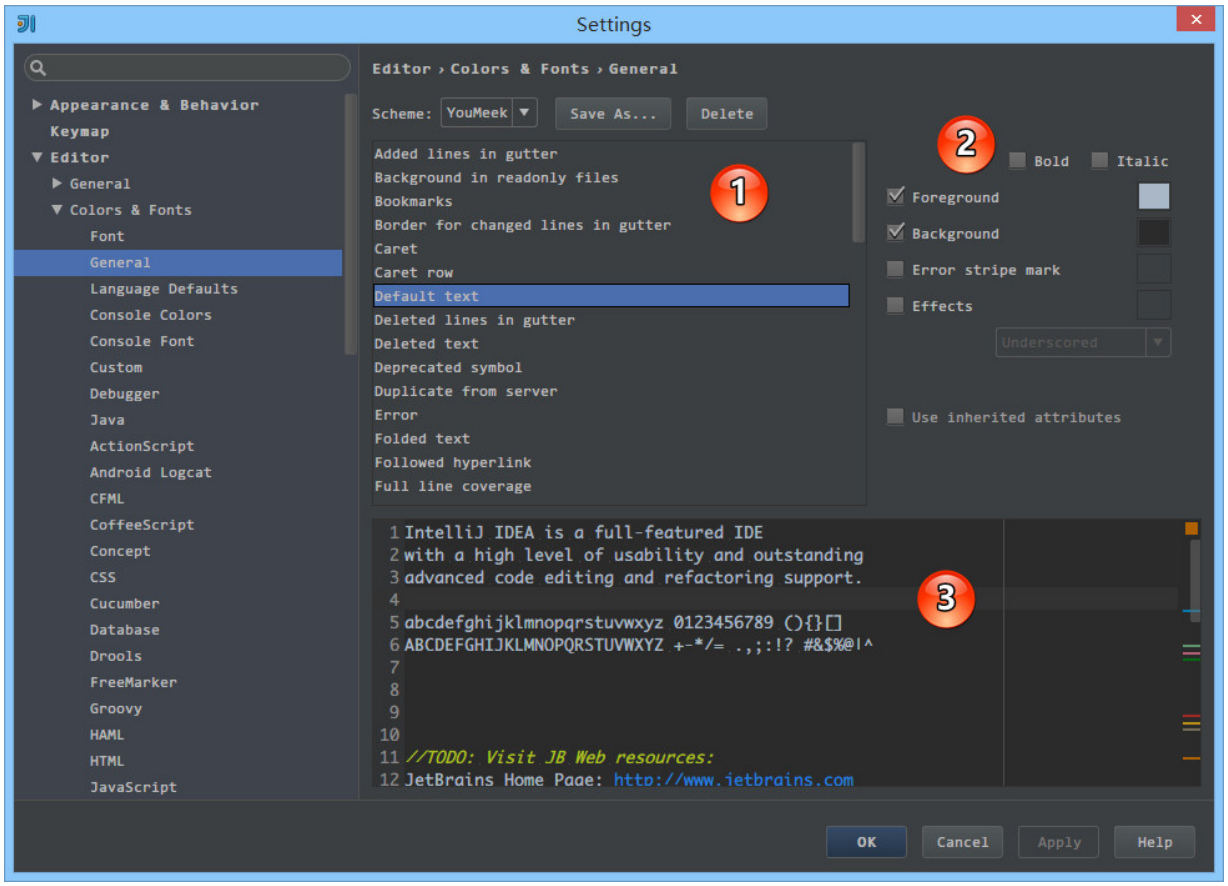
编辑区主题介绍



- 编辑区主题，也就是代码书写区的主题修改。基本上大家在 General 上都有对此进行小修小改，我下面也主要介绍下我个人在 General 上常修改的一些地方，其他特性的颜色修改我一般默认，但是修改方法原理一样。
- 如上图红圈下拉所示，展示的是我当前电脑可以选择的编辑区主题。
- 对于编辑区的主题，也有人制作成模板在网络上提供下载。这里主要介绍两个站点：

- http://www.ideacolorthemes.org/themes/，主要提供 jar 文件下载。
- http://www.phpstorm-themes.com/，主要提供 xml 和 icl 文件下载。
- 对应文件如何安装请查看网站对应的 Help 页面，都有详细说明的。

编辑区主题细节修改



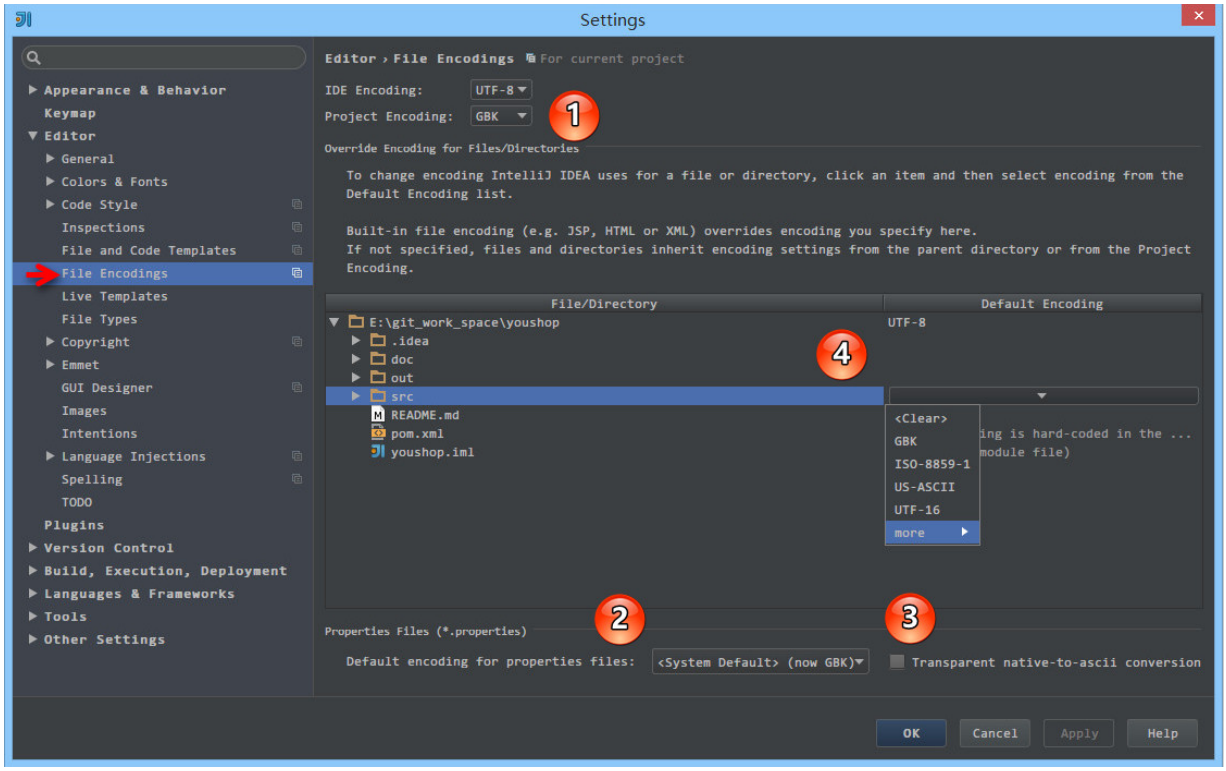
- 上图标注 1 为可修改的通用细节内容
- 上图标注 2 为可修改属性，其中并不是每个细节都可以修改所有属性的。比如细节：Default text 是可以勾选 Bold，而

Caret row 则是无法勾选 Bold，因为只有文本才有加粗的属性需求。

- 上图标注 3 为修改后的预览区，预览区是实时动态展示的。
- 在 General 区，我常修改的有：

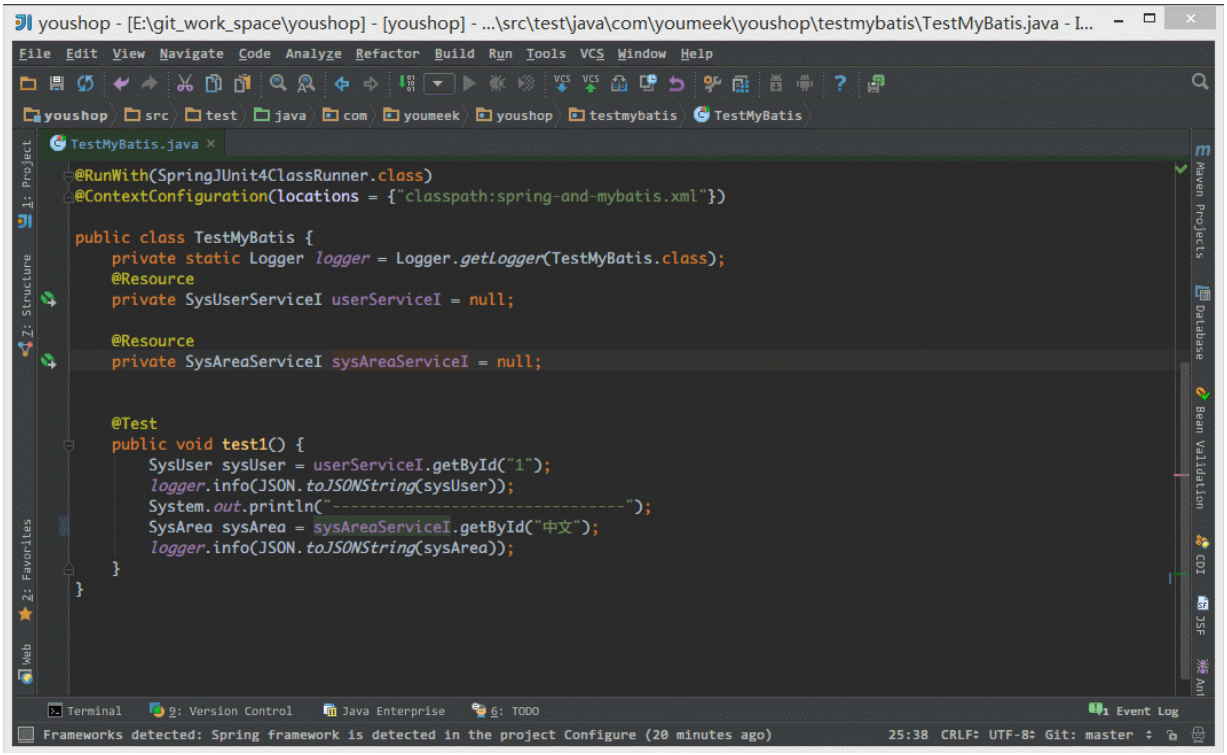
  - Default text，指的是默认代码文本，我一般会修改其 Background 属性。
  - Caret row，指的是光标所在行，我一般会修改其 Background 属性。
  - Vertical indent guide，指的是垂直缩进线，我一般会修改其 Foreground 属性。
  - Identifier under caret，指的是光标所在位置的相同标识符呈现什么效果，我一般会修改其 Background 属性。
  - Text search result，指的是在查找模式下，匹配字符的样式，我一般会修改其 Background 属性。

文件编码修改



- 上图标注 1 所示，IDE 的编码默认是 UTF-8，Project Encoding 虽然默认是 GBK，但是一般我都建议修改为 UTF-8。
- 上图标注 2 所示，IntelliJ IDEA 可以对 Properties 文件进行专门的编码设置，一般也建议改为 UTF-8，其中有一个重点就是属性 Transparent native-to-ascii conversion，
- 上图标注 3 所示，对于 Properties 文件，重要属性 Transparent native-to-ascii conversion 主要用于转换 ascii，一般都要勾选，不然 Properties 文件中的注释显示的都不会是中文。
- 上图标注 4 所示，IntelliJ IDEA 除了支持对整个 Project 设置编码之外，还支持对目录、文件进行编码设置。如果你要

对目录进行编码设置的话，可能会出现需要 Convert 编码的弹出操作选择，**强烈建议** 在转换之前做好文件备份，不然可能出现转换过程变成乱码，无法还原。



- 如上图演示，对单独文件的编码修改还可以点击右下角的编码设置区。如果代码内容中包含中文，则会弹出演示中的操作选择。

  - Reload 表示使用新编码重新加载，新编码不会保存到文件中，重新打开此文件，旧编码是什么依旧还是什么。
  - Convert 表示使用新编码进行转换，新编码会保存到文件中，重新打开此文件，新编码是什么则是什么。
  - 含有中文的代码文件，Convert 之后可能会使中文变成乱码，所以在转换成请做好备份，不然可能出现转换过程变成乱码，无法还原。

由于编码问题引起的编译错误

- 编译报错：找不到符号、 未结束的字符串文字 等的解决办法：
    - 由于 UTF-8 编码文件有分 有BOM 和 无BOM 之分，默认情况下 IntelliJ IDEA 使用的编译器是 javac，而此编译只能编译 无BOM 的文件，有很多 Eclipse 用户在使用 IntelliJ IDEA 开发 Eclipse 项目的时候常常会遇到此问题。主要是因为 Eclipse 的编译器是 Eclipse，此编译器支持 有BOM 的文件编译。顾，解决办法是对于此文件进行 BOM 去除。
    - 批量去除 BOM，你可以 Google：批量去除 BOM、批量转换无 BOM 等关键字，网络上已有提供各种方案。
    - 除了通过去除 BOM 还有设置 IntelliJ IDEA 的编译器为 Eclipse，但是一般不建议这样做。
    - 如果上述问题都无法解决，而且你也确认 IntelliJ IDEA 各个配置编码的地方都是 UTF-8，报错文件编码也是是 UTF-8 无 BOM 的话，那还有一种可能也会出现这种情况：项目配置文件有问题。项目编码的配置文件在：/项目目录/.idea/encodings.xml。如果你会修改此文件可以进行修改，如果不会，那就删除掉 .idea 整个目录，重启 IntelliJ IDEA 重新配置这个项目即可。

IntelliJ IDEA 缓存和索引介绍和清理方法

缓存和索引介绍

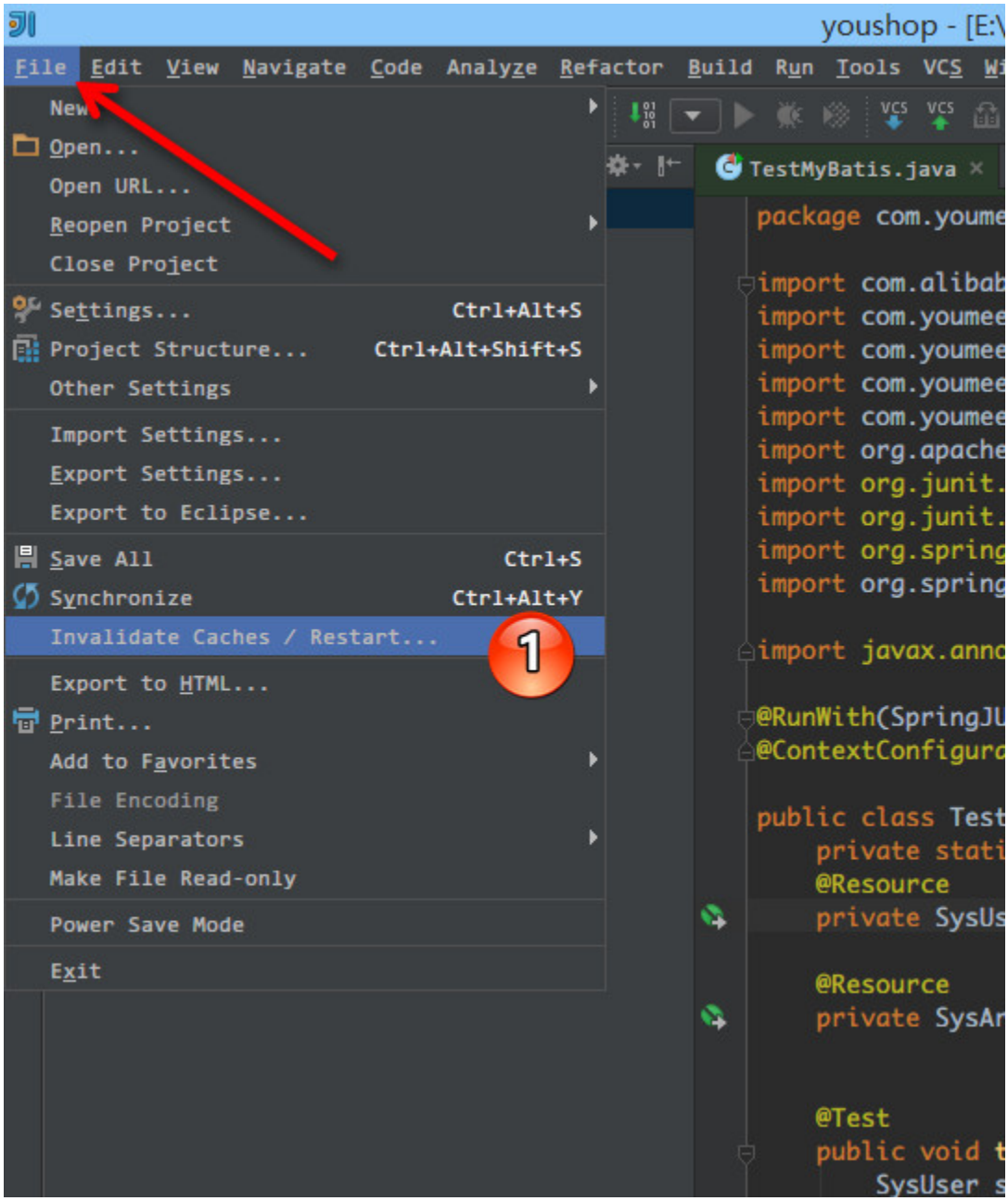
在《IntelliJ IDEA 界面介绍》章节里已经点到了 IntelliJ IDEA 首次加载项目的时候，都会创建索引，而创建索引的时间跟项目的文件多少成正比，我也简单强调了 IntelliJ IDEA 索引的重要性。这里我们再对此进行详细说明索引、缓存对 IntelliJ IDEA 的重要性。

通过《常见文件类型的图标介绍》章节，你已经认识到 IntelliJ IDEA 下各个文件类型的图标是什么样子的。其中有一个图标我是专门进行了讲解：  `Java class located out of the source root`。我们也都知道该图标是表示 Java 类文件没有在 `Source root` 目录下的文件夹下会显示此图标，但是其实还有一种情况也是会显示此图标的。那就是：在 IntelliJ IDEA 创建索引过程中，所有的 Java类 都是这个图标，如果你项目大的话很容易观察到的，几个文件的小项目倒是不一定会看到。所以在 IntelliJ IDEA 创建索引过程即使你编辑了代码也是编译不了、运行不起来的，所以还是安安静静等 IntelliJ IDEA 创建索引完成。

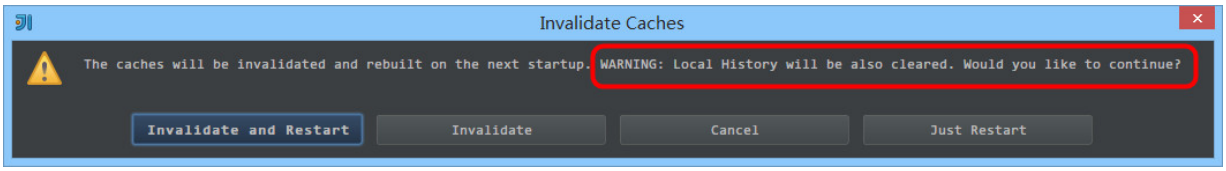
IntelliJ IDEA 的缓存和索引主要是用来加快文件查询，从而加快各种查找、代码提示等操作的速度，所以 IntelliJ IDEA 的索引的重要性我再唠叨一万遍都不为过。但是，IntelliJ IDEA 的索引和缓存并不是一直会良好地支持 IntelliJ IDEA 的，这某些特殊条件下，IntelliJ IDEA 的缓存和索引文件也是会损坏的，比如：断电、蓝屏引起的强制关机，当你重新打开 IntelliJ IDEA，基本上百分八十的可能 IntelliJ IDEA 都会报各种莫名其妙错误，甚至项目打不开，IntelliJ IDEA 主题还原成默认状态。也有一些即使没有断电、蓝屏，也会有莫名奇怪的问题的时候，也很有可能是 IntelliJ IDEA 缓存和索引出问题，这种

情况还不少。遇到此类问题也不用过多担心。，下面就来讲解如何解决。

清除缓存和索引



- IntelliJ IDEA 已经自带提供清除缓存、索引的路口，如上图标注 1 所示。

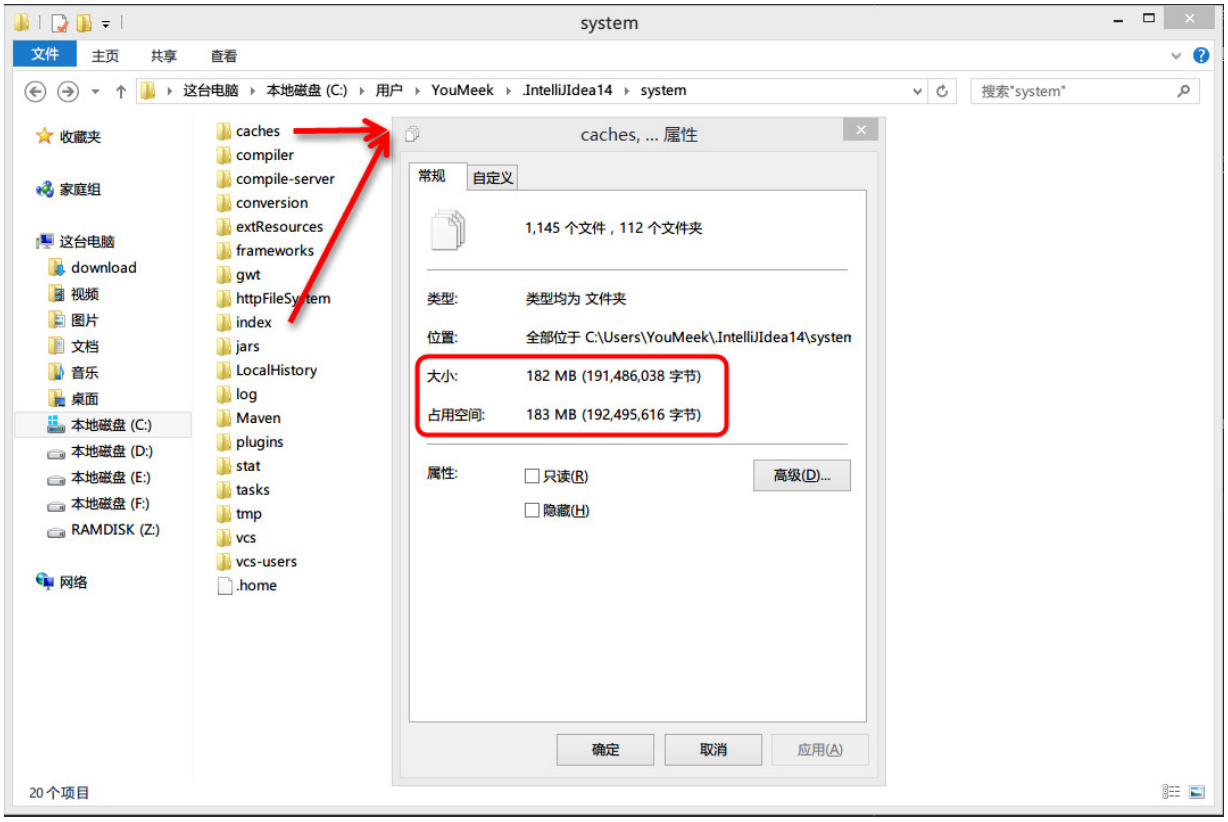


- 一般建议点击 Invalidate and Restart，这样会比较干净。
- 但是有一个需要提醒的是，如上图红圈标注的地方：清除索引和缓存会使得 IntelliJ IDEA 的 Local History 丢失，所以如果你项目没有加入到版本控制，而你又需要你项目文件的历史更改记录，那你最好备份下你的 LocalHistory 目录。目录地址在：C:\Users\当前登录的系统用户名\.IntelliJIdea14\system\LocalHistory 建议使用硬盘的全文搜索，这样效率更高。

通过上面方式清除缓存、索引本质也就是去删除 C 盘下的 system 目录下的对应的文件而已，所以如果你不用上述方法也可以删除整个 system。当 IntelliJ IDEA 再次启动项目的时候会重新创建新的 system 目录以及对应项目缓存和索引。

如果你遇到了因为索引、缓存坏了以至于项目打不开，那也建议你可以直接删除 system 目录，一般这样都可以很好地解决你的问题。

其他

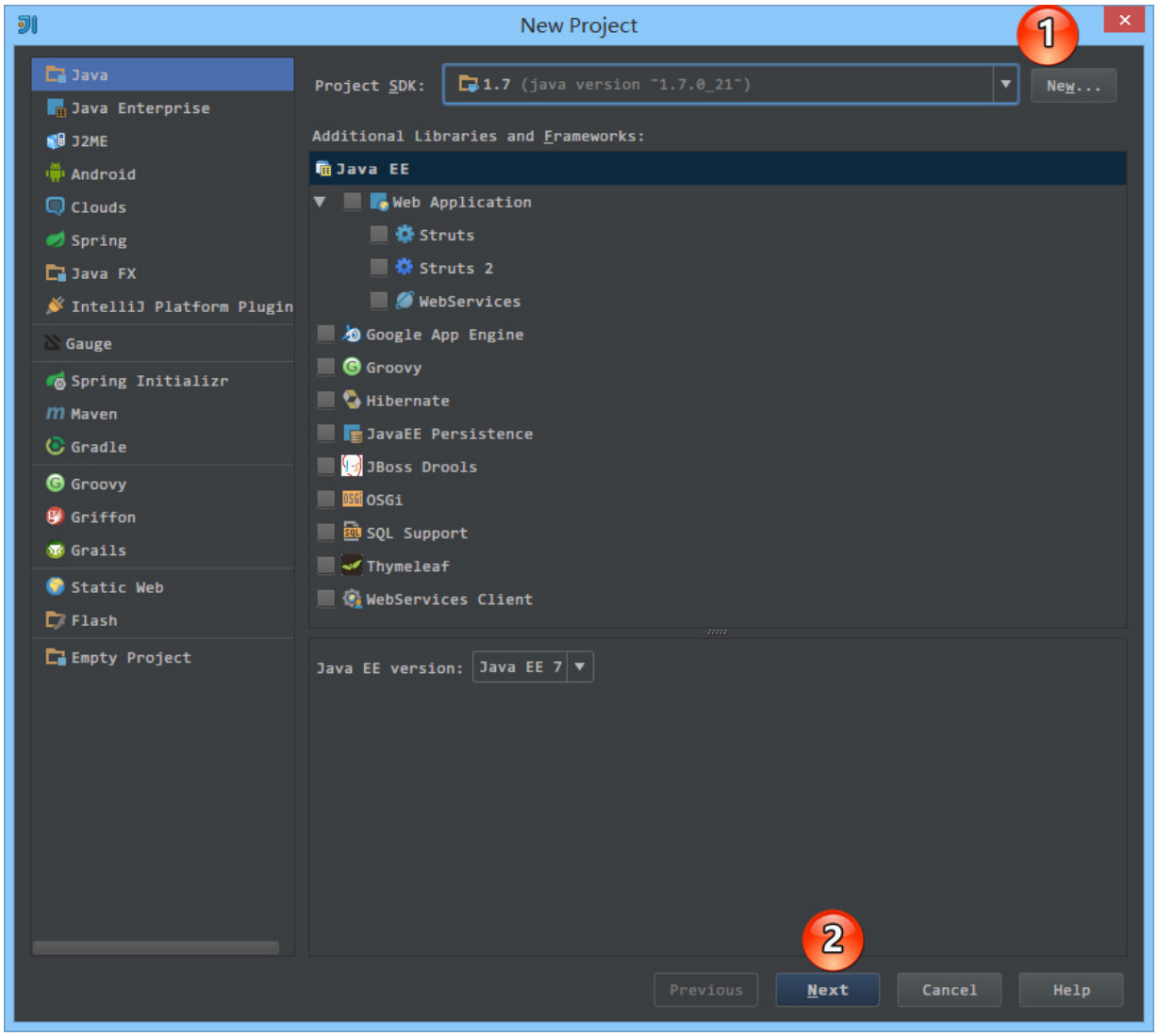


- 目前我电脑的 IntelliJ IDEA 是新装的，也就打开了几个小项目，所有打开的项目大小加起来不到 5M，但是他们创建的索引大家就已经上百兆了，如上图所示。所以如果你 C 盘空间不足的情况下，最好转移下 `system` 目录，方法可以根据《IntelliJ IDEA 相关核心文件和目录介绍》讲解的方法进行。

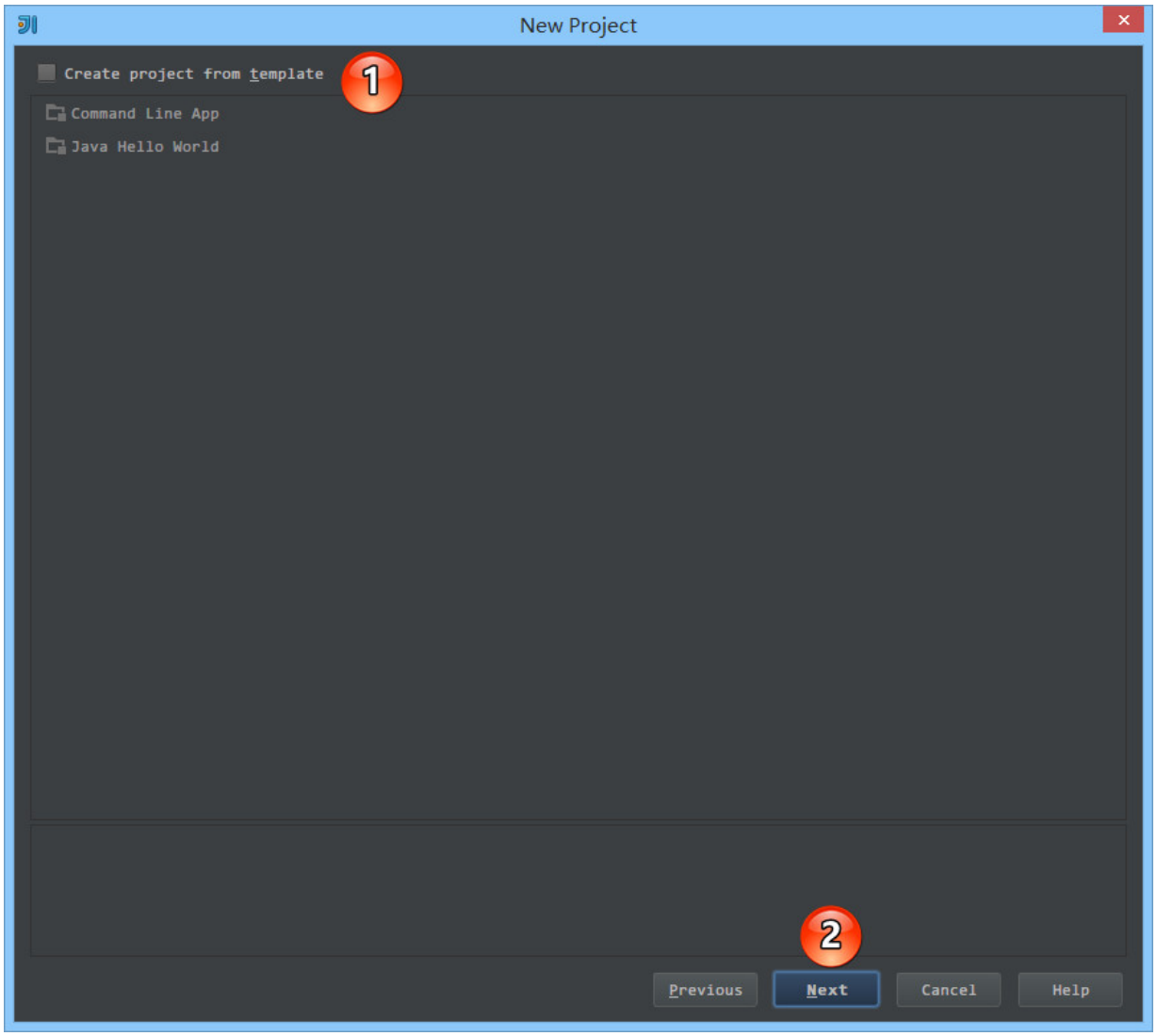
Hello World 项目创建与项目配置文件介绍

Hello World 项目创建

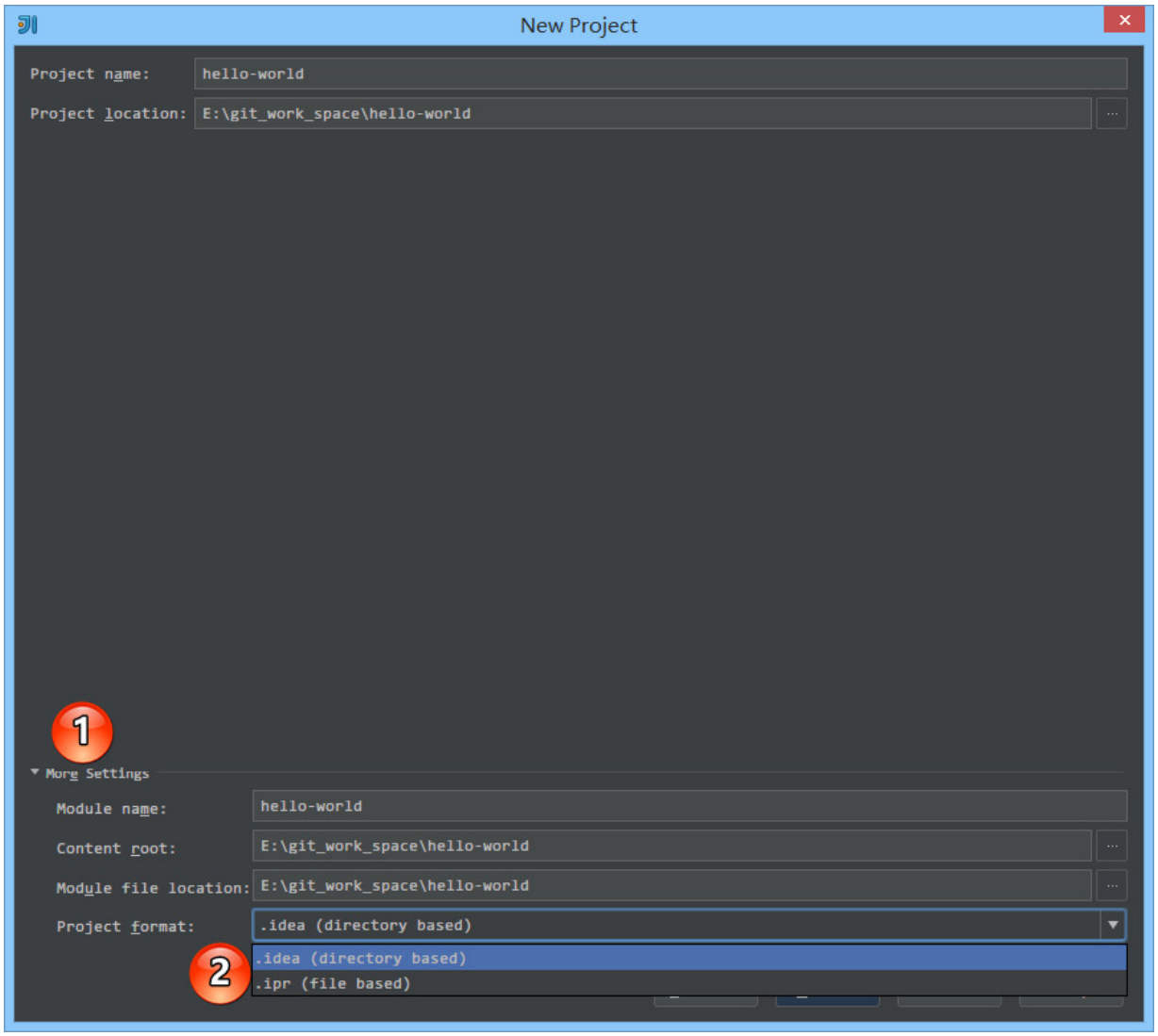
- 如上图标注 1 所示，点击 `Create New Project` 进入向导式创建项目



- 如上图标注 1 所示，如果此时 IntelliJ IDEA 还没有配置任何一个 SDK 的话，可以点击 New... 先进行 SDK 的配置。
- 如上图标注 2 所示，配置好 SDK 或选择好 SDK 之后，点击 Next 进入下一步。



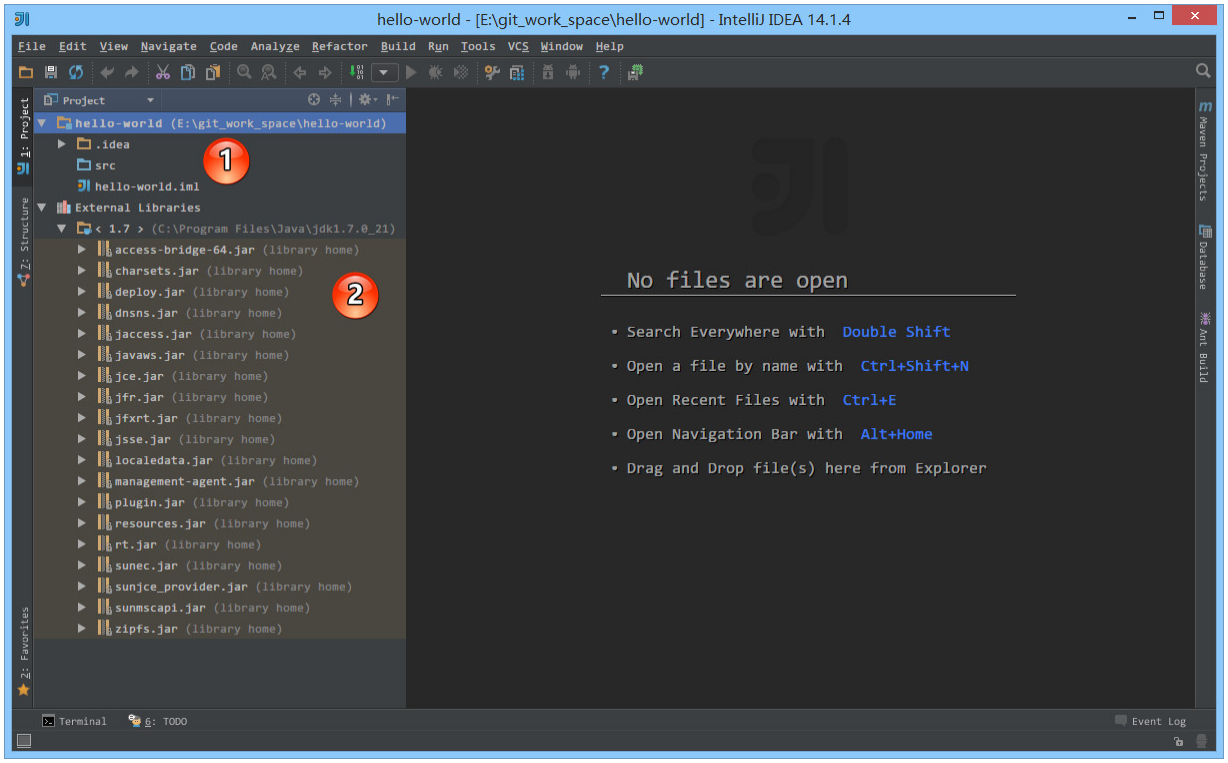
- 如上图标注 1 所示，可以选择模板快速创建项目。
  - `Command Line App` 会自动创建一个带有 main 方法的类。
  - `Java Hello World` 会自动创建一个带有 main 方法的并且会打印输出 Hello World 的类。
  - 我们这里不勾选使用模板，而是手工创建，所以我们点击上图标注 2，进入下一步。



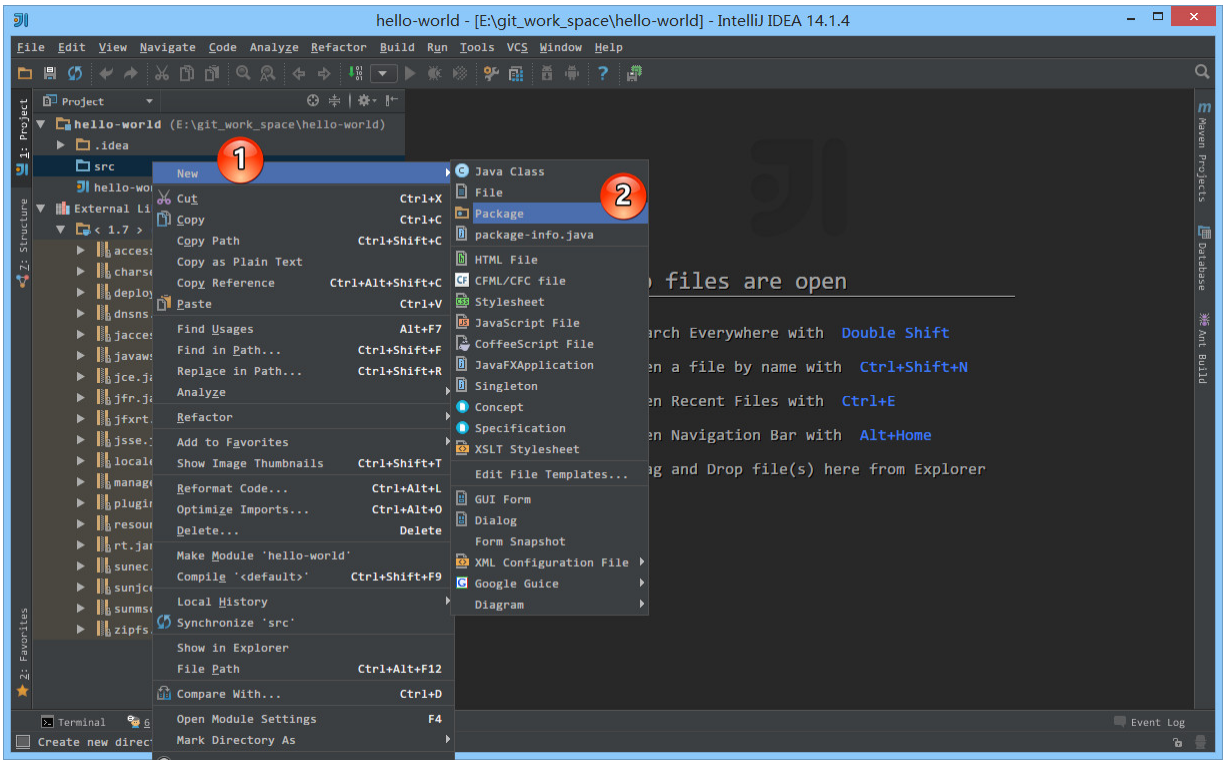
- 如上图标注 1 所示，默认 More Settings 是没有展开的，点击此处可以展开更多细节的信息。
- 如上图标注 2 所示，IntelliJ IDEA 的项目格式文件主要提供两种方式

    - .idea (directory based) 创建项目的时候自动创建一个 .idea 的项目配置目录来保存项目的配置信息。这是默认选项。

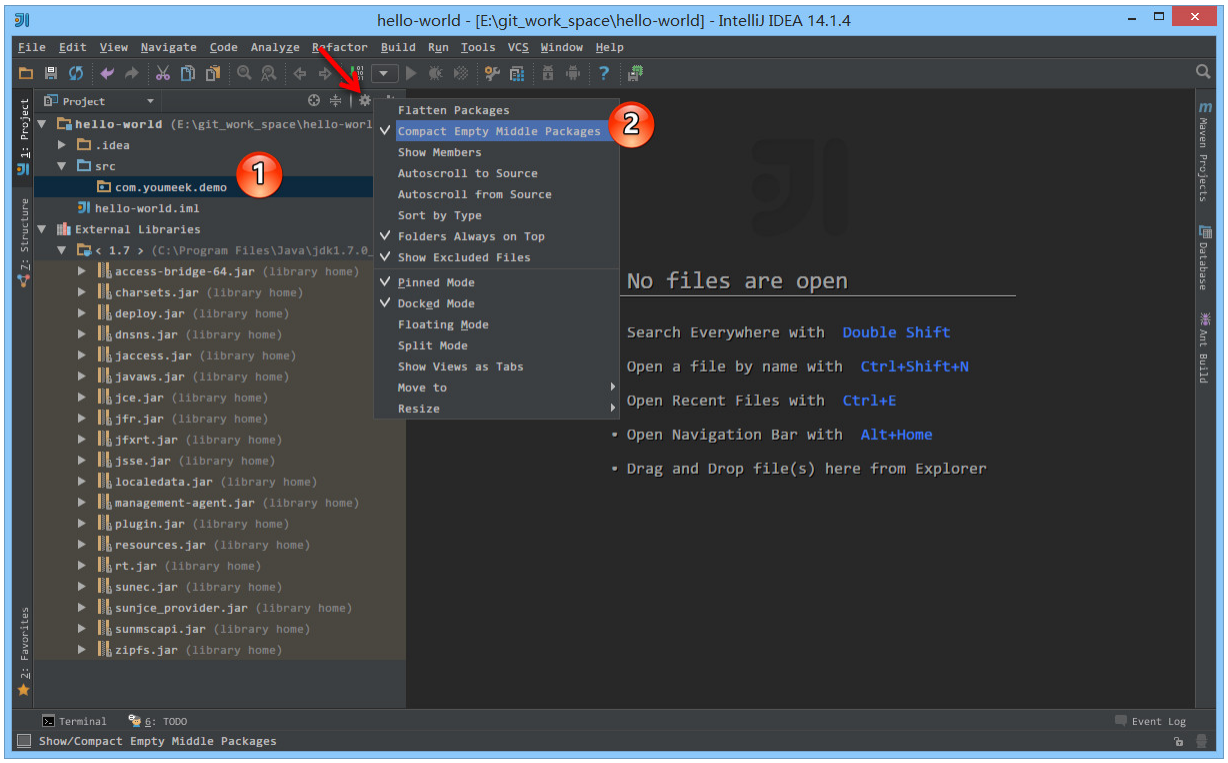
- `.ipr (file based)` 创建项目的时候自动创建一个 `.ipr` 的项目配置文件来保存项目的配置信息。
- 需要特别注意的是，我这边默认创建的项目编码是 GBK，而如果你需要 UTF-8 的话，修改编码的方式请看第 10 讲。



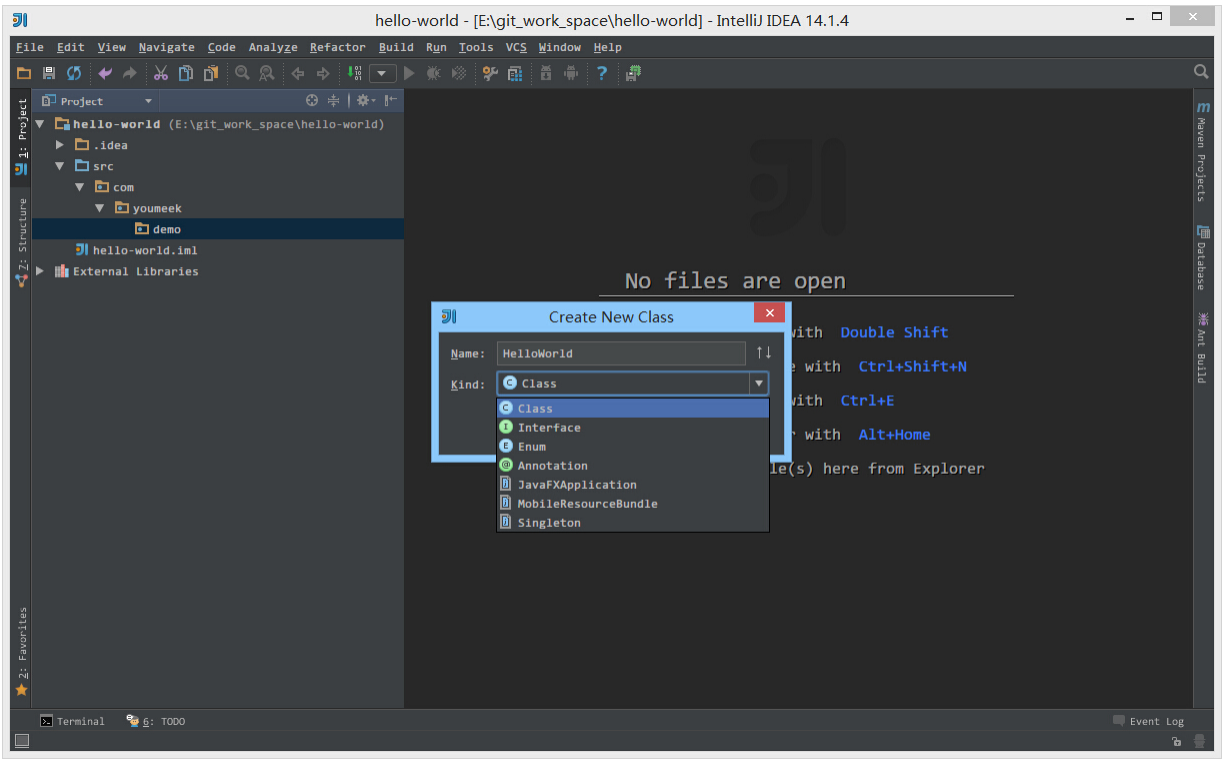
- 如上图标注 1 所示，根据《常见文件类型的图标介绍》章节我们知道，`src` 目录为蓝色表示 `Source root`，我们可以再此目录下创建包和类。
- 如上图标注 2 所示，由于该项目使用的是 JDK 7，所以项目是基于 JDK 7，我们可以调用 JDK 7 中的类。



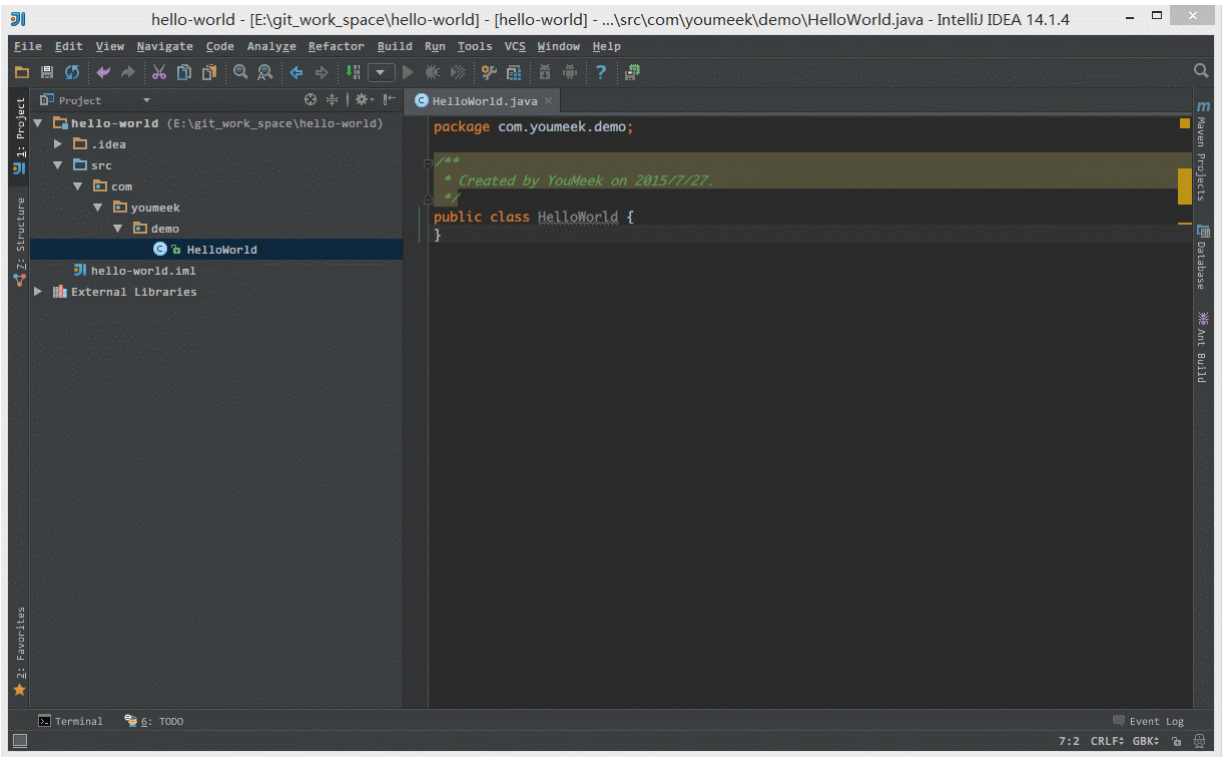
- 如上图标注 1，2 所示，在 src 目录右键，选择 New 创建包目录。



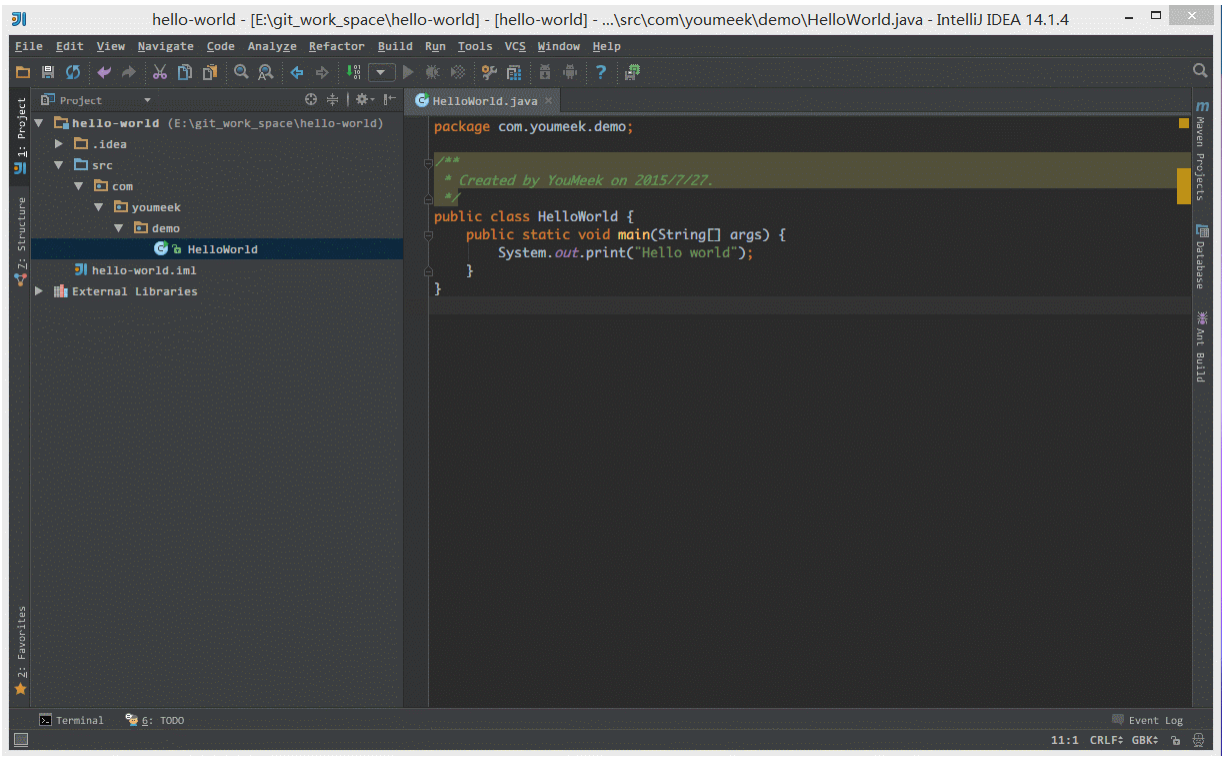
- 如上图标注 1 所示，在没有文件的情况下包目录默认是连在一起的，这不方便看目录层级关系。
- 如上图标注 箭头 所示，点击此齿轮，在弹出的菜单中去掉选择标注 2 选项：`Compact Empty Middle Packages`。



- 如上图所示，在包下可以直接创建 Class、Interface、Enum、Annotation 等常见类型文件。

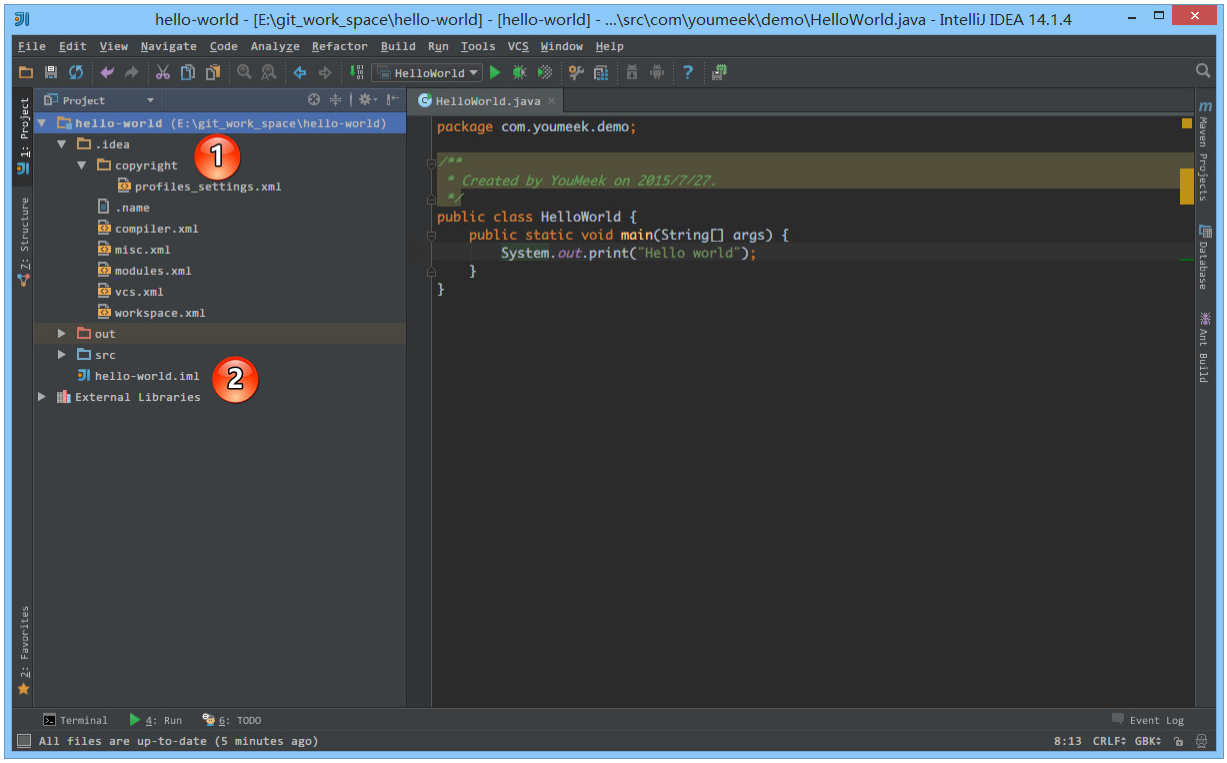


- 如上图 Gif 演示，写一个 main 方法，打印输出 Hello world。



- 如上图 Gif 演示，运行 main 方法，打印输出 Hello world。

项目配置文件介绍



- 如上图标注 1 所示，.idea 即为 Project 的配置文件目录。
- 如上图标注 2 所示，.iml 即为 Module 的配置文件。
- 通过上面的了解我们也知道 IntelliJ IDEA 项目的配置变动都是以这些 XML 文件的方式来表现的，所以我们也可以通过了解这些 XML 文件来了解 IntelliJ IDEA 的一些配置。也因为此特性，所以如果在项目协同中，我们要保证所有的项目配置一致，就可以考虑把这些配置文件上传到版本控制中（包括 .idea 目录和 .iml 文件）。如果把这些文件加入到版本控制之后，那又有一点是需要考虑的，那就是协同者 Checkout 项目下来之后，按自己的需求进行项目配置的之后，项目的 XML 文件也会跟着变化。此时协同者的这些变化的文件就不应该再上传到版本控制中。至于如何更好地控

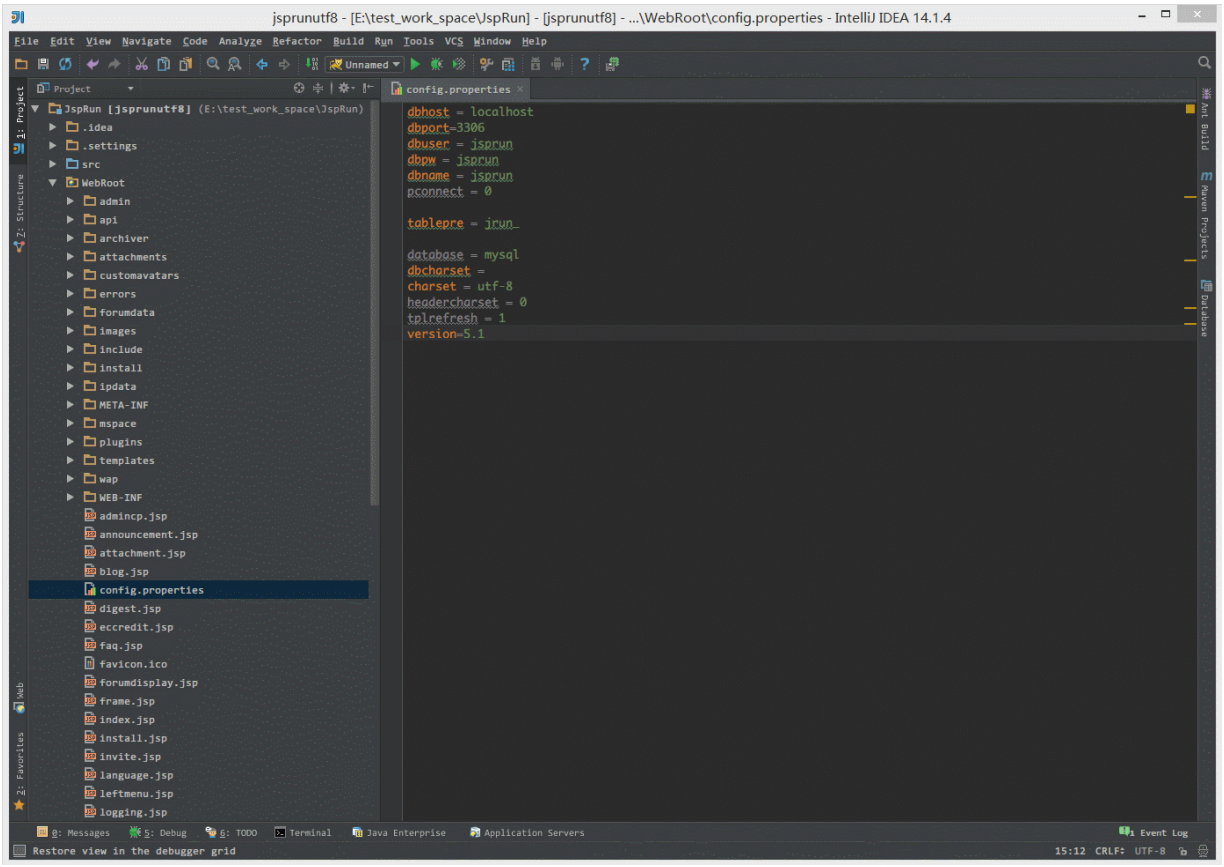
制这些不想随时提交的文件，在接下来的版本控制专讲中会进行详细讲解。

特别介绍

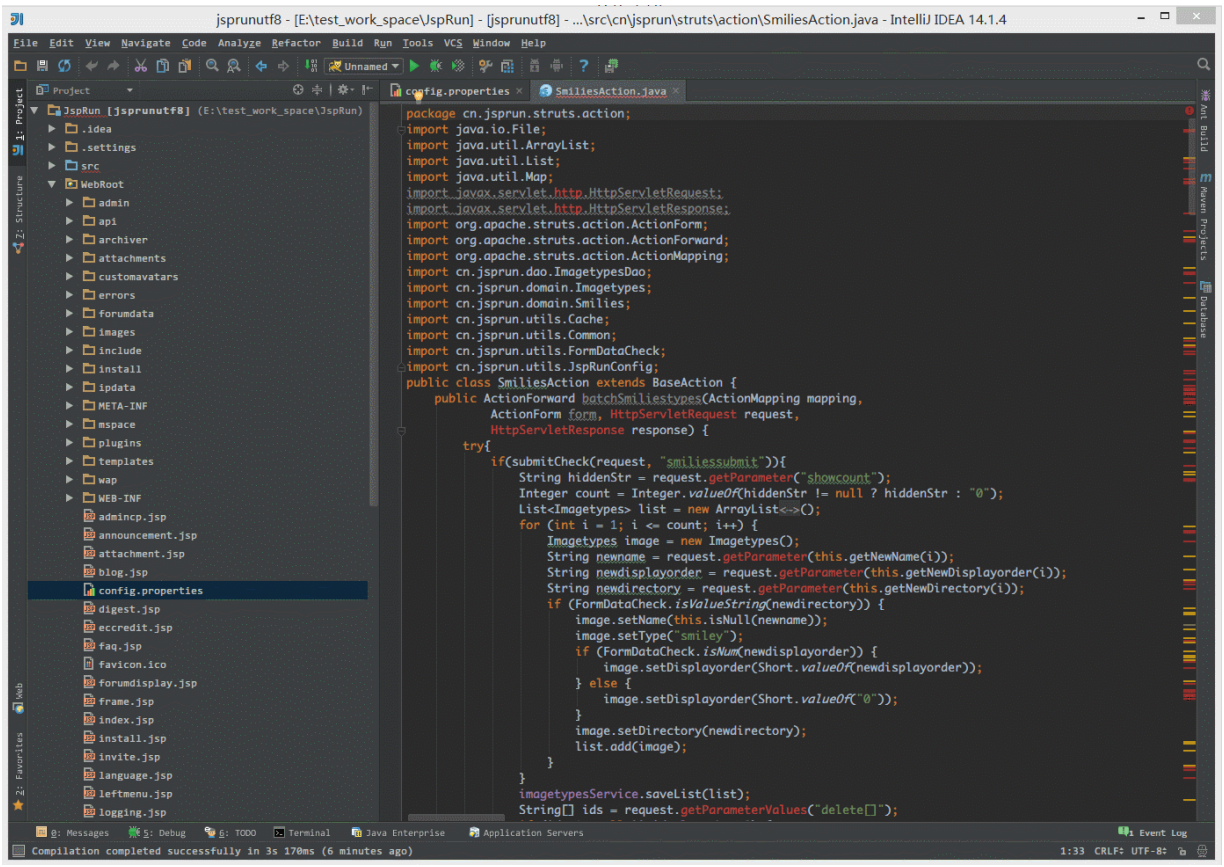
- IntelliJ IDEA 是一个没有 `Ctrl + S` 的 IDE，所以每次修改完代码你只要管着运行或者调试即可，无需担心保存或者丢失代码。
- 现在，放心、大胆地开始你的 Hello World。

CREATE DATABASE `<span class="pl-en">jsprun</span>` CHARACTER SET utf8; CREATE USER 'jsprun'@'localhost' IDENTIFIED BY 'jsprun'; GRANT ALL PRIVILEGES ON jsprun.* TO 'jsprun'@'localhost'; FLUSH PRIVILEGES;

- 切换到上面新建的 `jsprun` 数据库中执行项目中这个数据脚本，文件位置：`JspRun\WebRoot\install\jsprun_zh_CN.sql`。
- 修改 `JspRun\WebRoot\config.properties` 文件中的几个属性为下面内容：
- `dbuser = jsprun`
- `dbpw = jsprun`



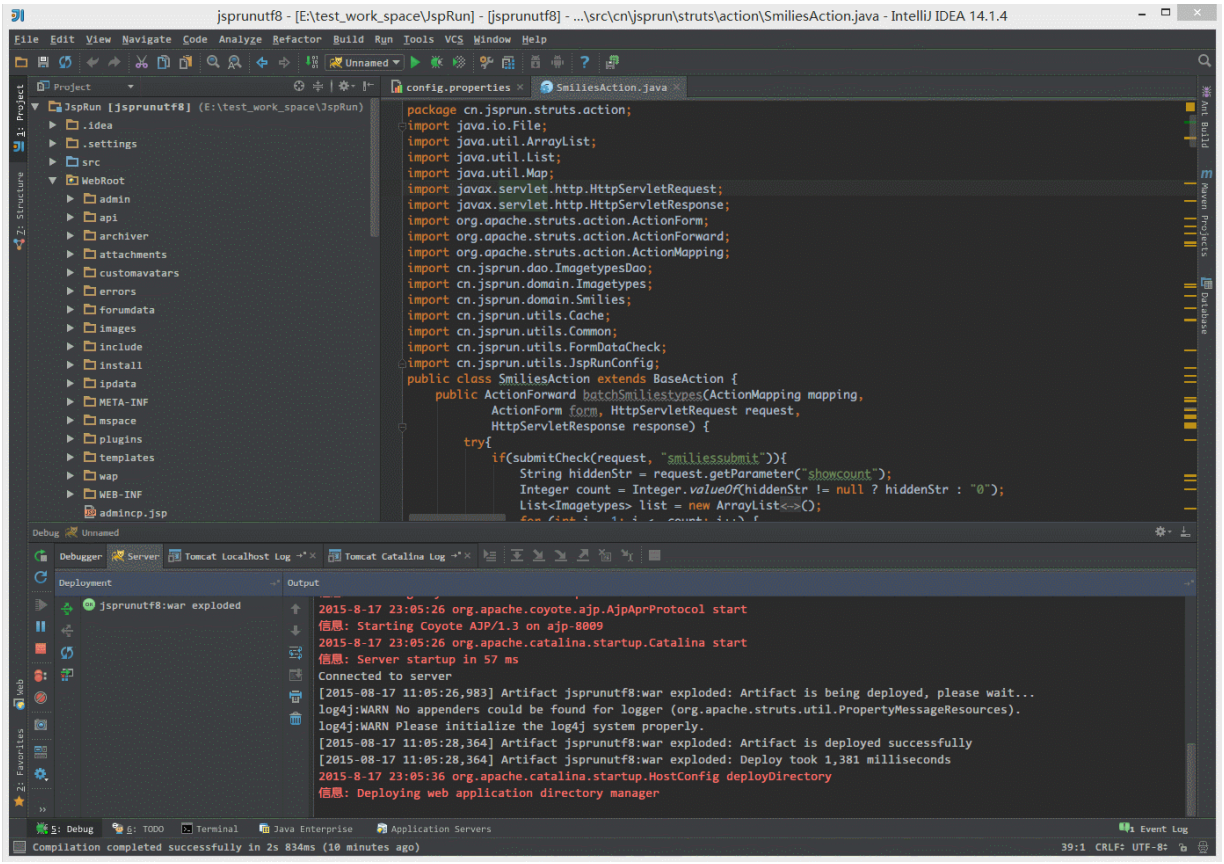
- 如上图 Gif 演示，我们缺少引入 Tomcat 的依赖包。



- 如上图 Gif 演示，我们引入 Tomcat 的依赖包之后，可以运行该项目。

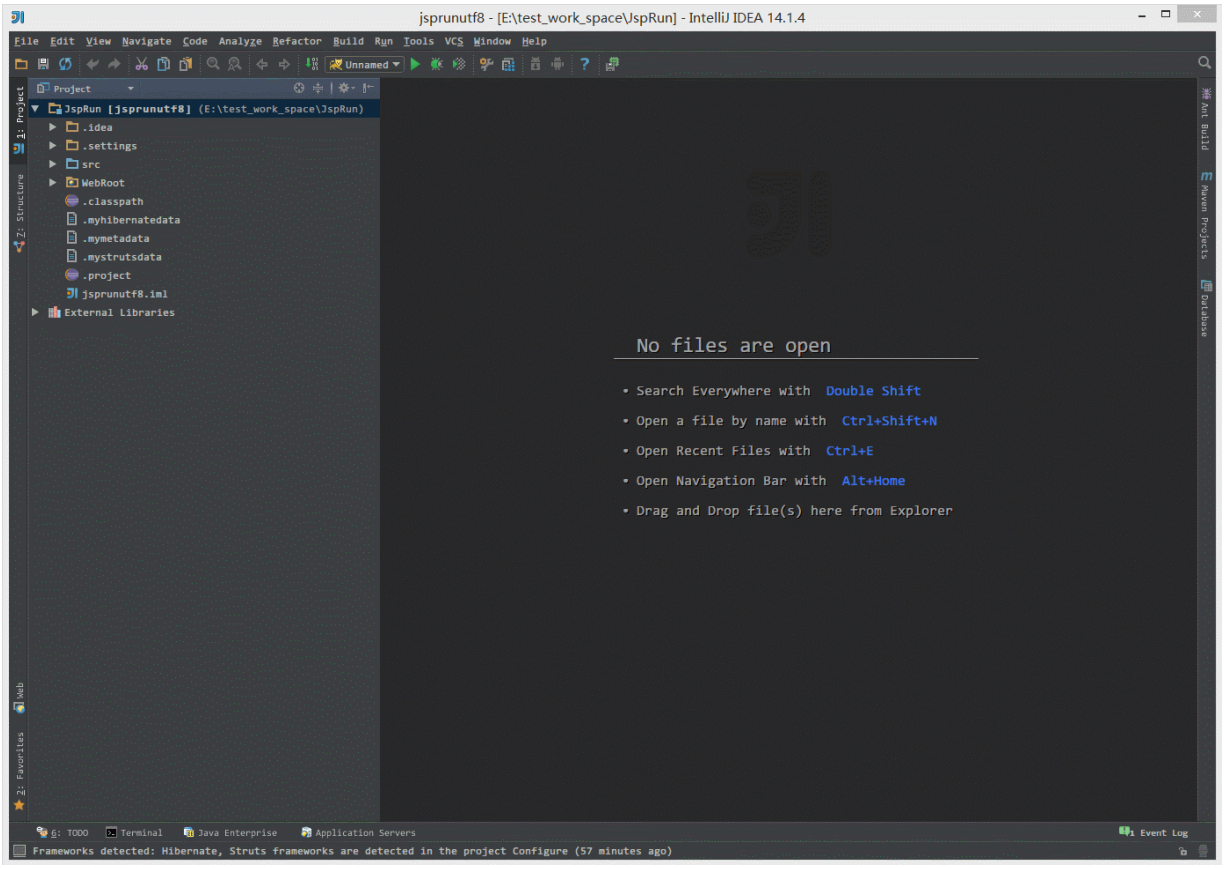
- 如上图所示，为最终项目运行效果图。

Tomcat 停止



- 如上图所示，停止按钮是要按两次，第一按完出现一个骷髅头并不是停止，需要再点击一次。
- 有时候即使点了两次，Tomcat 容器也不一定能完全停掉，这时候很容易出现端口被占用的操作，这时候你需要打开系统的资源管理器，手动 kill 系统上所有的 java 进程。

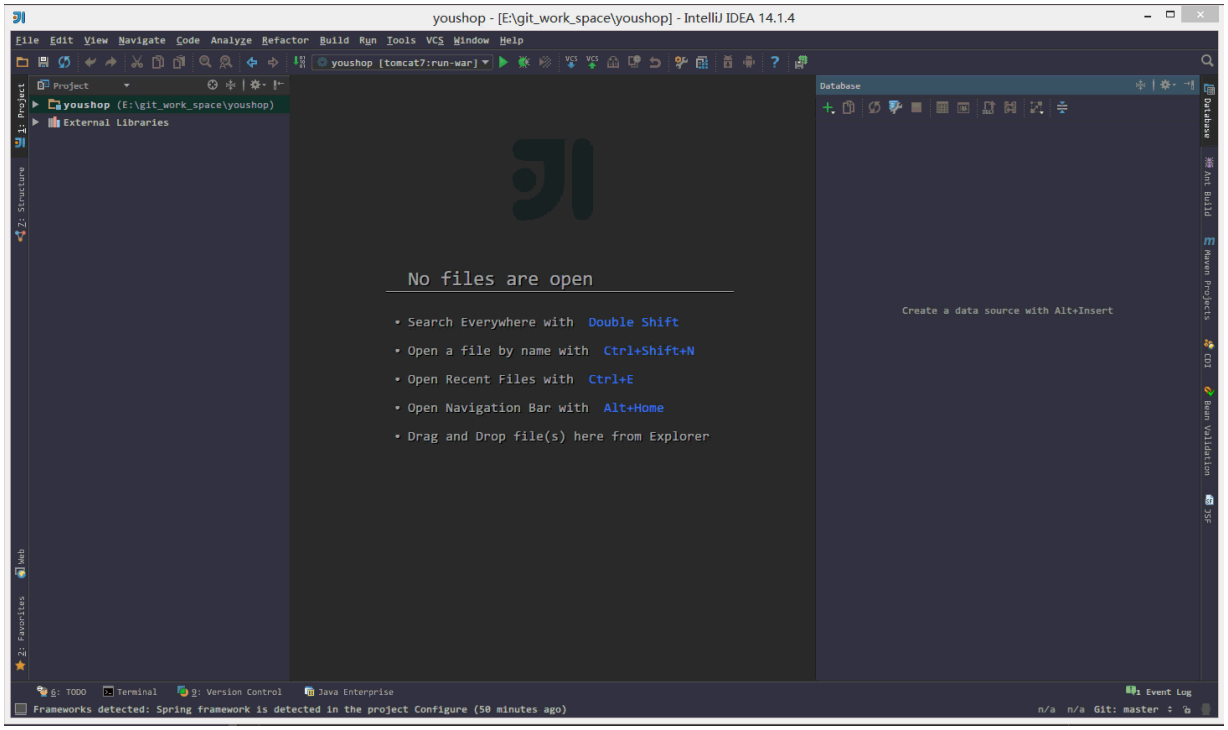
输出 war 压缩包



- 如上图 Gif 所示，除了在 Artifacts 中需要配置，还需要在容器中也跟着配置，这样在启动容器的时候才会输出一个 war 压缩包。
- 通过配置，我们也知道 war 的压缩包本质是根据展开的 war 输出包进行压缩的得来。

数据库管理工具介绍

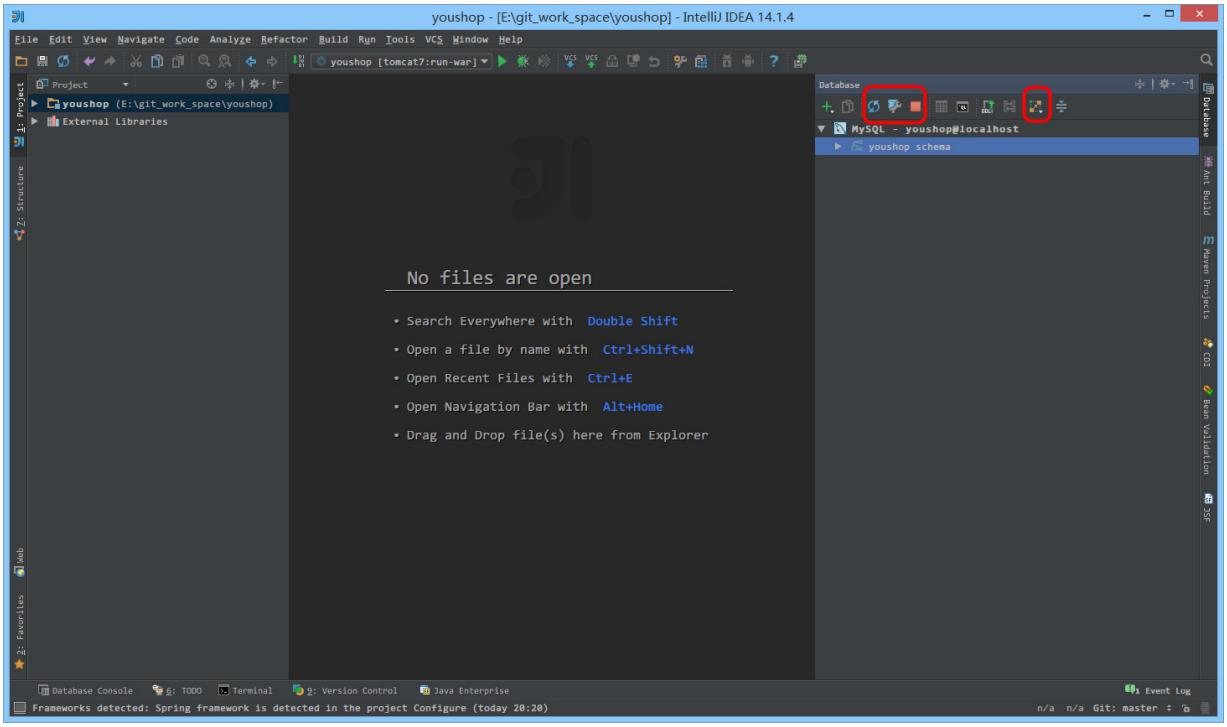
配置 Database 组件的数据库连接



- 表面上很多人认为配置 Database 就是为了有一个 GUI 管理数据库功能，但是这并不是 IntelliJ IDEA 的 Database 最重要特性。数据库的 GUI 工具有很多，IntelliJ IDEA 的 Database 也没有太明显的优势。IntelliJ IDEA 的 Database 最大特性就是对于 Java Web 项目来讲，常使用的 ORM 框架，如 Hibernate、Mybatis 有很好的支持，比如配置好了 Database 之后，IntelliJ IDEA 会自动识别 domain 对象与数据表的关系，也可以通过 Database 的数据表直接生成 domain 对象等等。

- 如上图 Gif 所示，这是一个完成的配置 Database 过程，对于数据库需要的依赖包，IntelliJ IDEA 可以自动帮我们下载，所以我们只要配置对应的连接参数即可。

Database 设置



- 如上图标注的红圈所示，这是 Database 常用的四个操作。
  - 第一个按钮是同步当前数据库连接。这个是最重要的操作，有一些情况下，当我们配置好连接之后，没有显示数据表，那就是需要点击该按钮进行同步。还有一种情况就是我们在 IntelliJ IDEA 之外用其他工具操作数据库，比如新建表。而此时 IntelliJ IDEA 的 Database 如果没有同步到新表，也是需要点击此按钮进行同步的。
  - 第二个按钮是配置当前连接，跟我们首次设置连接的界面是一样的。
  - 第三个按钮是断点当前的连接。

- 第四个按钮是查看当前所选对象的图标结构，比如我们当前选中的是整个数据库名，我们如果点击此按钮，则是显示该数据库下的所有数据表的图标结构图。

快捷键

说明

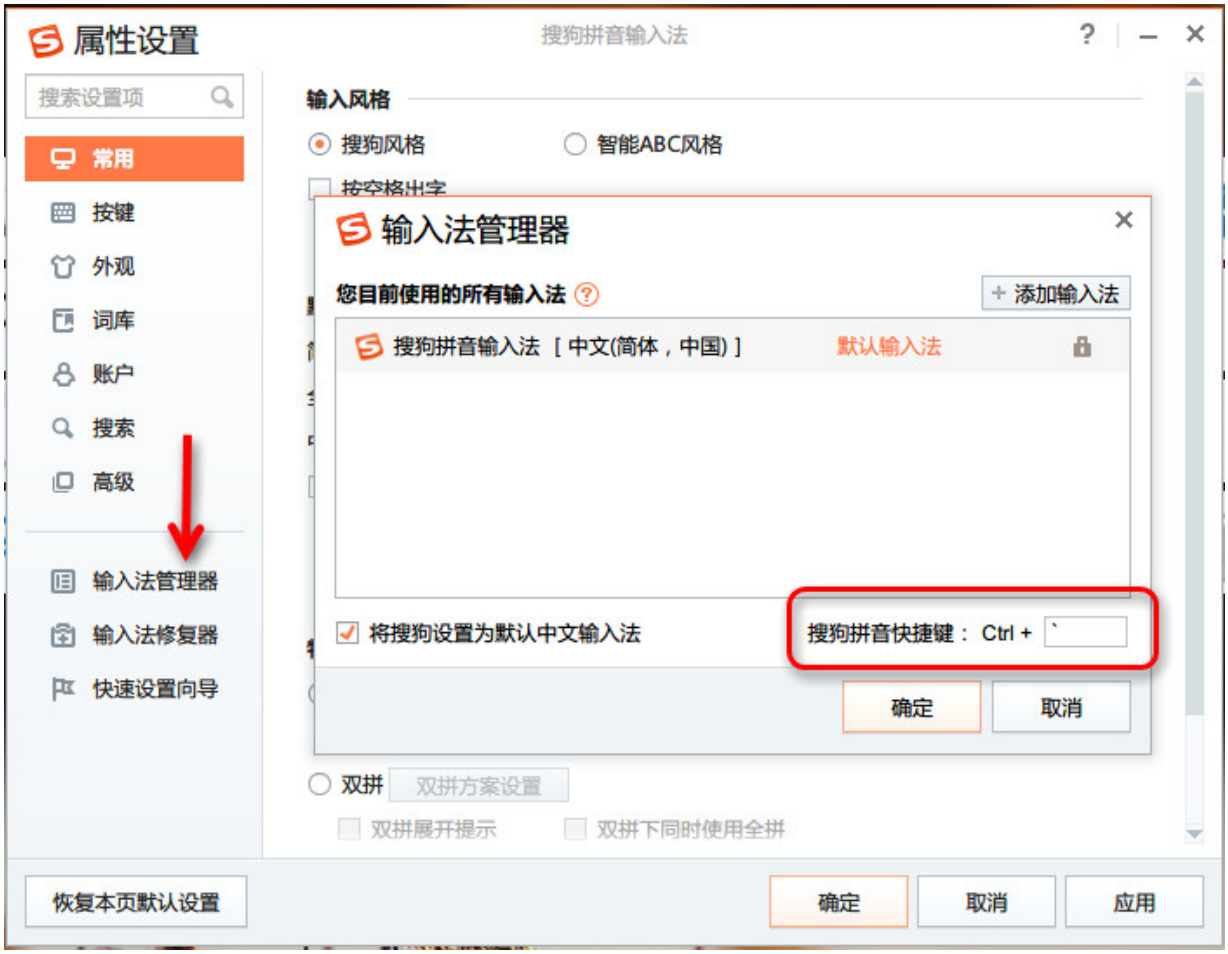
IntelliJ IDEA 的便捷操作性，快捷键的功劳占了一大半，对于各个快捷键组合请认真对待。IntelliJ IDEA 本身的设计思维是提倡键盘优先于鼠标的，所以各种快捷键组合层出不穷，对于快捷键设置也有各种支持，对于其他 IDE 的快捷键组合也有预设模板进行支持。

关于各个快捷键的频率分类上可能每个人都有各自的看法，下面的整理也只是已我个人的使用习惯来划分的，而我应该是可以代表某一部分小众人员。但是我个人还是建议你可以在我的基础上整理一份属于的快捷键目录，本篇文章也只是起到一个工具和引子的作用。

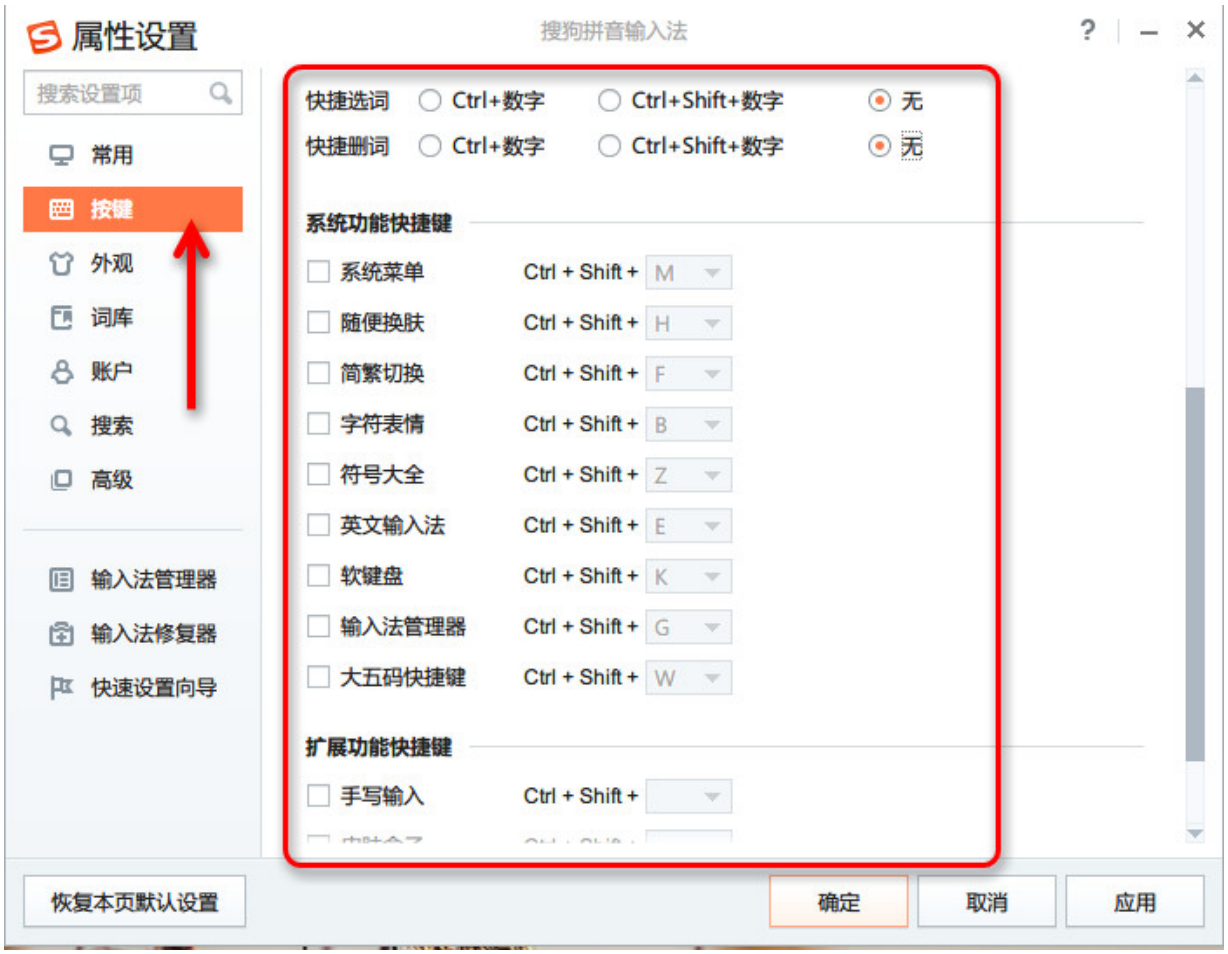
对于下面各个快捷键的使介绍描述也许用我个人语言翻译起来不够准确或是不全面，且在不同的文件类型上按出来的效果也可能结果不太一样,所以 **强烈建议** 你自己把各个快捷键都亲自操作下体会下各个快捷键的实际用法。

前提

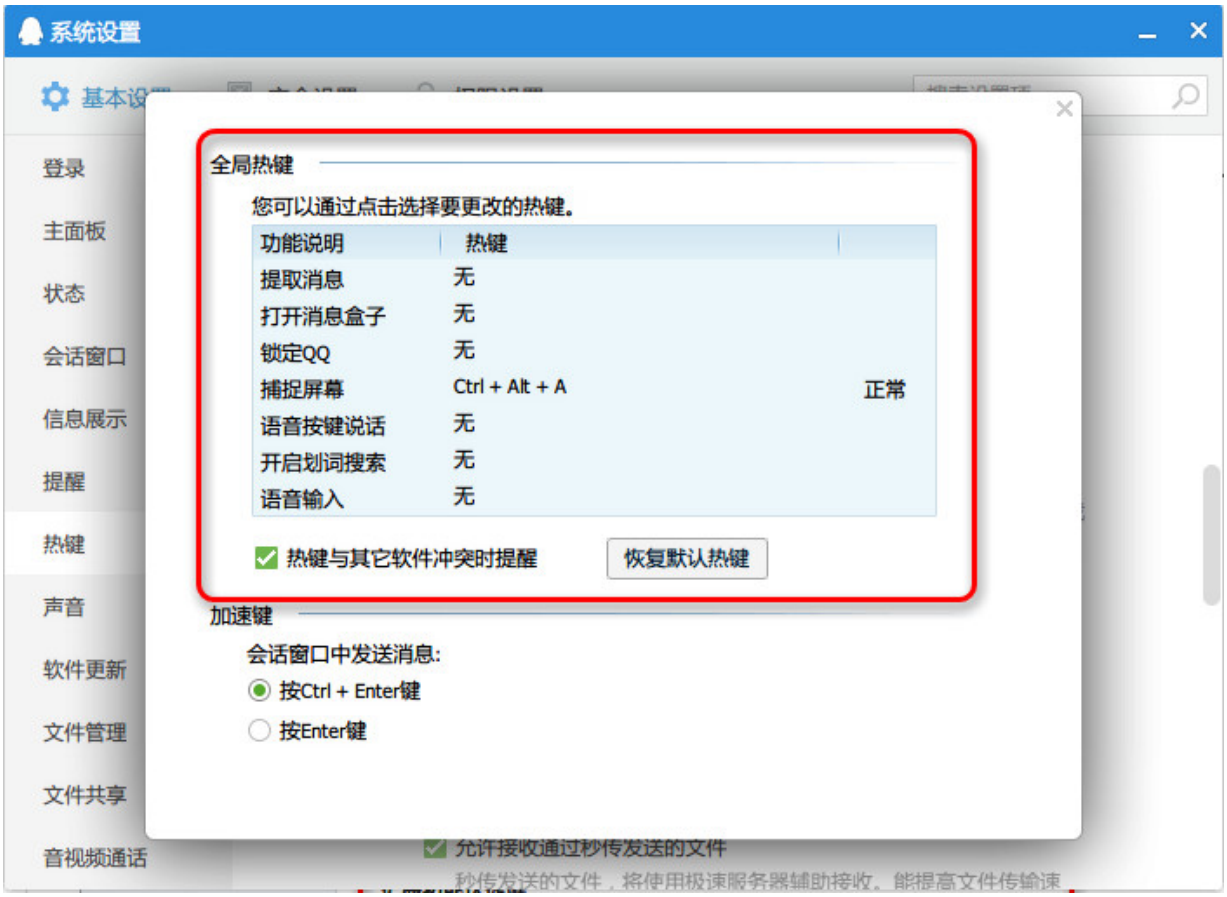
由于文化的不同，我们使用的电脑必备一个软件就是中文输入法，而目前大多数人都使用搜狗拼音输入法或是其他类似的。而这些输入法跟 IntelliJ IDEA 有一个万恶的冲突永恒不变：快捷键冲突。所以为了配合 IntelliJ IDEA，我们要去掉这些输入法下的所有快捷键。



- 如上图红色圈住内容所示，默认是 `逗号` 我改为了 ESC 键下的 `波浪号`，`Ctrl + 逗号` 这个快捷键适合做智能提示用，下面的快捷键列表会讲。



- 如上图红色圈住内容所示，这些快捷键很容易跟 IntelliJ IDEA 快捷键冲突，需要全部去掉。



- 如上图红色圈住内容所示，QQ 这些快捷键也很容易跟 IntelliJ IDEA 快捷键冲突，需要全部去掉，最多保持一个截图功能即可。

可能还有其他一些软件，比如网易云音乐、有道词典等等这些软件都可能存在快捷键冲突，所以为了 IntelliJ IDEA 这些软件的快捷键都是值得舍弃的，如果你在开发的时候。

Ctrl

快捷键	介绍
Ctrl + F	在当前文件进行文本查找 （必备）
Ctrl + R	在当前文件进行文本替换 （必备）
Ctrl + Z	撤销 （必备）
Ctrl + Y	删除光标所在行 或 删除选中的行 （必备）
Ctrl + X	剪切光标所在行 或 剪切选择内容
Ctrl + C	复制光标所在行 或 复制选择内容
Ctrl + D	复制光标所在行 或 复制选择内容，并把复制内容插入光标位置下面 （必备）
Ctrl + W	递进式选择代码块。可选中光标所在的单词或段落，连续按会在原有选中的基础上再扩展选中范围 （必备）
Ctrl + E	显示最近打开的文件记录列表
Ctrl + N	根据输入的 **类名** 查找类文件
Ctrl + G	在当前文件跳转到指定行处
Ctrl + J	插入自定义动态代码模板
Ctrl + P	方法参数提示显示
Ctrl + Q	光标所在的变量 / 类名 / 方法名等上面（也可以在提示补充的时候按），显示文档内容
Ctrl + U	前往当前光标所在的方法的父类的方法 / 接口定义
Ctrl + B	进入光标所在的方法/变量的接口或是定义出，等效于 `Ctrl + 左键单击`
Ctrl + K	版本控制提交项目，需要此项目有加入到版本控制才可用
Ctrl + T	版本控制更新项目，需要此项目有加入到版本控制才可用
Ctrl + H	显示当前类的层次结构

快捷键	介绍
Ctrl + O	选择可重写的方法
Ctrl + I	选择可继承的方法
Ctrl + +	展开代码
Ctrl + -	折叠代码
Ctrl + /	注释光标所在行代码，会根据当前不同文件类型使用不同的注释符号 （必备）
Ctrl + [	移动光标到当前所在代码的花括号开始位置
Ctrl +]	移动光标到当前所在代码的花括号结束位置
Ctrl + F1	在光标所在的错误代码出显示错误信息
Ctrl + F3	调转到所选中的词的下一个引用位置
Ctrl + F4	关闭当前编辑文件
Ctrl + F8	在 Debug 模式下，设置光标当前行为断点，如果当前已经是断点则去掉断点
Ctrl + F9	执行 Make Project 操作
Ctrl + F11	选中文件 / 文件夹，使用助记符设定 / 取消书签
Ctrl + F12	弹出当前文件结构层，可以在弹出的层上直接输入，进行筛选
Ctrl + Tab	编辑窗口切换，如果在切换的过程又加按上delete，则是关闭对应选中的窗口
Ctrl + Enter	智能分隔行
Ctrl + End	跳到文件尾
Ctrl + Home	跳到文件头
Ctrl + Space	基础代码补全，默认在 Windows 系统上被输入法占用，需要进行修改，建议修改为 `Ctrl + 逗号` （必备）
Ctrl + Delete	删除光标后面的单词或是中文句

快捷键	介绍
Ctrl + BackSpace	删除光标前面的单词或是中文句
Ctrl + 1,2,3...9	定位到对应数值的书签位置
Ctrl + 左键单击	在打开的文件标题上，弹出该文件路径
Ctrl + 光标定位	按 Ctrl 不要松开，会显示光标所在的类信息摘要
Ctrl + 左方向键	光标跳转到当前单词 / 中文句的左侧开头位置
Ctrl + 右方向键	光标跳转到当前单词 / 中文句的右侧开头位置
Ctrl + 前方向键	等效于鼠标滚轮向前效果
Ctrl + 后方向键	等效于鼠标滚轮向后效果

Alt

快捷键	介绍
Alt + `	显示版本控制常用操作菜单弹出层
Alt + Q	弹出一个提示，显示当前类的声明 / 上下文信息
Alt + F1	显示当前文件选择目标弹出层，弹出层中有很多目标可以进行选择
Alt + F2	对于前面页面，显示各类浏览器打开目标选择弹出层
Alt + F3	选中文本，逐个往下查找相同文本，并高亮显示
Alt + F7	查找光标所在的方法 / 变量 / 类被调用的地方
Alt + F8	在 Debug 的状态下，选中对象，弹出可输入计算表达式调试框，查看该输入内容的调试结果
Alt + Home	定位 / 显示到当前文件的 Navigation Bar
Alt + Enter	IntelliJ IDEA 根据光标所在问题，提供快速修复选择，光标放在的位置不同提示的结果也不同 （必备）
Alt + Insert	代码自动生成，如生成对象的 set / get 方法，构造函数，toString() 等
Alt + 左方向键	按左方向切换当前已打开的文件视图
Alt + 右方向键	按右方向切换当前已打开的文件视图
Alt + 前方向键	当前光标跳转到当前文件的前一个方法名位置
Alt + 后方向键	当前光标跳转到当前文件的后一个方法名位置
Alt + 1,2,3...9	显示对应数值的选项卡，其中 1 是 Project 用得最多

Shift

快捷键	介绍
Shift + F1	如果有外部文档可以连接外部文档
Shift + F2	跳转到上一个高亮错误 或 警告位置
Shift + F3	在查找模式下，查找匹配上一个
Shift + F4	对当前打开的文件，使用新Windows窗口打开，旧窗口保留
Shift + F6	对文件 / 文件夹 重命名
Shift + F7	在 Debug 模式下，智能步入。断点所在行上有多个方法调用，会弹出进入哪个方法
Shift + F8	在 Debug 模式下，跳出，表现出来的效果跟 F9 一样
Shift + F9	等效于点击工具栏的 Debug 按钮
Shift + F10	等效于点击工具栏的 Run 按钮
Shift + F11	弹出书签显示层
Shift + Tab	取消缩进
Shift + ESC	隐藏当前 或 最后一个激活的工具窗口
Shift + End	选中光标到当前行尾位置
Shift + Home	选中光标到当前行头位置
Shift + Enter	开始新一行。光标所在行下空出一行，光标定位到新行位置
Shift + 左键单击	在打开的文件名上按此快捷键，可以关闭当前打开文件
Shift + 滚轮前后滚动	当前文件的横向滚动轴滚动

Ctrl + Alt

快捷键	介绍
Ctrl + Alt + L	格式化代码，可以对当前文件和整个包目录使用 （必备）
Ctrl + Alt + O	优化导入的类，可以对当前文件和整个包目录使用 （必备）
Ctrl + Alt + I	光标所在行 或 选中部分进行自动代码缩进，有点类似格式化
Ctrl + Alt + T	对选中的代码弹出环绕选项弹出层
Ctrl + Alt + J	弹出模板选择窗口，讲选定的代码加入动态模板中
Ctrl + Alt + H	调用层次
Ctrl + Alt + B	在某个调用的方法名上使用会跳到具体的实现处，可以跳过接口
Ctrl + Alt + V	快速引进变量
Ctrl + Alt + Y	同步、刷新
Ctrl + Alt + S	打开 IntelliJ IDEA 系统设置
Ctrl + Alt + F7	显示使用的地方。寻找被该类或是变量被调用的地方，用弹出框的方式找出来
Ctrl + Alt + F11	切换全屏模式
Ctrl + Alt + Enter	光标所在行上空出一行，光标定位到新行
Ctrl + Alt + Home	弹出跟当前文件有关联的文件弹出层
Ctrl + Alt + Space	类名自动完成
Ctrl + Alt + 左方向键	退回到上一个操作的地方 （必备）

快捷键	介绍
Ctrl + Alt + 右方向键	前进到上一个操作的地方 （必备）
Ctrl + Alt + 前方向键	在查找模式下，跳到上个查找的文件
Ctrl + Alt + 后方向键	在查找模式下，跳到下个查找的文件

Ctrl + Shift

快捷键	介绍
Ctrl + Shift + F	根据输入内容查找整个项目 或 指定目录内文件 （必备）
Ctrl + Shift + R	根据输入内容替换对应内容，范围为整个项目 或 指定目录内文件 （必备）
Ctrl + Shift + J	自动将下一行合并到当前行末尾 （必备）
Ctrl + Shift + Z	取消撤销 （必备）
Ctrl + Shift + W	递进式取消选择代码块。可选中光标所在的单词或段落，连续按会在原有选中的基础上再扩展取消选中范围 （必备）
Ctrl + Shift + N	通过文件名定位 / 打开文件 / 目录，打开目录需要在输入的内容后面多加一个正斜杠 （必备）
Ctrl + Shift + U	对选中的代码进行大 / 小写轮流转换 （必备）
Ctrl + Shift + T	对当前类生成单元测试类，如果已经存在的单元测试类则可以进行选择
Ctrl + Shift + C	复制当前文件磁盘路径到剪贴板
Ctrl + Shift + V	弹出缓存的最近拷贝的内容管理器弹出层
Ctrl + Shift + E	显示最近修改的文件列表的弹出层
Ctrl + Shift + H	显示方法层次结构
Ctrl + Shift + B	跳转到类型声明处

快捷键	介绍
Ctrl + Shift + I	快速查看光标所在的方法 或 类的定义
Ctrl + Shift + A	查找动作 / 设置
Ctrl + Shift + /	代码块注释 （必备）
Ctrl + Shift + [	选中从光标所在位置到它的顶部中括号位置
Ctrl + Shift +]	选中从光标所在位置到它的底部中括号位置
Ctrl + Shift + +	展开所有代码
Ctrl + Shift + -	折叠所有代码
Ctrl + Shift + F7	高亮显示所有该选中文本，按Esc高亮消失
Ctrl + Shift + F8	在 Debug 模式下，指定断点进入条件
Ctrl + Shift + F9	编译选中的文件 / 包 / Module
Ctrl + Shift + F12	编辑器最大化
Ctrl + Shift + Space	智能代码提示
Ctrl + Shift + Enter	自动结束代码，行末自动添加分号 （必备）
Ctrl + Shift + Backspace	退回到上次修改的地方
Ctrl + Shift + 1,2,3...9	快速添加指定数值的书签

快捷键	介绍
Ctrl + Shift + 左方向键	在代码文件上，光标跳转到当前单词 / 中文句的左侧开头位置，同时选中该单词 / 中文句
Ctrl + Shift + 右方向键	在代码文件上，光标跳转到当前单词 / 中文句的右侧开头位置，同时选中该单词 / 中文句
Ctrl + Shift + 左方向键	在光标焦点是在工具选项卡上，缩小选项卡区域
Ctrl + Shift + 右方向键	在光标焦点是在工具选项卡上，扩大选项卡区域
Ctrl + Shift + 前方向键	光标放在方法名上，将方法移动到上一个方法前面，调整方法排序
Ctrl + Shift + 后方向键	光标放在方法名上，将方法移动到下一个方法前面，调整方法排序

Alt + Shift

快捷键	介绍
Alt + Shift + N	选择 / 添加 task
Alt + Shift + F	显示添加到收藏夹弹出层
Alt + Shift + C	查看最近操作项目的变化情况列表
Alt + Shift + F	添加到收藏夹
Alt + Shift + I	查看项目当前文件
Alt + Shift + F7	在 Debug 模式下，下一步，进入当前方法体内，如果方法体还有方法，则会进入该内嵌的方法中，依此循环进入
Alt + Shift + F9	弹出 Debug 的可选择菜单
Alt + Shift + F10	弹出 Run 的可选择菜单
Alt + Shift + 左键双击	选择被双击的单词 / 中文句，按住不放，可以同时选择其他单词 / 中文句
Alt + Shift + 前方向键	移动光标所在行向上移动
Alt + Shift + 后方向键	移动光标所在行向下移动

Ctrl + Shift + Alt

快捷键	介绍
Ctrl + Shift + Alt + V	无格式黏贴
Ctrl + Shift + Alt + N	前往指定的变量 / 方法
Ctrl + Shift + Alt + S	打开当前项目设置
Ctrl + Shift + Alt + C	复制参考信息

其他

快捷键	介绍
F2	跳转到下一个高亮错误 或 警告位置 （必备）
F3	在查找模式下，定位到下一个匹配处
F4	编辑源
F7	在 Debug 模式下，进入下一步，如果当前行断点是一个方法，则进入当前方法体内，如果该方法体还有方法，则不会进入该内嵌的方法中
F8	在 Debug 模式下，进入下一步，如果当前行断点是一个方法，则不进入当前方法体内
F9	在 Debug 模式下，恢复程序运行，但是如果该断点下面代码还有断点则停在下一个断点上
F11	添加书签
F12	回到前一个工具窗口
Tab	缩进
ESC	从工具窗口进入代码文件窗口
连按两次 Shift	弹出 `Search Everywhere` 弹出层

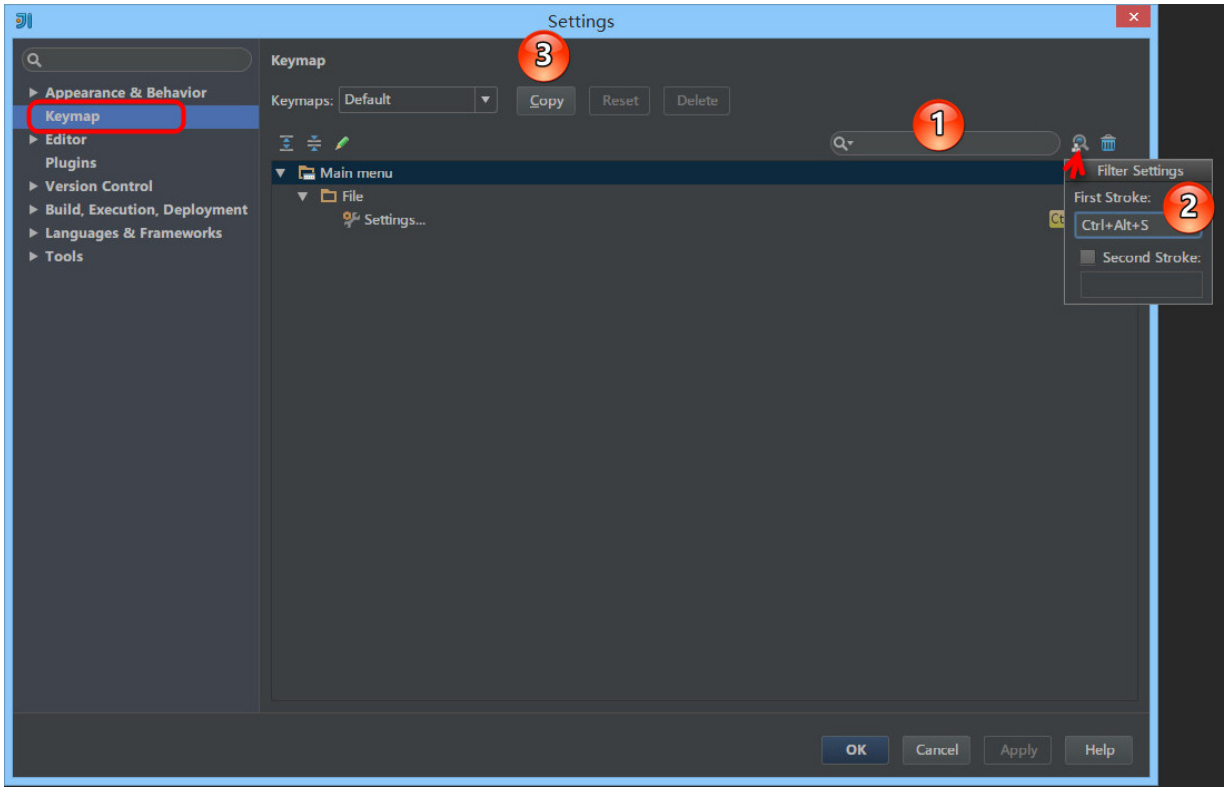
官网快捷键资料

- Windows / Linux：https://www.jetbrains.com/idea/docs/IntelliJIDEA_ReferenceCard.pdf
- Mac OS X：https://www.jetbrains.com/idea/docs/IntelliJIDEA_ReferenceCard_Mac.pdf

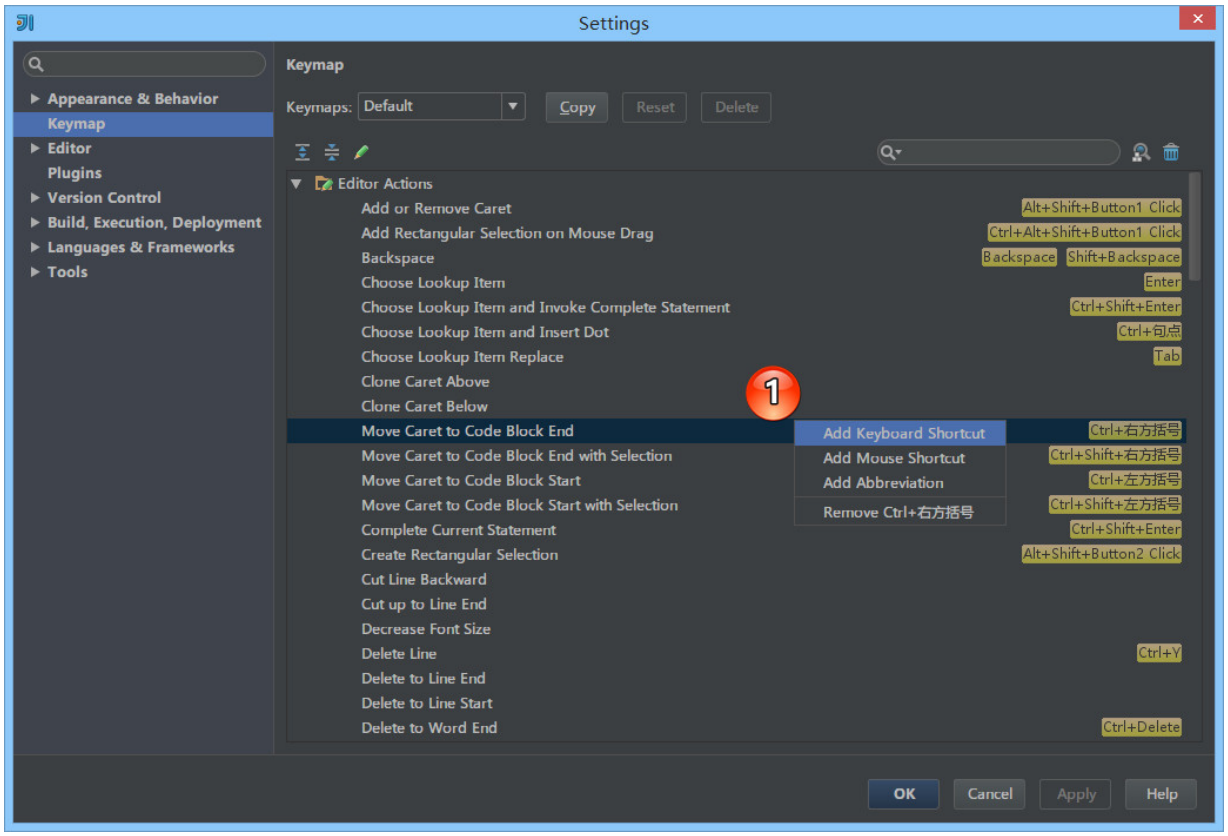
第三方快捷键资料

- 来自 eta02913：
  http://xinyuwu.iteye.com/blog/1005454

快捷键修改



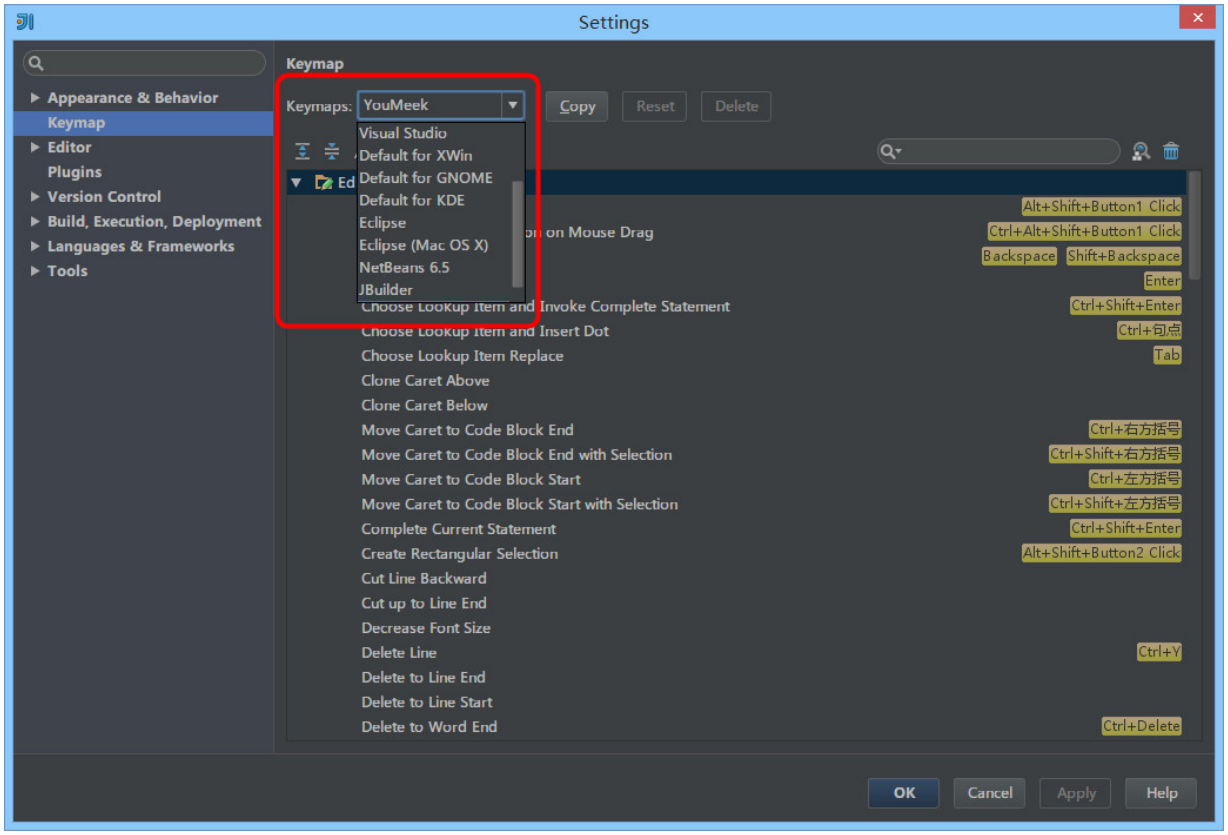
- 按 `Ctrl + Alt + S` 弹出 IDE 设置，如上图选择左侧的 `Keymap`。
- IntelliJ IDEA 支持两种方式来筛选我们要找的快捷键，一种是上图标注 1 所示的，通过输入快捷键描述内容；一种是上图标注 2 所示的，通过 **按** 指定快捷键快捷键，这里需要再次强调的是，这个输入框是自动监听你当前按下的按键，而不是用来输入的。
- 上图标注 3 所示，初安装的 IntelliJ IDEA 使用的是 `Default` 的快捷键模板，IntelliJ IDEA 默认的快捷键模板都是不可修改的。如果你直接修改，当前这个位置 IntelliJ IDEA 会自动变成 `Default Copy`，建议你养成习惯，修改之前先点击 `Copy`，拷贝一套快捷键模板，然后输入自己的命名。



- IntelliJ IDEA 是支持一个操作命令同时设置多个快捷键组合，就如上图的 `Backspace`，同时支持 `Backspace` 和 `Shift + Backspace` 两组快捷键。
- 要修改某个快捷键，选中快捷键介绍内容，右键，就会弹出如上图标注 1 所示操作选择。
- 命令 `Add Keyboard Shortcut` 用来添加新纯键盘快捷键组合。
- 命令 `Add Mouse Shortcut` 用来添加新 `键盘 + 鼠标` 快捷键组合，比如设置 `Ctrl + 左键单击` 这类快捷组合。其中在弹出的添加面板中 `Click Pad` 是用来监听当前鼠标是左键单击还是右键单击。
- 命令 `Add Abbreviation` 根据 IntelliJ IDEA 的版本文档解释，添加简称主要是为了方便 `Search Everywhere` 中使用，

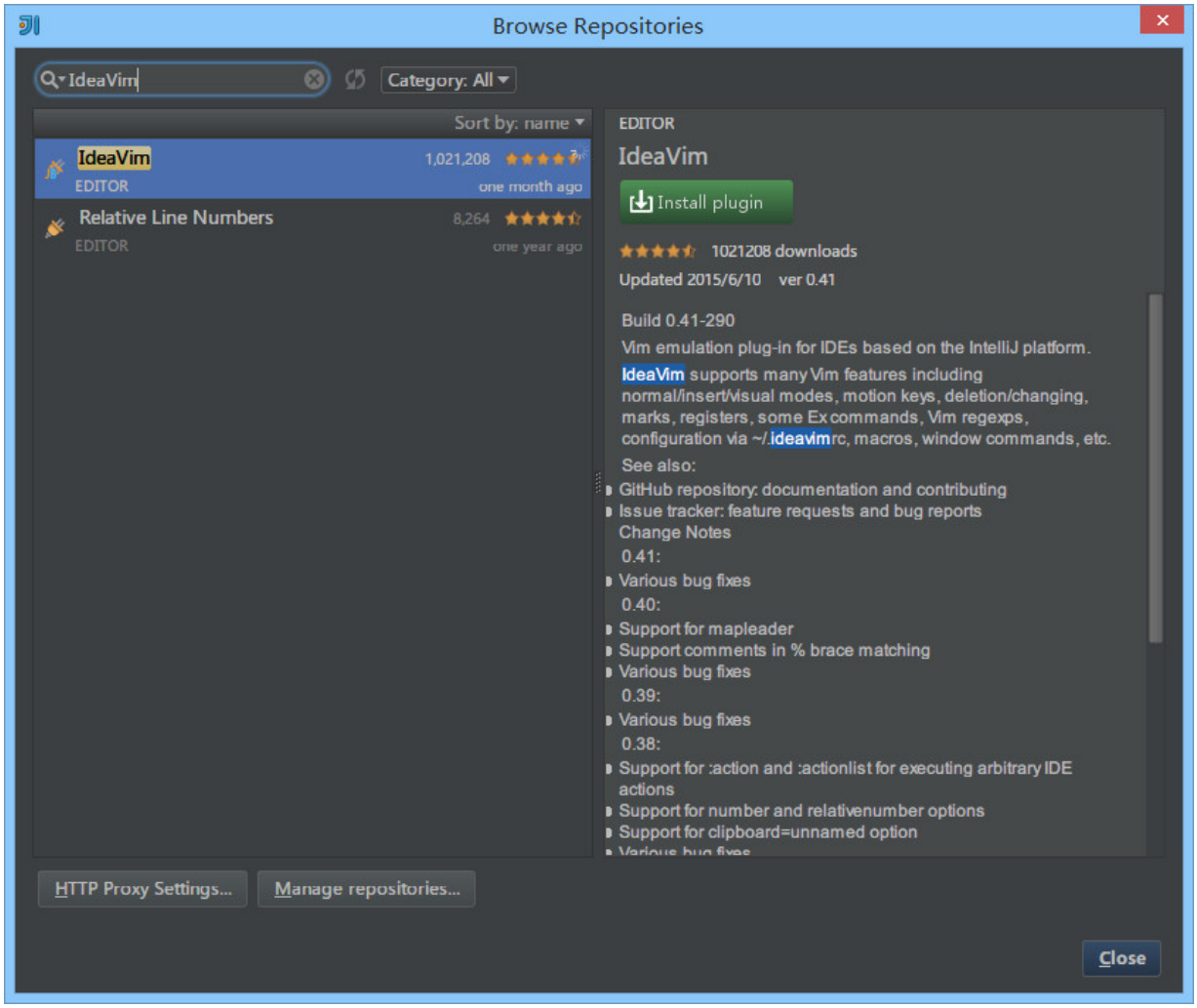
但是我尝试之后发现没办法根据我设置的简称搜索，暂时无法了解其作用。

- 命令 `Remove 快捷键` 移出当前操作命令已设置的快捷键组合，由于 IntelliJ IDEA 默认就占用了很多快捷键组合，所以如果你要修改某个快捷键，建议还是删除掉旧的。



- IntelliJ IDEA 对其他 IDE 用户很友好，比如如上图对于其他主流的 IDE，快捷键上已经默认了有其过度快捷键模板了，但是我还是建议你专心使用 IntelliJ IDEA 的默认。

其他



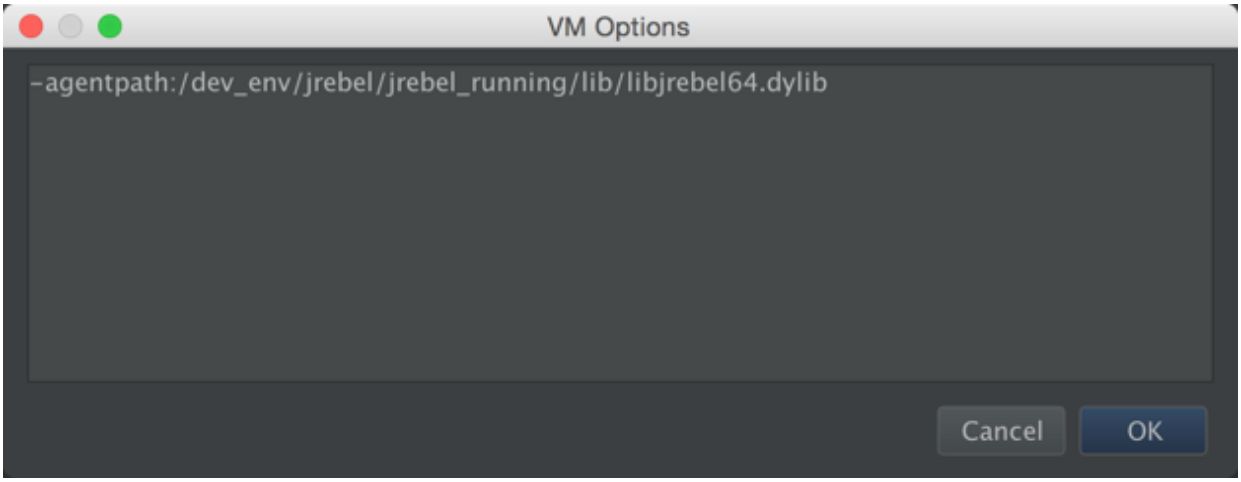
- 如果你是一个 Vim 粉，IntelliJ IDEA 也为你准备了一个方案：如上图安装 IdeaVim 插件即可。

<code>-noverify

-agentpath:D:/dev_env/jrebel/jrebel_running/lib/jrebel64.dll
</code>

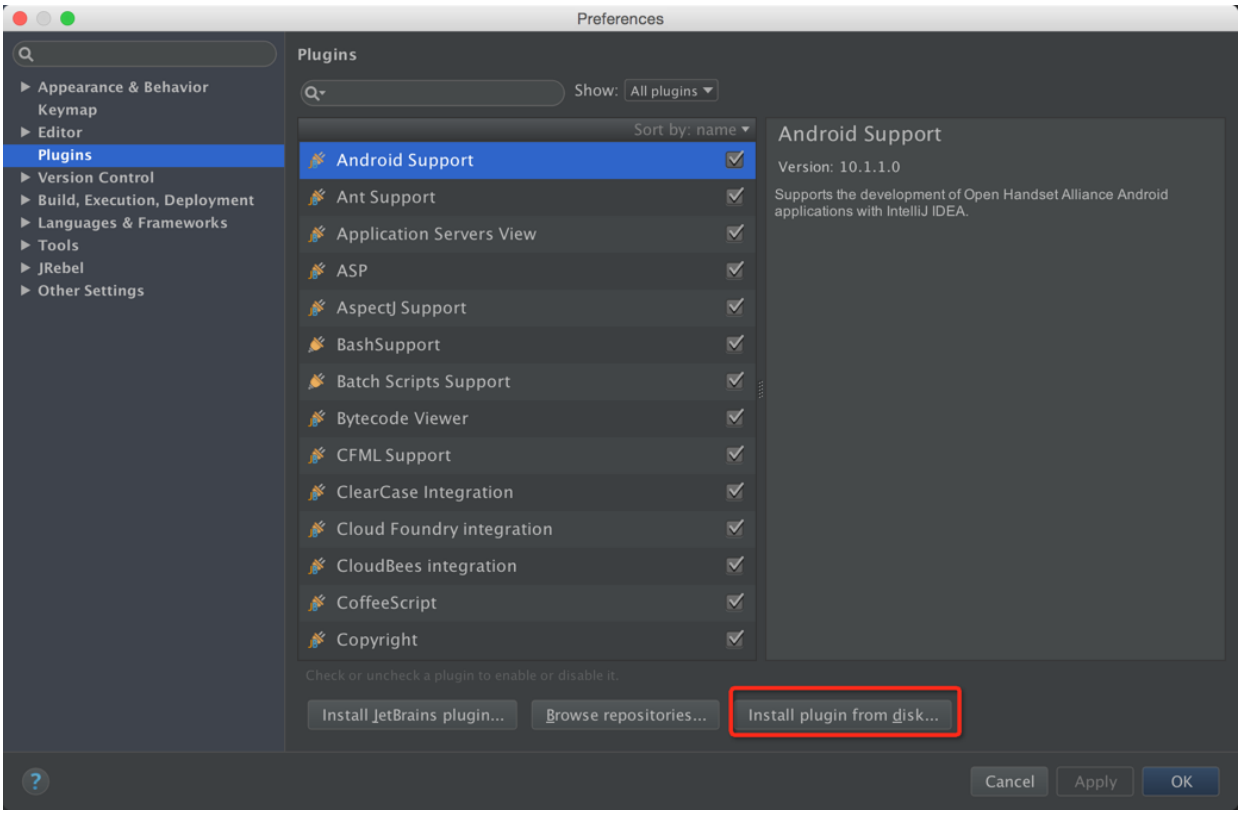
<code>-
agentpath:/dev_env/jrebel/jrebel_running/lib/libjrebel64.so
</code>

<code>-
agentpath:/dev_env/jrebel/jrebel_running/lib/libjrebel64.dylib
</code>



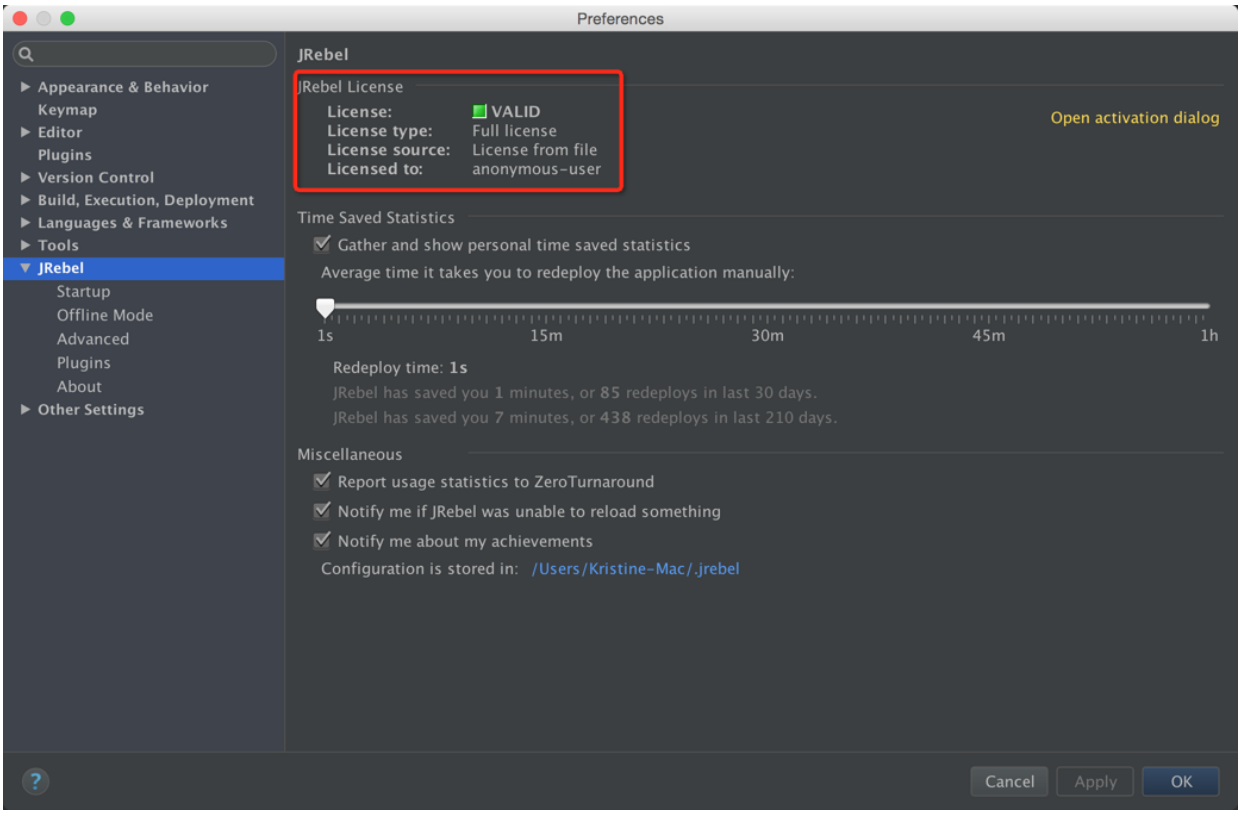
配置完成， 直接启动 Tomcat 即可， 不过此方法麻烦， 每次新建项目都要从新配置

接下来介绍使用 IntelliJ IDEA 插件的方式启动 JRebel 首先是安装 JRebel 的插件， 安装方法和其他插件安装方法一样， 不过这里不采用在线安装， 直接选择本地安装， 直接选择插件安装即可



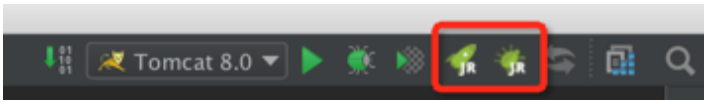
安装好后在设置里面会多出一项JRebel的配置

查看一下插件是否有效

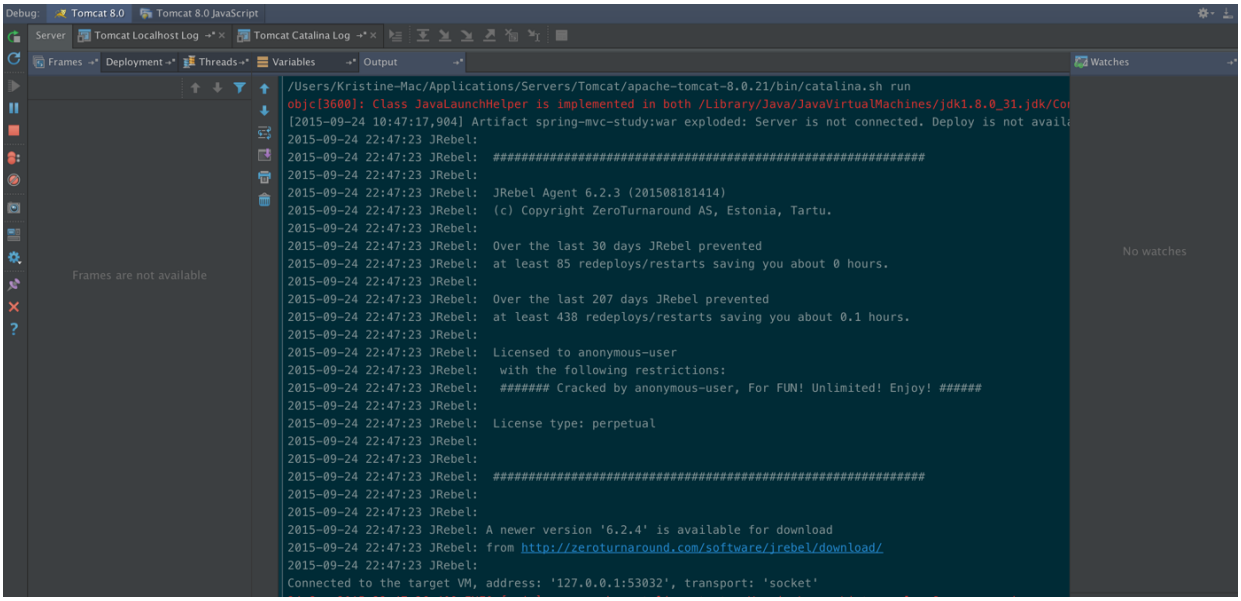


绿色的 VALID 表示是有效的

在原来运行项目的按钮边上会多出两个绿色的按钮， 如图， 前面那个是 Run， 后面那个是 Debug



配置 Tomcat 的方法和直接上面说的直接调用配置方法一样， 同样需要注意的是 On 'Update' action 和 On frame deactivation 这两项目一定要选择 Update classes and resources， 唯一不同的是 VM options 这项不需要填， 放空就好
接下来直接启动项目， 一般选择后面那个 Debug 按钮



看到 Log 有 JRebel 输出的版本信息， 没有报错就是表示成功执行了， 随便改一个类试试吧

JRebel 官网有免费激活服务，到官网注册领取，请支持正版

本系列教程结束

赠语

只希望本系列教程只是起到一个让你真正了解 IntelliJ IDEA 的一个引子作用，没有什么比 IntelliJ IDEA 自己提供的帮助文档更有意义和说服力。希望以后的学习中，你能多看看 IntelliJ IDEA 自带的帮助文档和关注官网动态。

最后，感谢 JetBrains 公司做了如此优秀的产品，感谢你和我对 IntelliJ IDEA 抱有兴趣，感谢极客学院对本系列教程的支持！

如果你发现本系列教程有错误或是表达不清的地方，请邮件联系：judas.n@qq.com，真心地万分感谢（鞠躬）。