# Intellij IDEA

## IDEA的特点（相较Eclips而言）

1.强大的整合能力。比如：Git、Maven、Spring 等

2.提示功能的快速、便捷、范围广

3.好用的快捷键和代码模板

4.精准搜索

5.实时保存

6.对硬件要求高

## IDEA安装目录

bin： 容器，执行文件和启动参数等

help： 快捷键文档和其他帮助文档

jre64： 64位java 运行环境

lib： idea依赖的类库

license： 各个插件许可

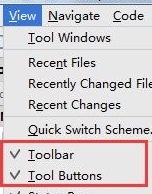
plugin： 插件

/bin/ idea64.exe.vmoptions: 64位idea的VM配置文件

C:\Users\Administrator\.IntelliJIdea2017.2： idea设置目录，有idea的配置目录config和系统目录system，没有首次启动不会存在，如果想还原idea的默认配置，可以删除该目录 来实现，其就相当于Eclipse的.metadata

IDEA插件官方下载地址：<https://plugins.jetbrains.com/idea>

## 界面



工程下的 src类似Eclipse下的src目录，

用于存放代码。

.idea和.iml文件都是IDEA特有的。

类似于Eclipse工程下的.settings、

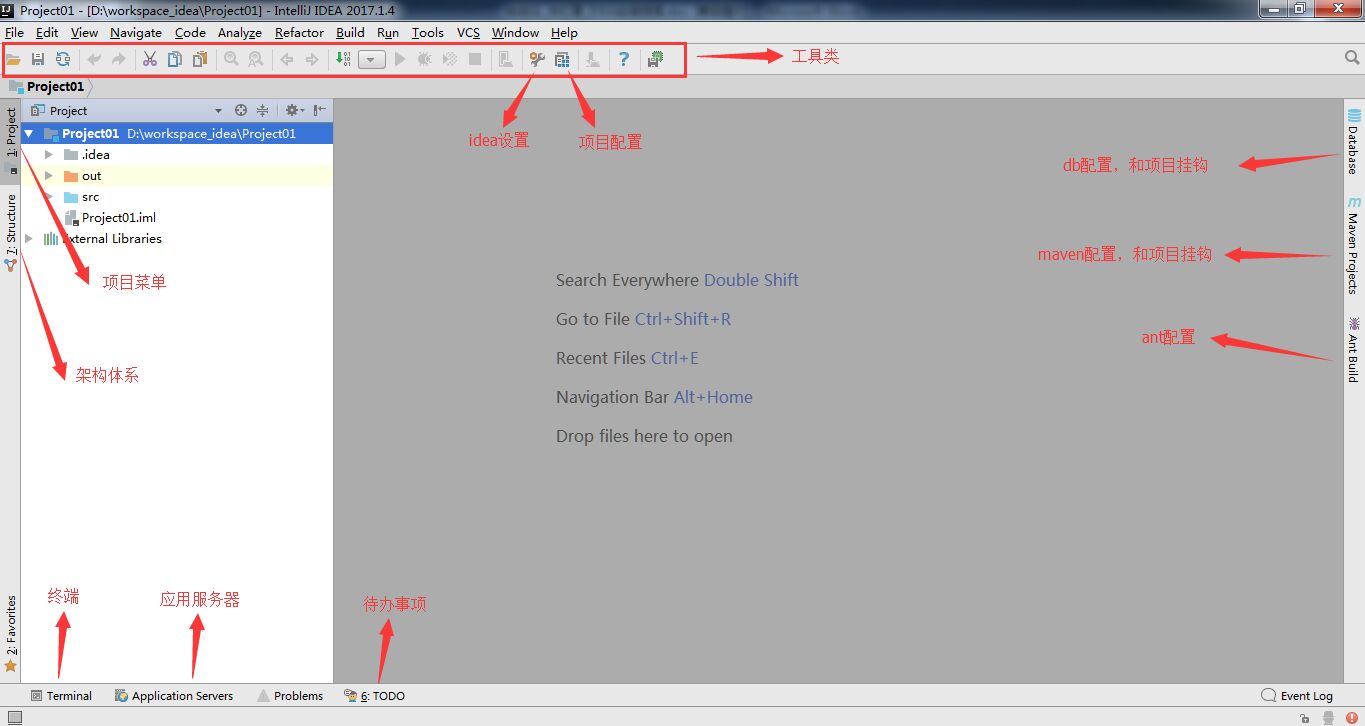
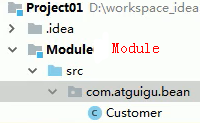
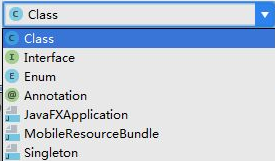
.classpath、.project等。

.iml是依赖文件，使用另一个模板中的

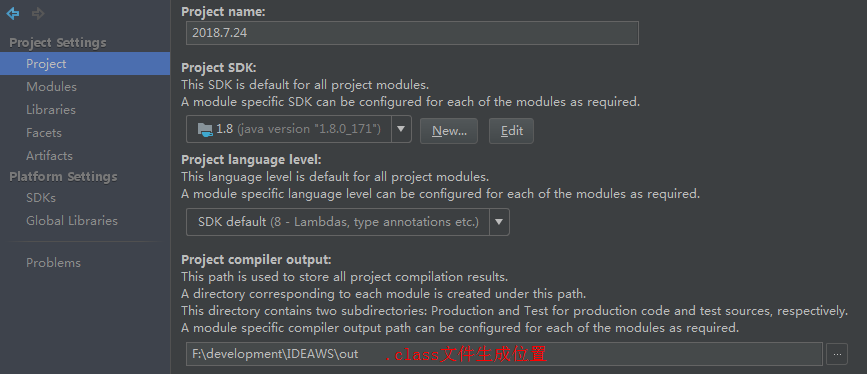
类时就需要添加内容

Project相当于Eclipse中的workspace

Module相当于Eclipse中的Project

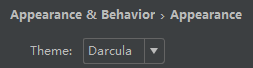
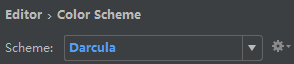
移除Module：Module右键 → Open Module Setting →  → Module右键 → Remove Module（此时硬盘中也不存在了）



## 常用配置



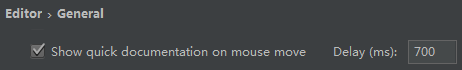
#### 1.主题

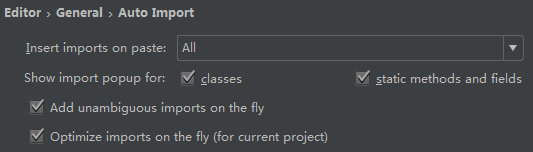
#### 2.鼠标滚轮控制字体大小



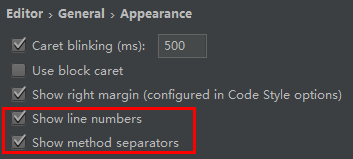
#### 3.文档说明



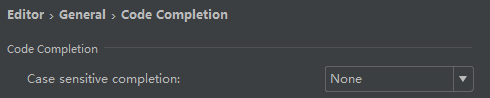
#### 4.自动导包



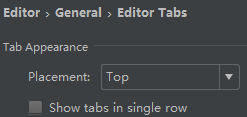
#### 5.行号和方法分隔符



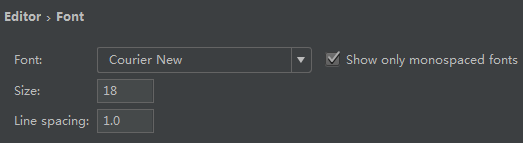
#### 6.提示忽略大小写



#### 7.多行显示修改文件



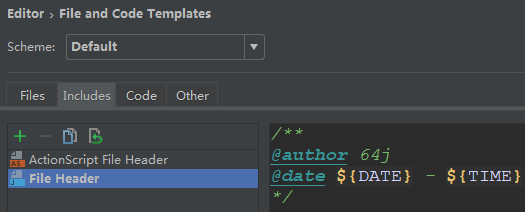
#### 8.字体样式、大小、行间距

可能需要重启IDEA来生效

#### 9.注释、方法、变量等颜色



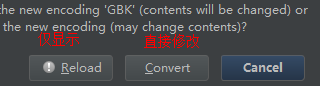
#### 10.新建文件自动添加头部信息



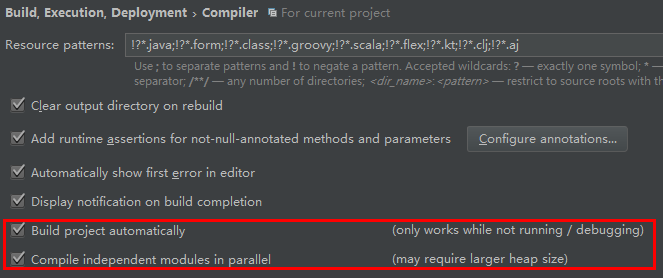
#### 11.项目文件编码



#### 12.当前文件编码修改

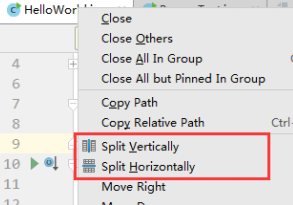
#### 13.自动编译



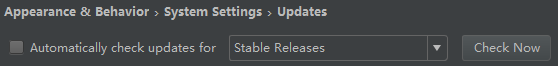
#### 14.省电模式（为了防止进入该模式而提出）

File → Power Save Mode

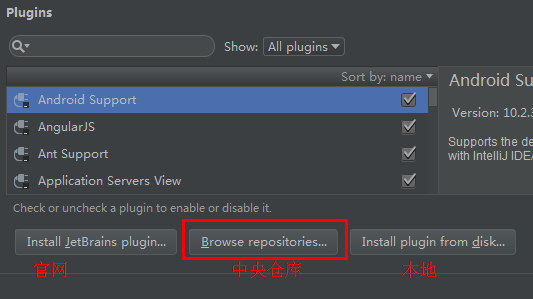
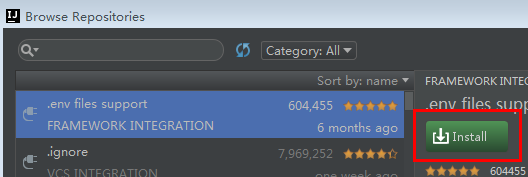
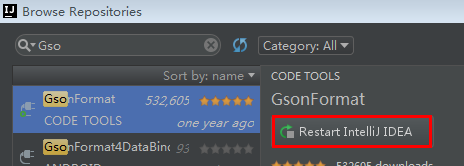
15.垂直显示和水平显示



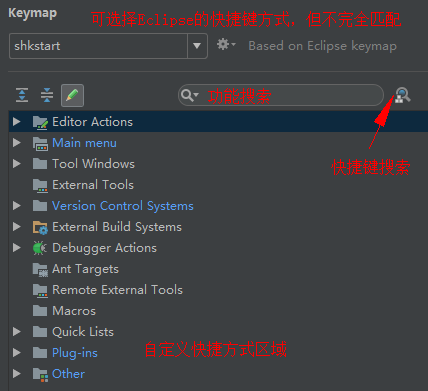
#### 16.取消软件更新



#### 17.插件

## 快捷键

 亦可导入自定义快捷方式，然后在Keymap中选择（File → Import Settings）

该版本Eclipse快捷键介绍（只介绍目前自己不常用但很有用的）

run alt + r

开始下一行 shift + enter

开始上一行 ctrl + shift + enter

解错/提示/优化/生成返回值变量 alt + enter

回退编辑的页面 alt + left

反回退编辑的页面 alt + right

查看继承关系 F4

方法参数类型提示 ctrl + alt + /

反撤销 ctrl + y

查看类的结构（outline） ctrl + o

重构 (rename) alt + shift + r

大小写切换 ctrl + shift + y

立即显示文档说明 F2

收起所有的方法 alt + shift + c

打开所有方法 alt + shift + x

从磁盘打开文件 ctrl + shift + x

抽取为成员变量(introduce field) alt + shift + f

全局查找 ctrl + h

查找文件 double Shift

查看类的继承结构图 ctrl + shift + u

添加到收藏 ctrl + alt + f

抽取方法 alt + shift + m

打开最近修改的文件 ctrl + e

关闭当前打开的代码栏 ctrl + w

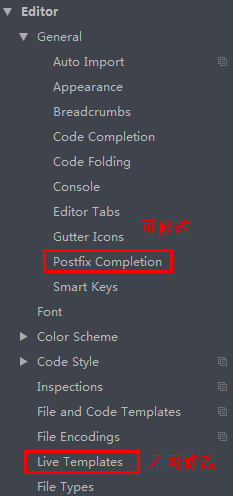
关闭打开的所有代码栏 ctrl + shift + w

快速找错 ctrl + shift + q

选择粘贴 ctrl + shift + v

查找方法在哪里被调用 ctrl + shift + h

## 模板



### 1.常用模板

1. sout 无

soutp 方法形参

soutm 方法名

soutv 变量

xxx.sout 指定

2.psvm（main）

3.fori/itar 普通

iter 增强for

list.for 集合增强for

list.fori 集合普通

list.forr 集合逆序普通

4.ifn

inn

xxx.null

xxx.nn

5.prsf

psf

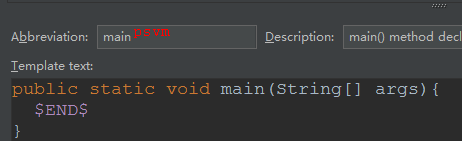
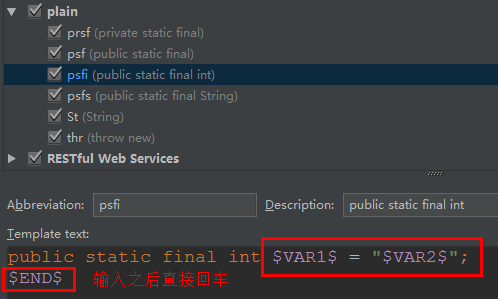
psfi

psfs

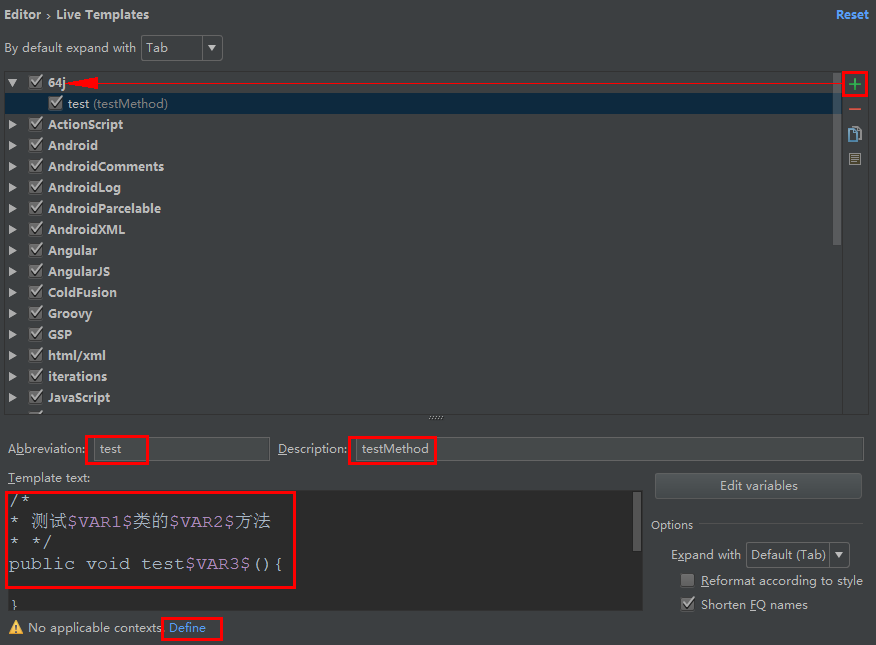
6.thr

xxx.try

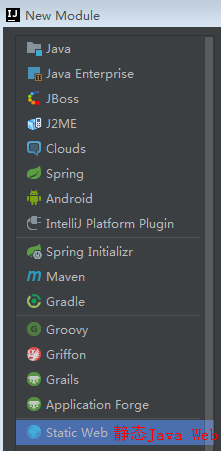
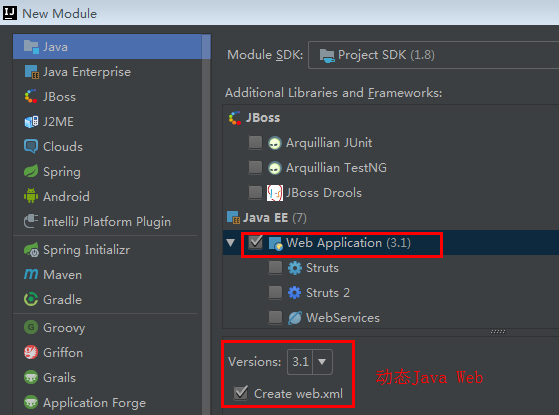
### 2.修改模板

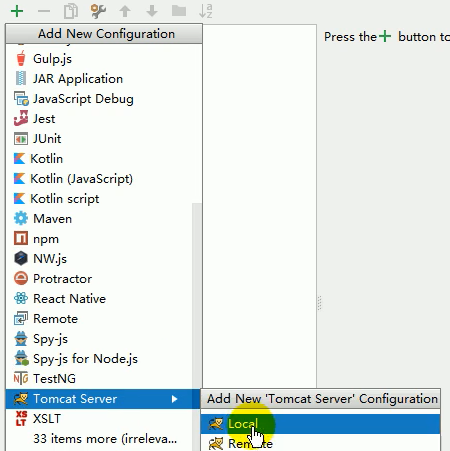
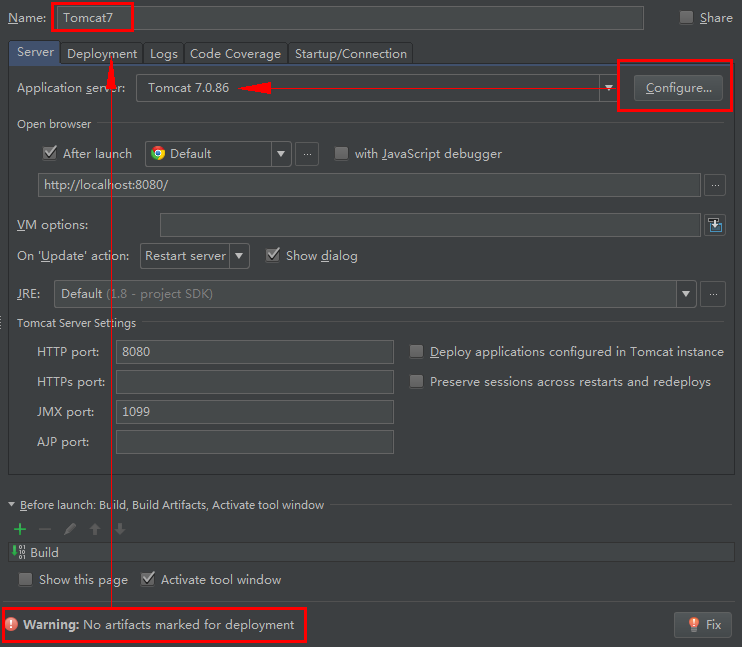
 

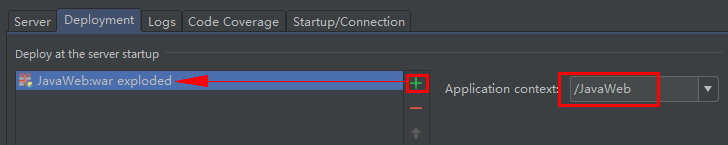
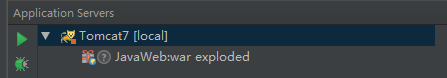
### 3.自定义模板



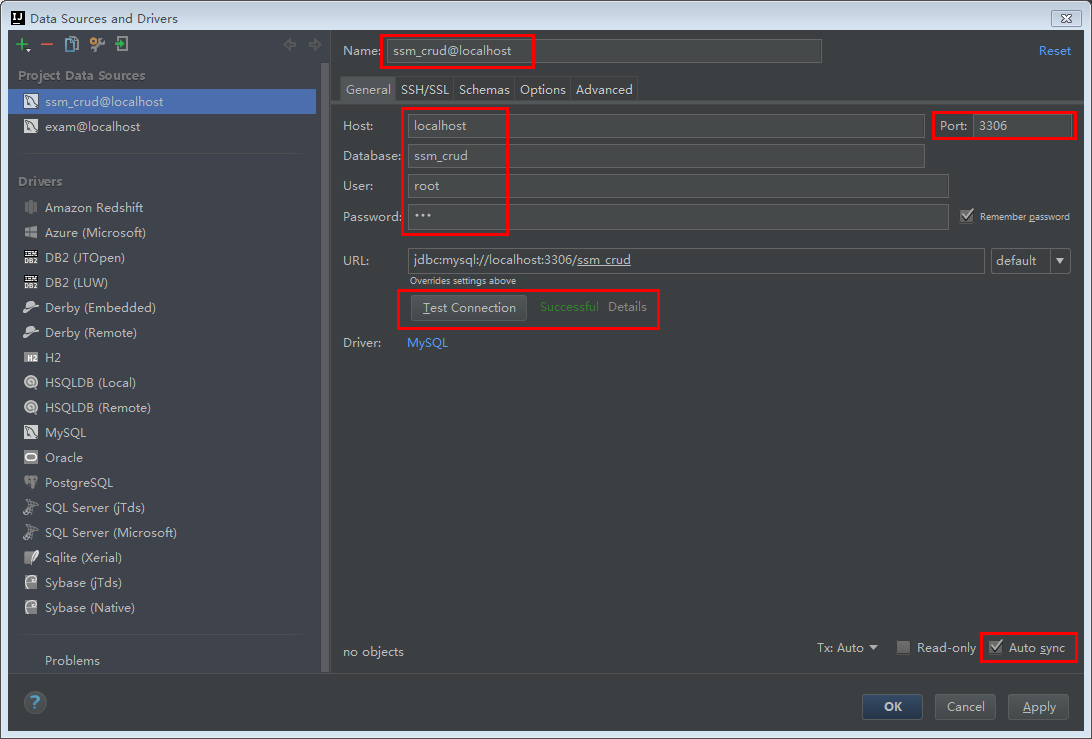
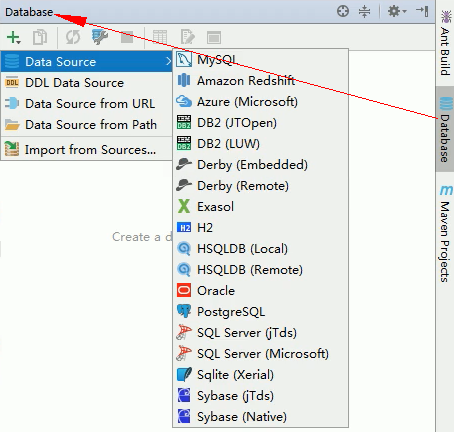
## 配置服务器

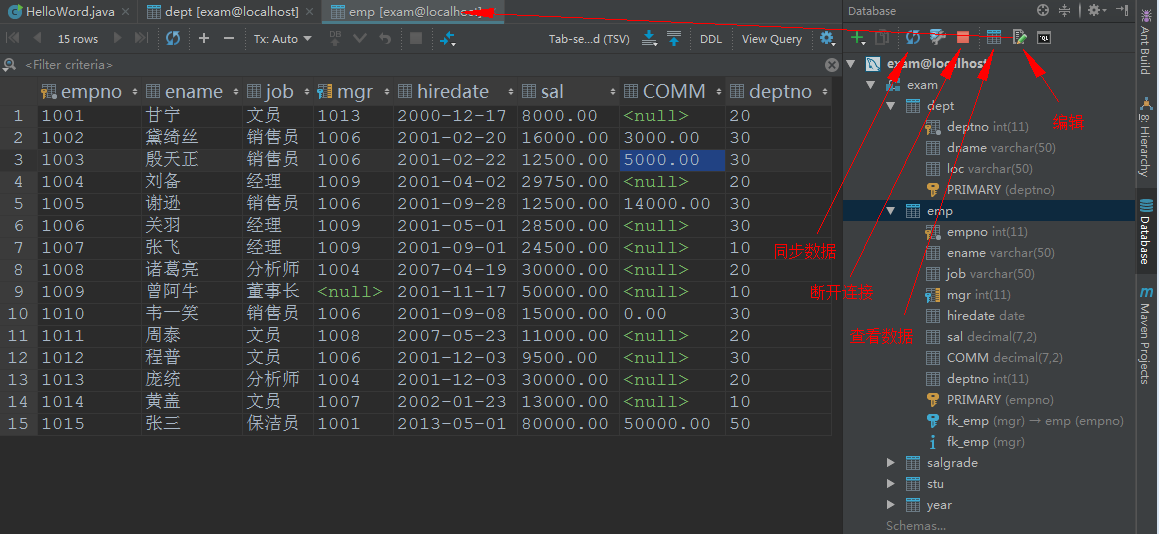
 

Run → Edit Configurations →  → →

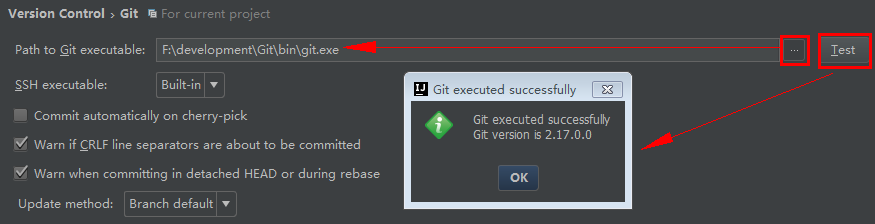
→  → 

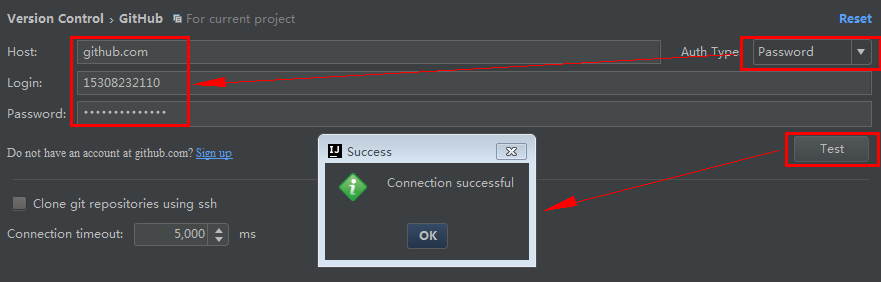
## 关联数据库

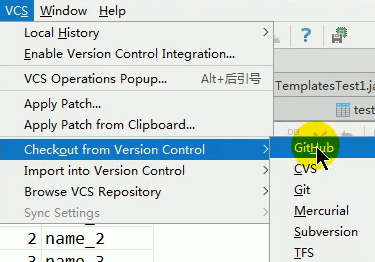


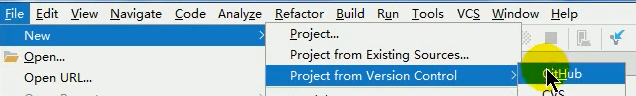


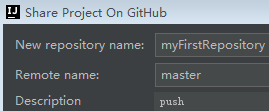
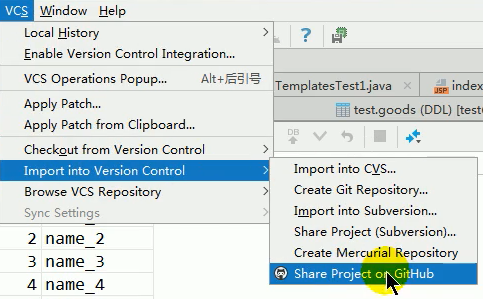
## 版本控制

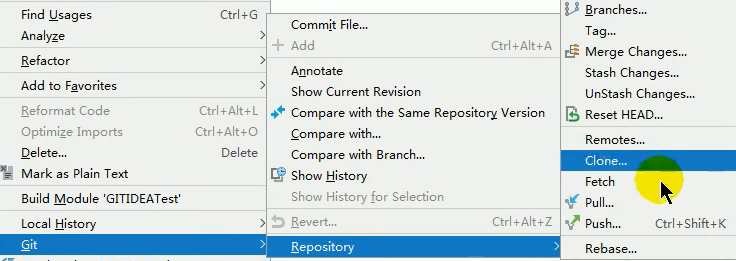




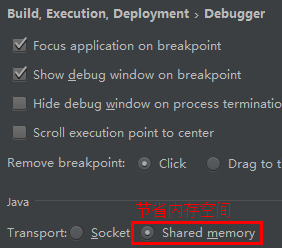
 







## 断点调试



step over  ： 下一步，不进入方法体内

step into ： 下一步，进入方法

force step into ： 下一步，进入方法

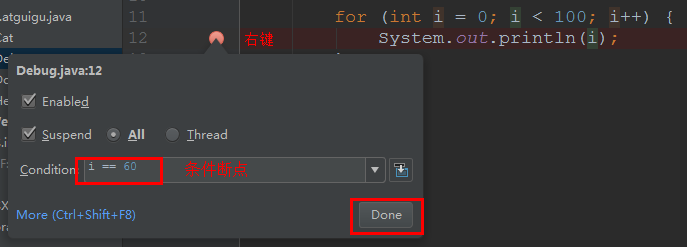
step out ： 跳出方法

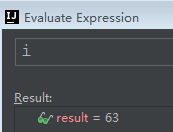
resume program ： 直接跳到下一个断点

stop ： 停止

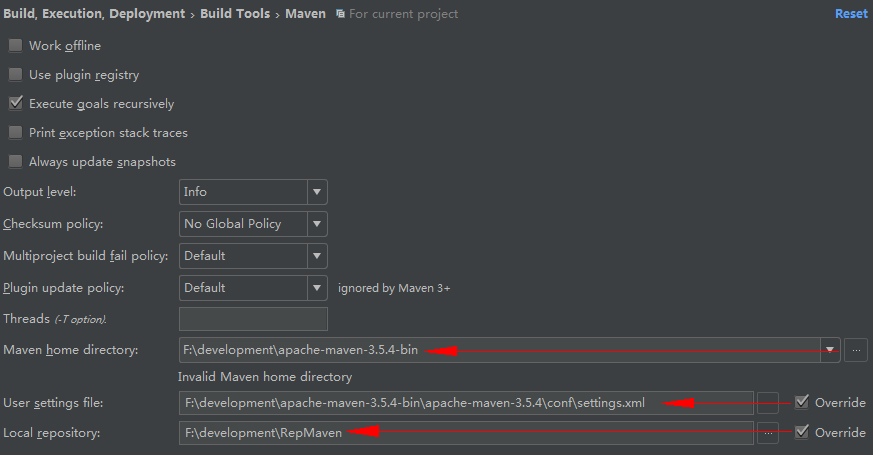
mute breakpoints ： 使断点失效

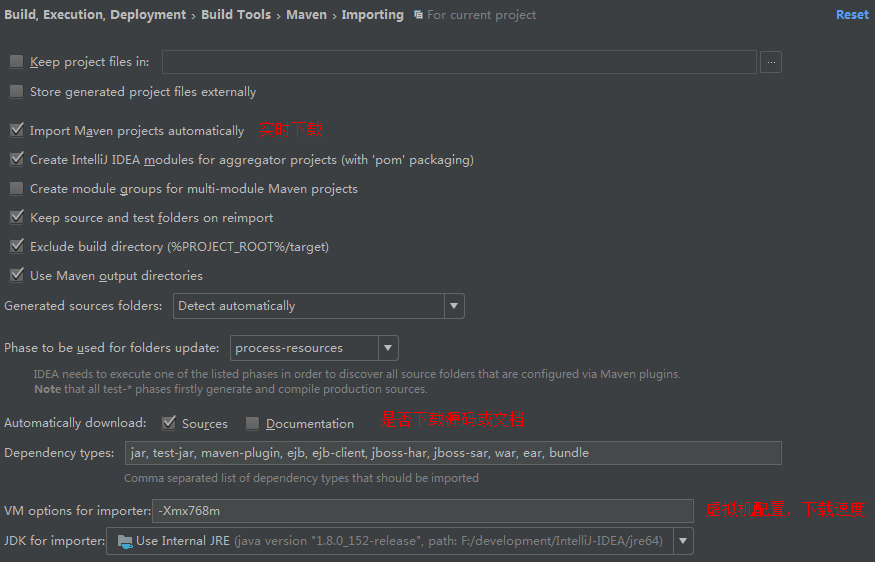
view breakpoints ： 查看断点个数

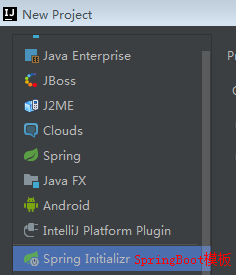
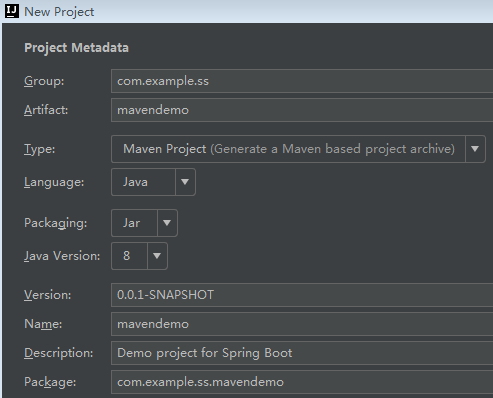
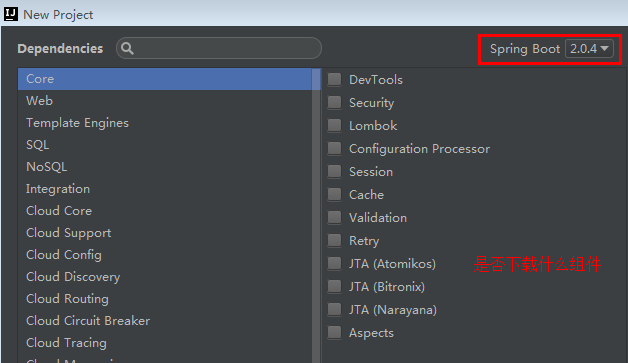


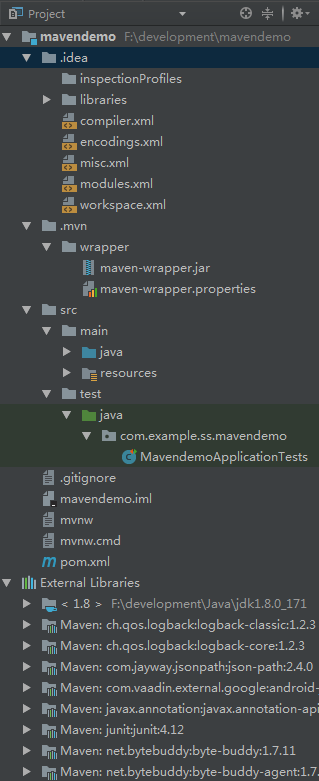
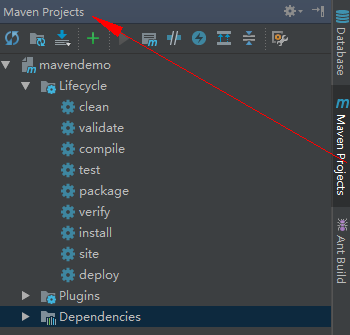
ctrl + u 查看此时已执行结果 

## Maven



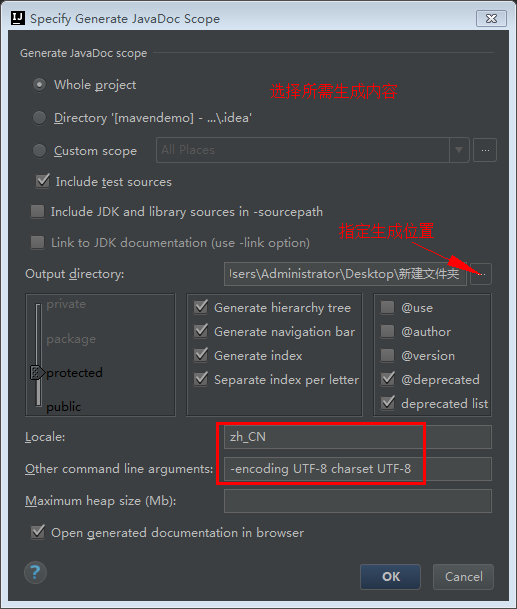


## 其他

#### 1.生成JavaDoc

Tools → Generate JavaDoc → 

#### 2.清理缓存和索引

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。

这时应该等 IntelliJ IDEA 创建索引完成，因为在这个过程中编辑的代码无法编译和运行。

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。

但是，IntelliJ IDEA 的索引和缓存并不是一直会良好地支持IntelliJ IDEA的，某些特殊条件下，IntelliJ IDEA 的缓存和索引文件也是会损坏的。

当你重新打开IntelliJ IDEA，很可能IntelliJ IDEA会报各种莫名的错误，甚至打不开项目，IntelliJ IDEA主题还原成默认状态等。

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缓存和索引出现了问题。遇到此类问题就需要清理缓存和索引。

File → Invalidata Caches/Restart → Invalidata and Restart

这会清除本地历史记录，所以如果需要可以进行备份：C:\Users\Administrator\.IntelliJIdea2017.2\system\LocalHistory

也可以通过删除system目录的方式来解决：C:\Users\Administrator\.IntelliJIdea2017.2\system