

Java环境变量

环境变量类似代码中的变量，需要定义，之后可以使用。

Java的环境变量一定要配置好，很多第三方开发工具都会依赖这个环境变量，如tomcat。需要配置的内容如下：

Windows环境配置

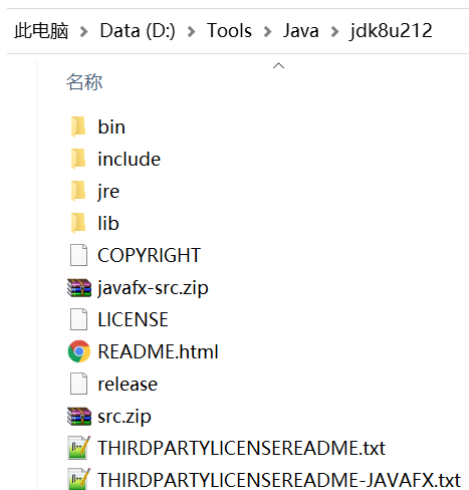
在“此电脑”右键--->属性--->高级系统设置--->环境变量：

系统变量中新建（如果已有就编辑）：

1. JAVA_HOME：指定为自己JDK 根路径。

说明：定义了一个系统变量，名称为 JAVA_HOME，值就是输入的JDK 路径。

注意JDK 根路径内容类似下图：



配置完成后，新打开一个cmd，执行以下命令：

```
1 | "%JAVA_HOME%/bin/java" -version
```

能正常显示以下内容才是配置成功：

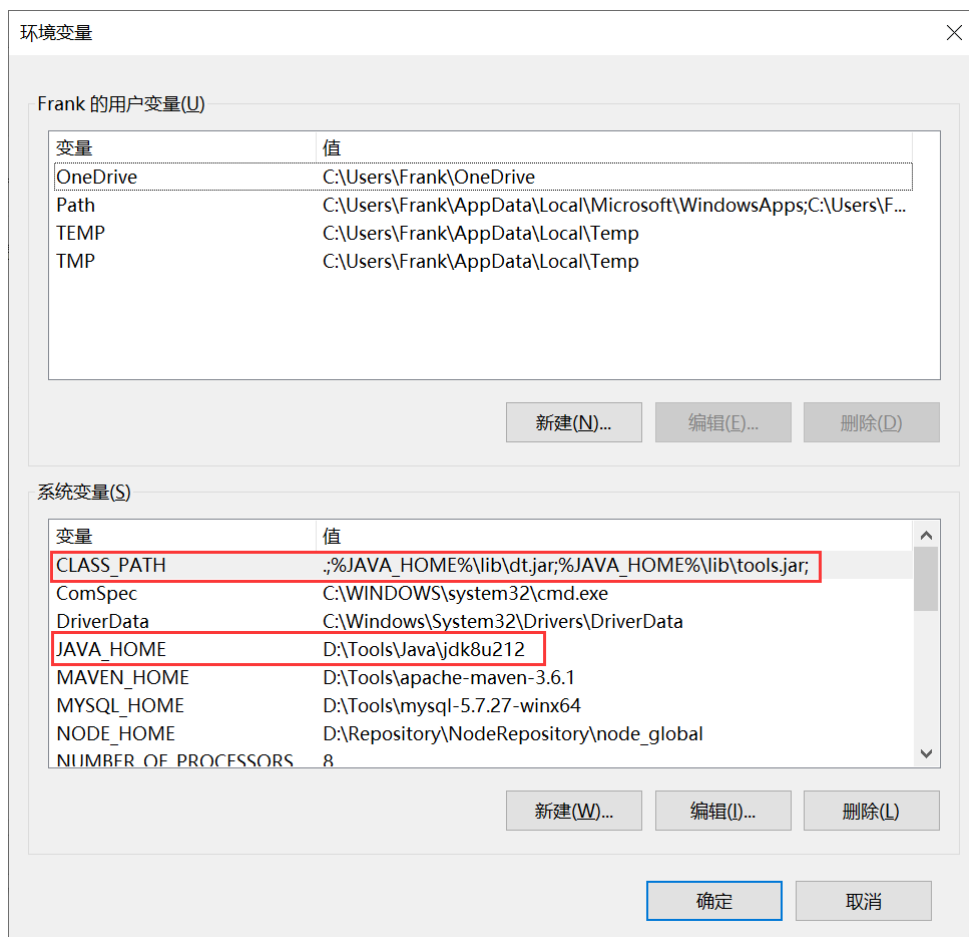
```
java version "1.8.0_261"  
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)
```

2. CLASS_PATH：

```
1 | .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
```

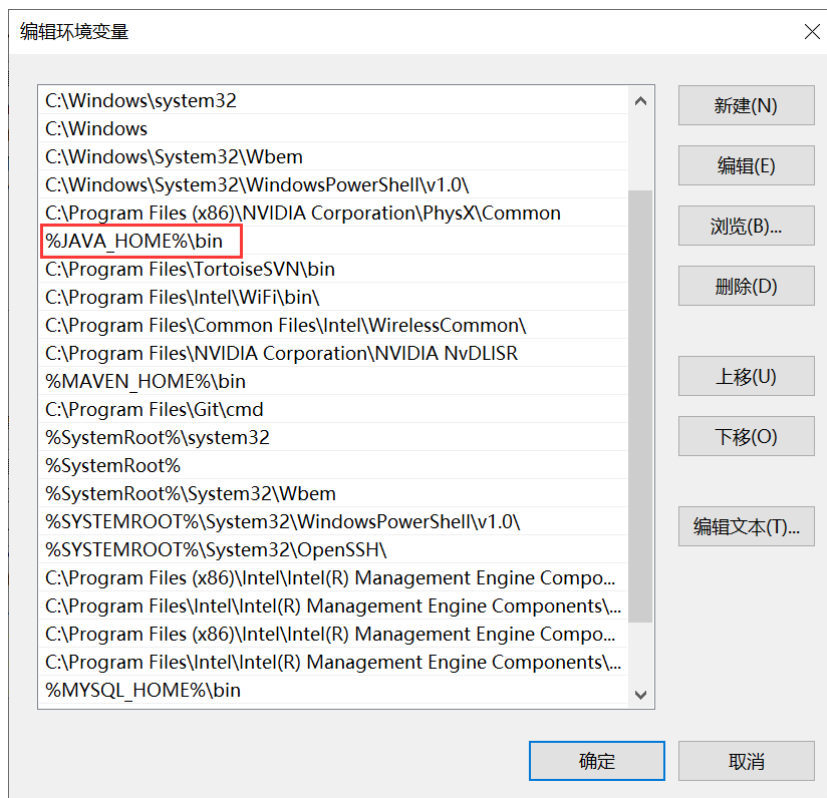
说明：以上定义了一个系统变量 CLASS_PATH，%变量名%表示Windows引用变量的方式，相当于把变量 JAVA_HOME 的值（JDK 路径）替换到 %JAVA_HOME%

- 以上配置如下图：



3. 配置Path变量：双击Path，添加 JDK 命令的路径：

1 | %JAVA_HOME%\bin



说明：

1. Path变量表示命令的系统全局路径，在该Path路径中的任何可执行命令，在任何路径下都可以直接使用。

2. 如果命令在Path路径中没有配置，只能通过命令的全路径执行。假如没有配置此处的%JAVA_HOME%\bin，java命令只能在CMD中输入全路径来运行：

```
1 | C:\Users\Frank>D:\Tools\Java\jdk8u212\bin\java 类名
```

3. 如果直接在路径 C:\Users\Frank 下输入java是不能执行的，会作为相对路径在 C:\Users\Frank 下一级查找叫java的命令。

Mac环境配置

1. 查看JAVA_HOME：

```
1 | /usr/libexec/java_home -V
```

可以显示JDK 的路径

2. 修改环境变量：如果使用默认的bash环境

```
1 | vim ~/.bash_profile
```

在最后输入

```
1 | export JAVA_HOME=第一步显示的JDK路径，复制到这里
2 | export CLASSPAHT=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
3 | export PATH=$JAVA_HOME/bin:$PATH
```

说明：和Windows类似，只是引用变量的方式为 \$变量名

:wq保存退出

3. 刷新配置：

```
1 | source ~/.bash_profile
```

IDEA管理项目

导入已存在项目

点击File--->New--->Project from Existing Sources...

- 普通项目直接选定项目根路径导入即可。
- Maven项目需要选定项目的pom.xml文件，然后点击OK导入

打开已有项目

点击File--->Open

- 同样的，普通项目选项目根路径，maven项目选pom.xml文件

返回欢迎界面

点击File--->Close Project，可以回到欢迎界面。

- 以上导入和打开项目的操作，也可以在欢迎界面操作。

IDEA基本配置

尽量保证 IDEA 版本是一致的，否则有些部分可能需要配置的内容不同而出现问题，目前我们使用的版本是2020.1.3。

IDEA需要对当前项目和新建项目都进行配置。

- 当前项目配置：File -> Settings
说明：对当前打开的项目进行配置。
- 当前项目结构：File -> Project Structure

说明：以上配置对新建的项目不会生效，建议使用以下操作方式：

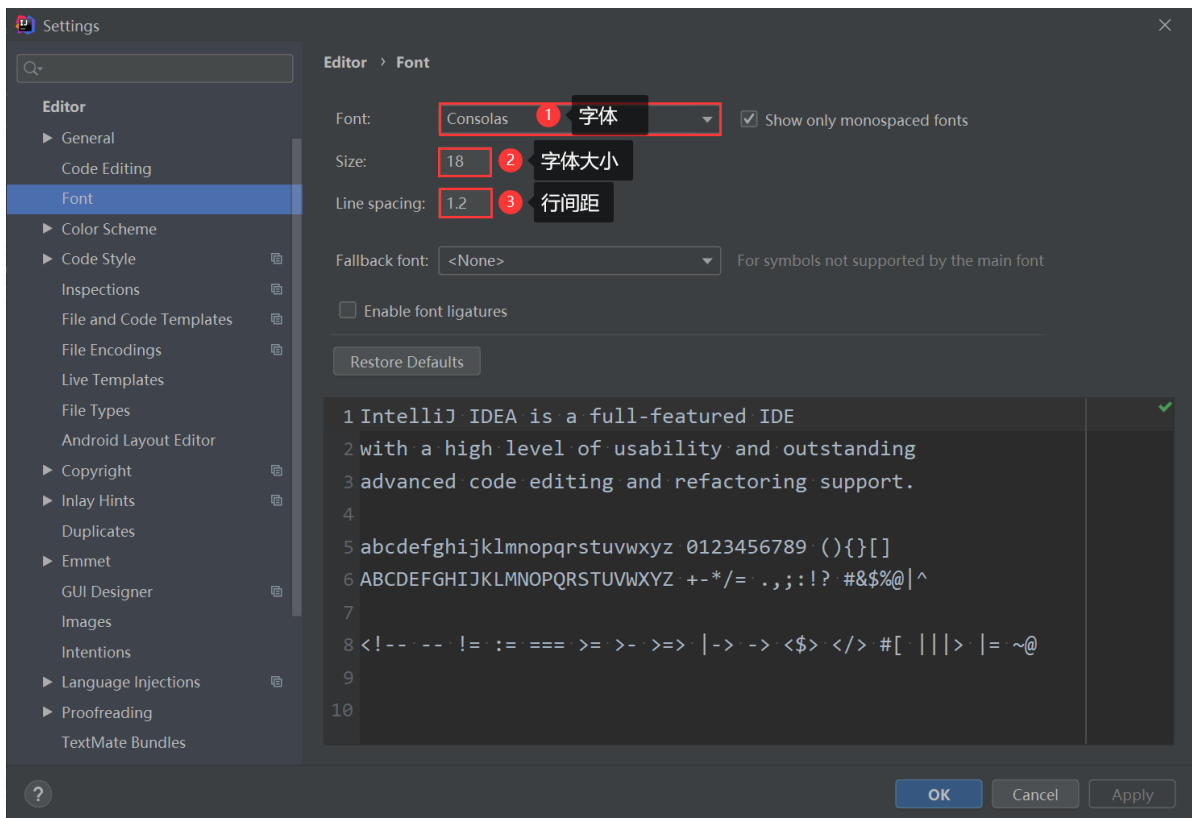
1. 先随便创建一个临时项目，再分别配置如下内容，完成配置后删除临时创建的项目，之后新建项目都会使用统一的配置。（新建项目配置有的，都不要在当前项目配）。
2. 新建项目配置：File -> New Projects Settings -> Settings for New Projects...
3. 新建项目结构：File -> New Projects Settings -> Structure for New Projects...

说明：

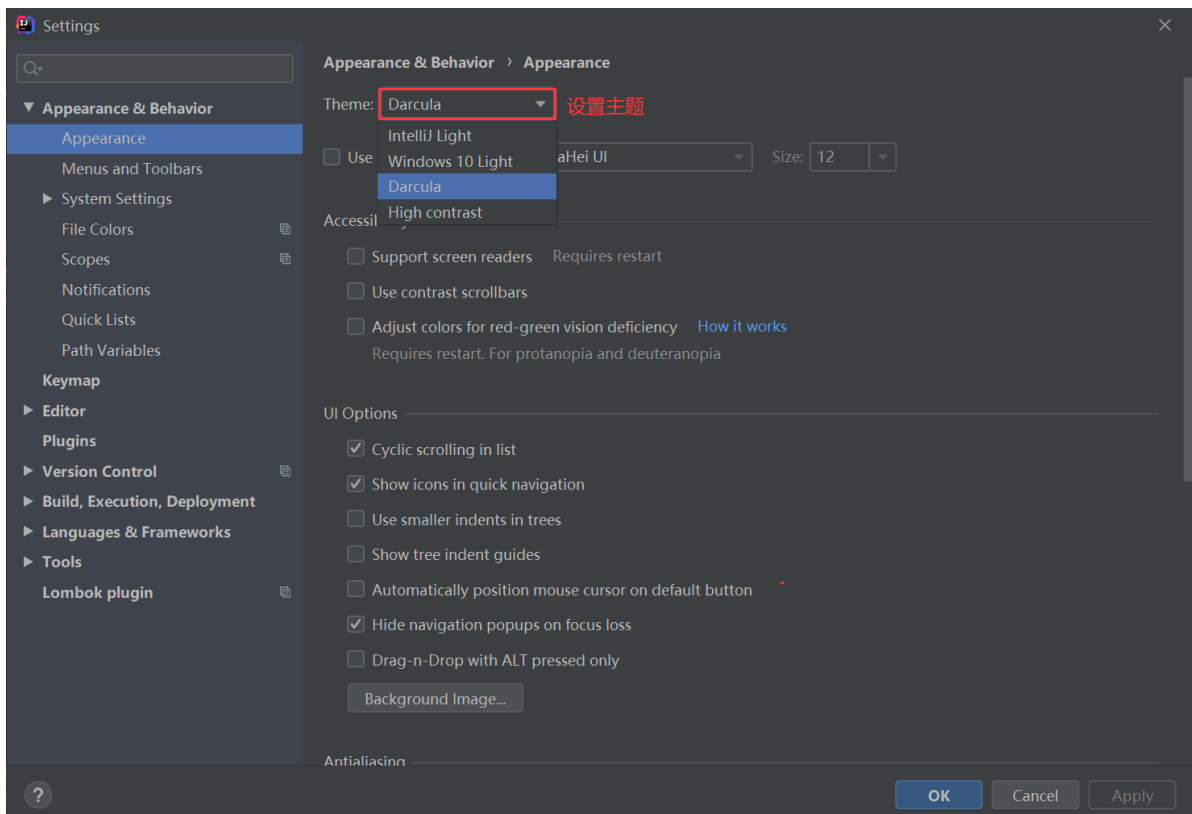
- 对新创建的项目进行配置，注意对当前打开的项目不生效。
- 不同版本的 IDEA 可能在不同的选项中，如 IDEA2018 就是Other Settings -> Settings for New Projects
- 请按照以下内容配置 IDEA

当前项目配置：File -> Settings

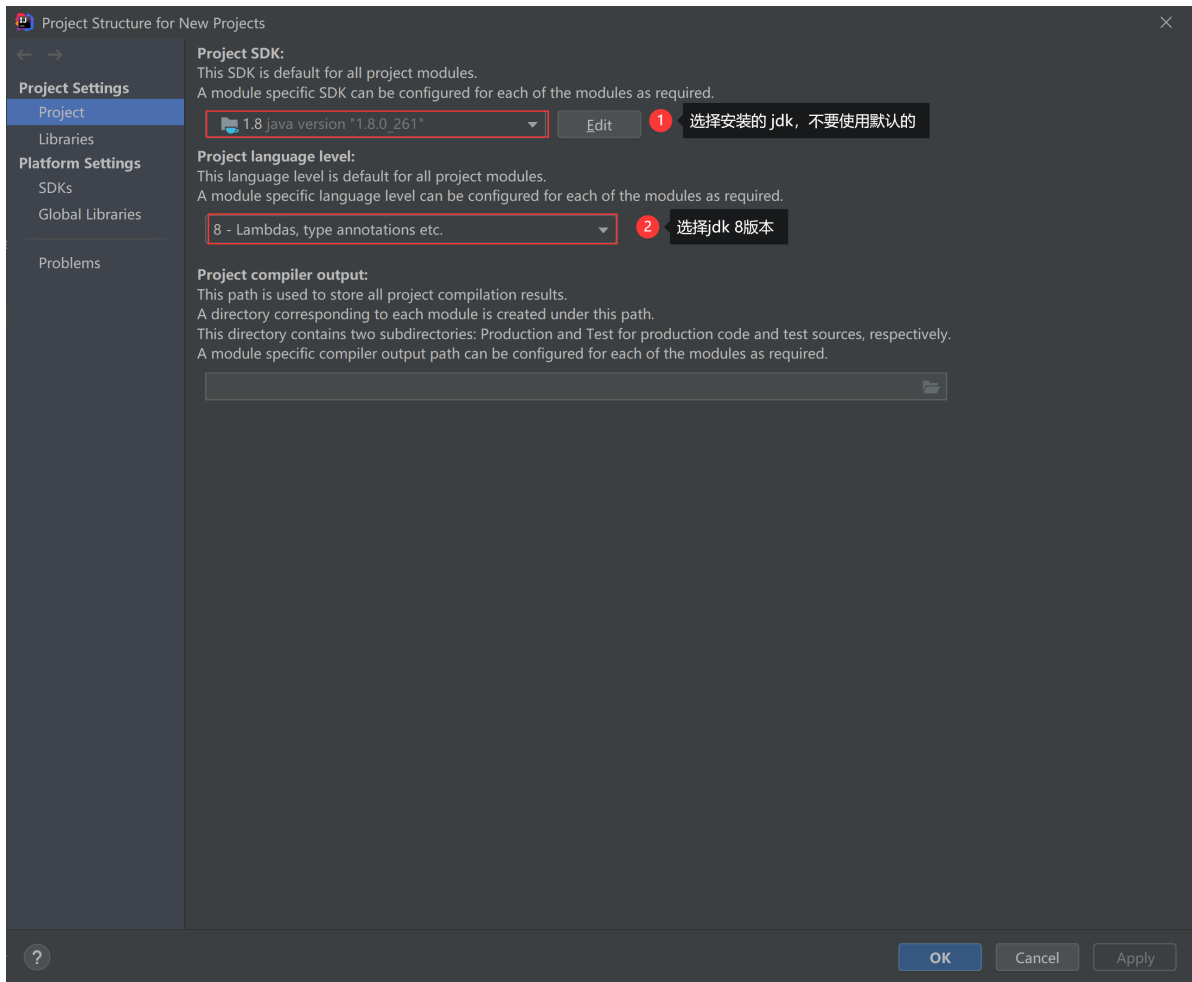
设置字体



设置主题

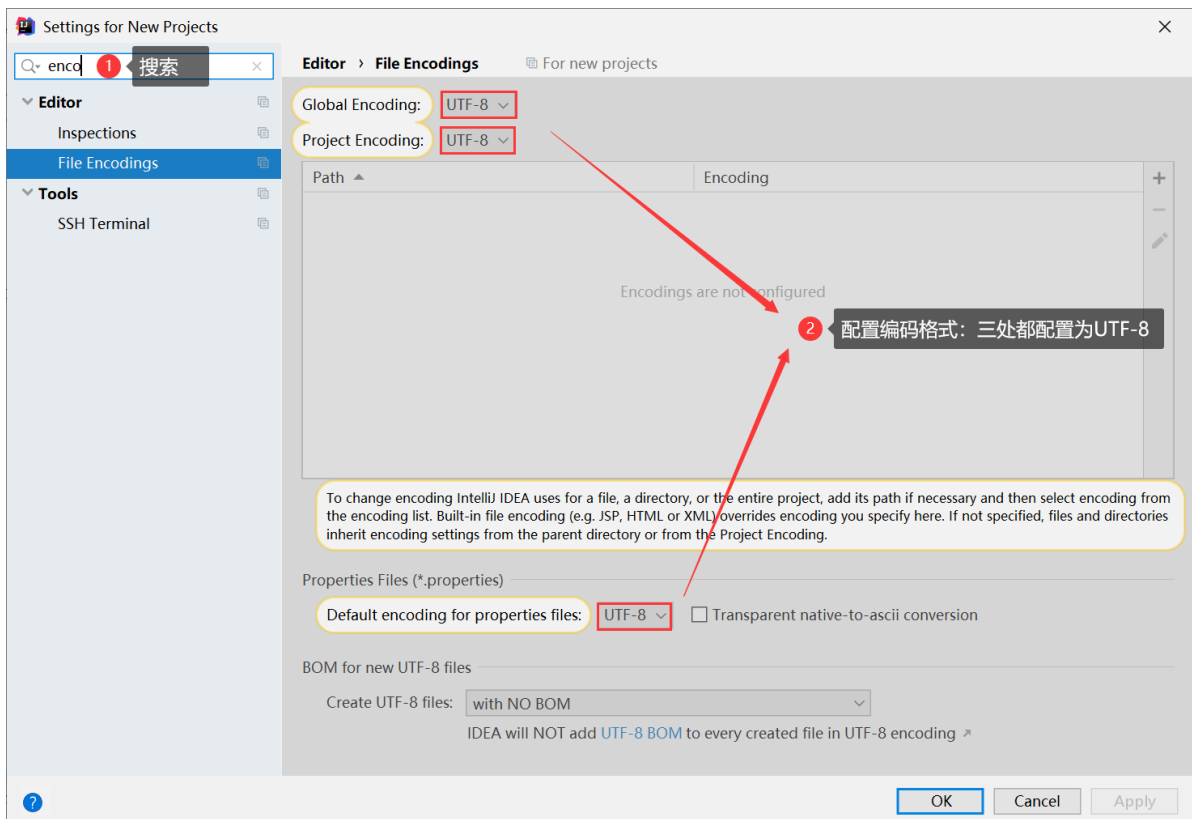


新建项目结构：Structure for New Projects

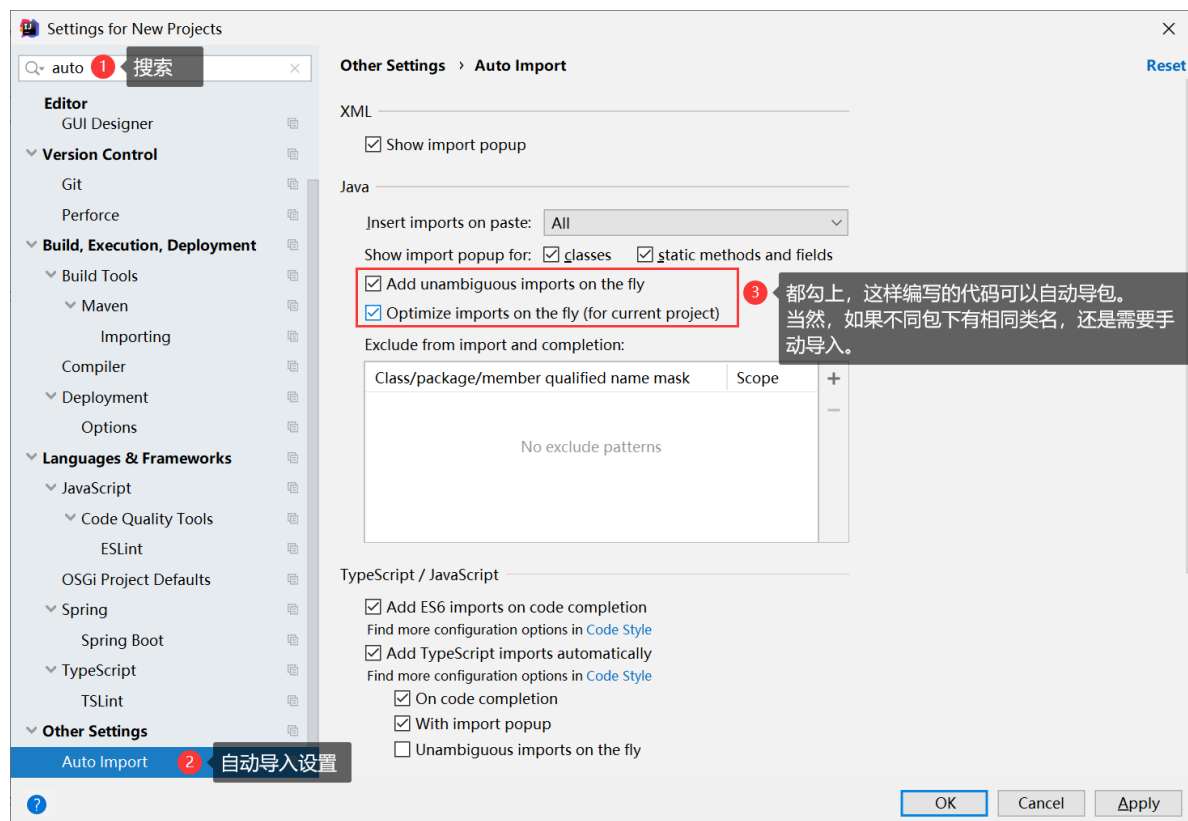


新建项目配置: Settings for New Projects

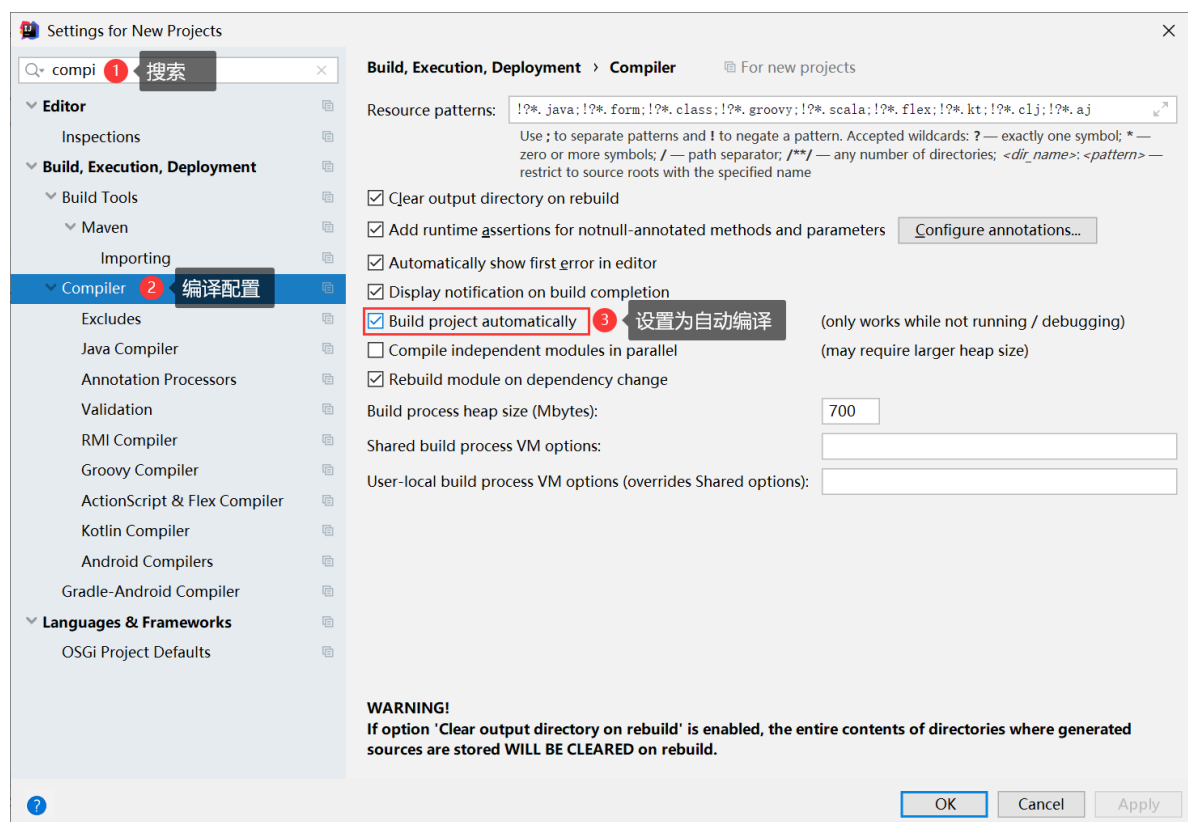
配置编码



设置自动导包

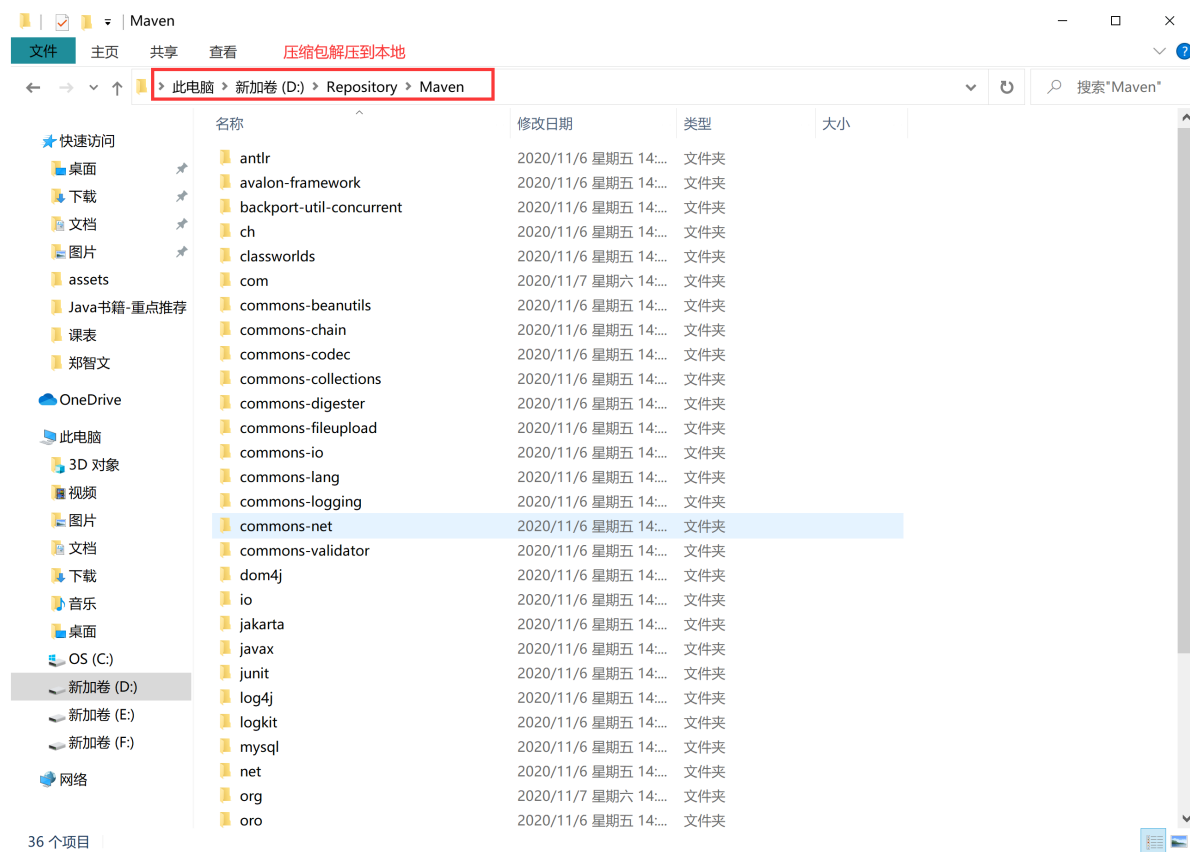


设置自动编译

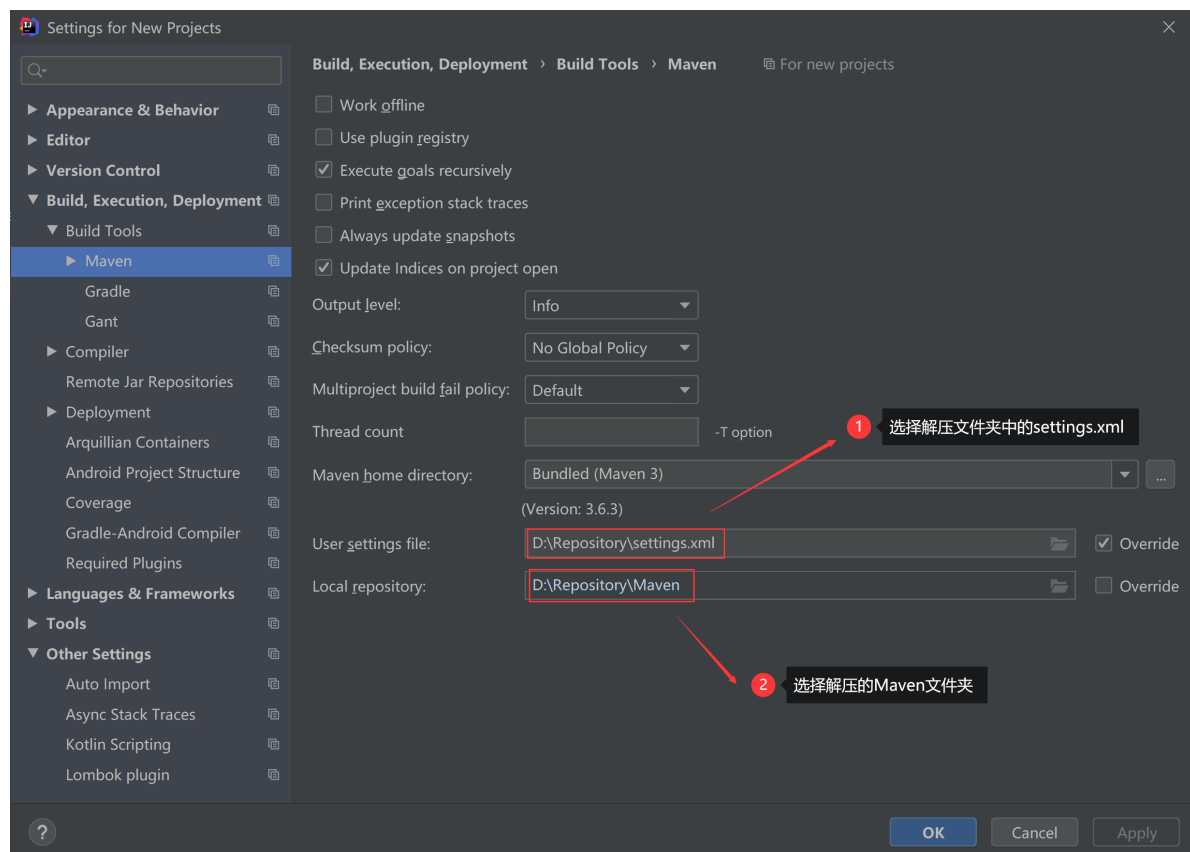


配置Maven本地仓库

下载settings.xml和Repository.zip，确保存放在本地某个路径中，之后别移动了。并解压压缩包，解压后是这样的：



IDEA配置Maven



IDEA常用快捷键/常用操作

几乎通用的快捷键：

复制： Ctrl+C

粘贴： Ctrl+V

剪切： Ctrl+X（去除选中内容，并将选中内容放在剪贴板，可以粘贴到其他地方）

文件内搜索：Ctrl+F，在输入字符串搜索

代码自动修复：

- 说明：代码编译错误的地方，包括没有引包，出现编译时异常等，都可以使用自动提示来修复：
- 操作方式：光标移动到代码飘红的地方：alt+Enter，或是点前面红色的小灯泡

代码格式化

- 说明：选中的内容进行格式化，如果没有选中任何内容，只是光标放在一个地方，会对所有内容进行格式化
- 操作方式：菜单栏中，选择Code -> Reformat Code

代码自动生成

- 说明：
可以自动生成构造方法、toString方法、父类方法重写
如果当前类中有属性时，还可以生成：Getter方法、Setter方法、Getter和Setter方法、equals和hashCode方法
- 操作方式：在当前类的 {} 花括号中右键，选择generate，出现的面板中选择Constructor即生成构造方法，其他的如Getter、Setter等。

生成代码块等

- 说明：选中某部分代码后，可以生成很多代码块，如判断条件代码块，循环代码块，同步代码块，捕获异常代码块等等
- 操作方式：选中某部分代码，在菜单栏中，选择 Code -> Surround With..., 再根据需要选择需要生成的代码块

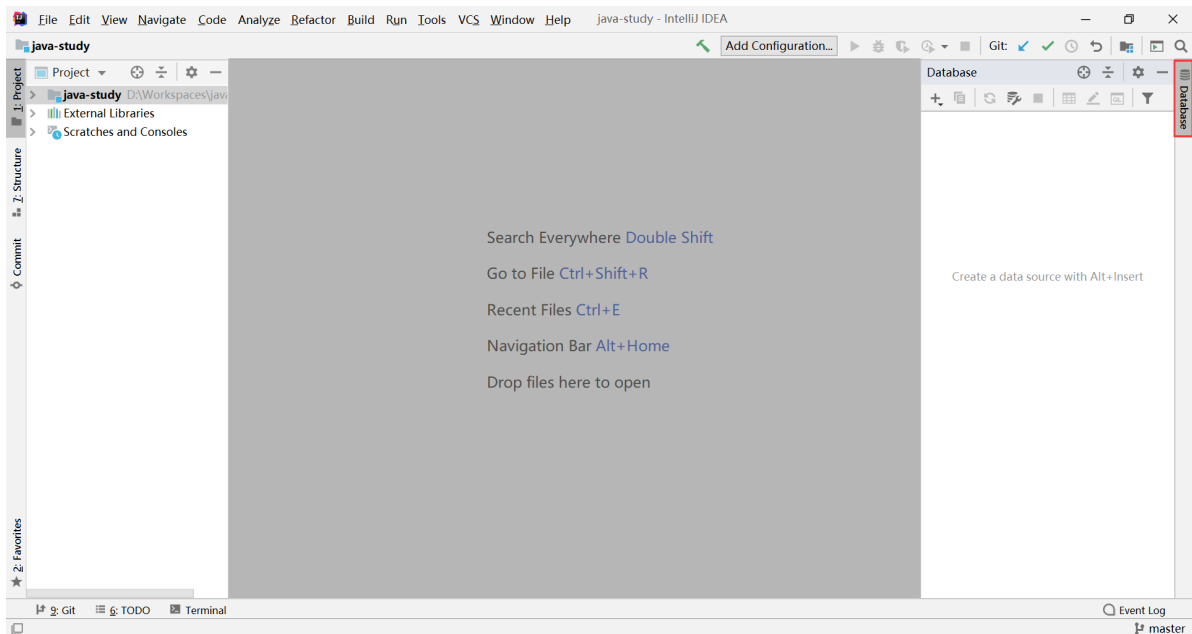
数据库GUI客户端配置

可以在IDEA中配置数据库的GUI客户端（可视化界面）来对数据库进行操作。

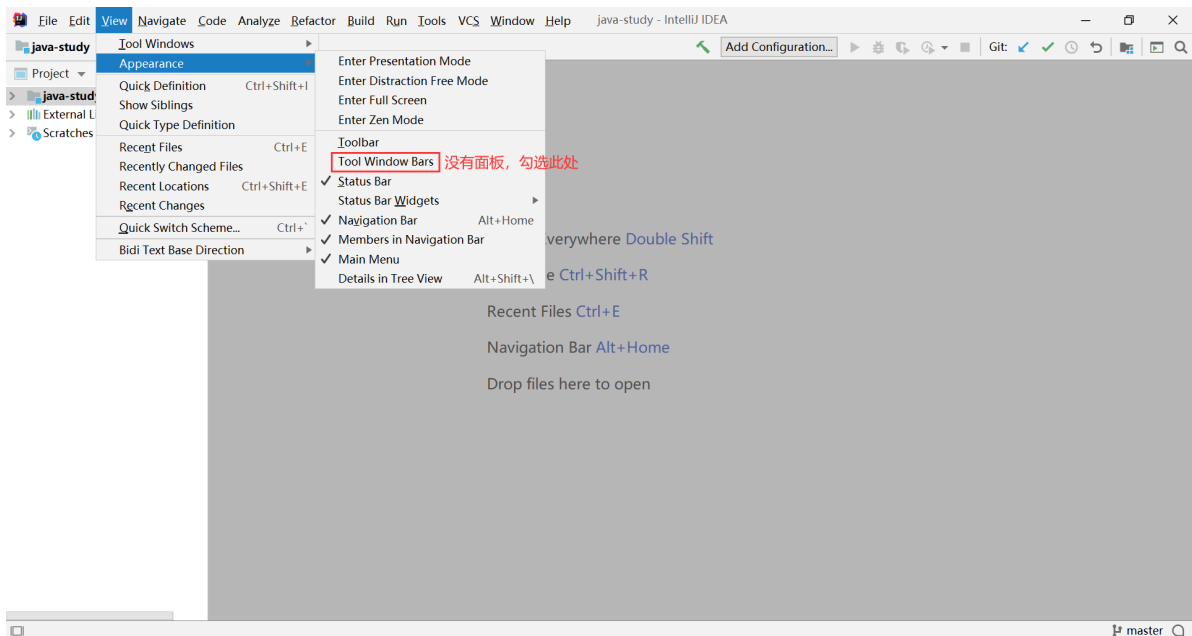
需要注意的是这里和代码一点关系都没有，只是提供了一种可视化的客户端。

和数据库 JDBC 操作类似，需要先有数据库驱动包，再使用数据库驱动包的驱动类来创建数据库连接。

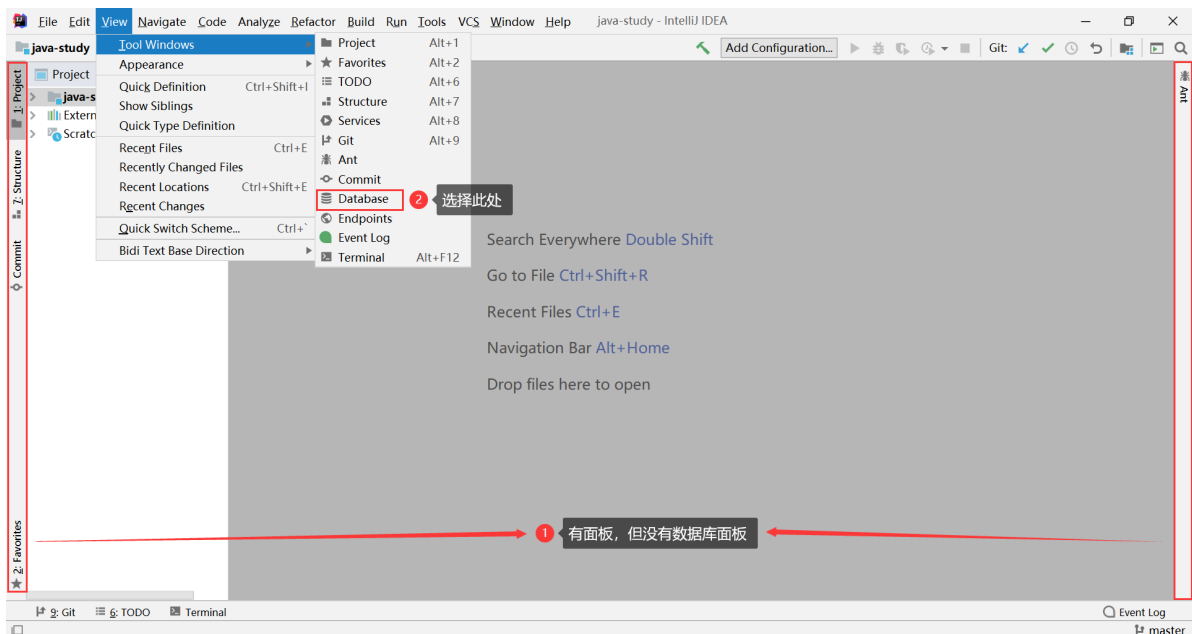
打开数据库面板



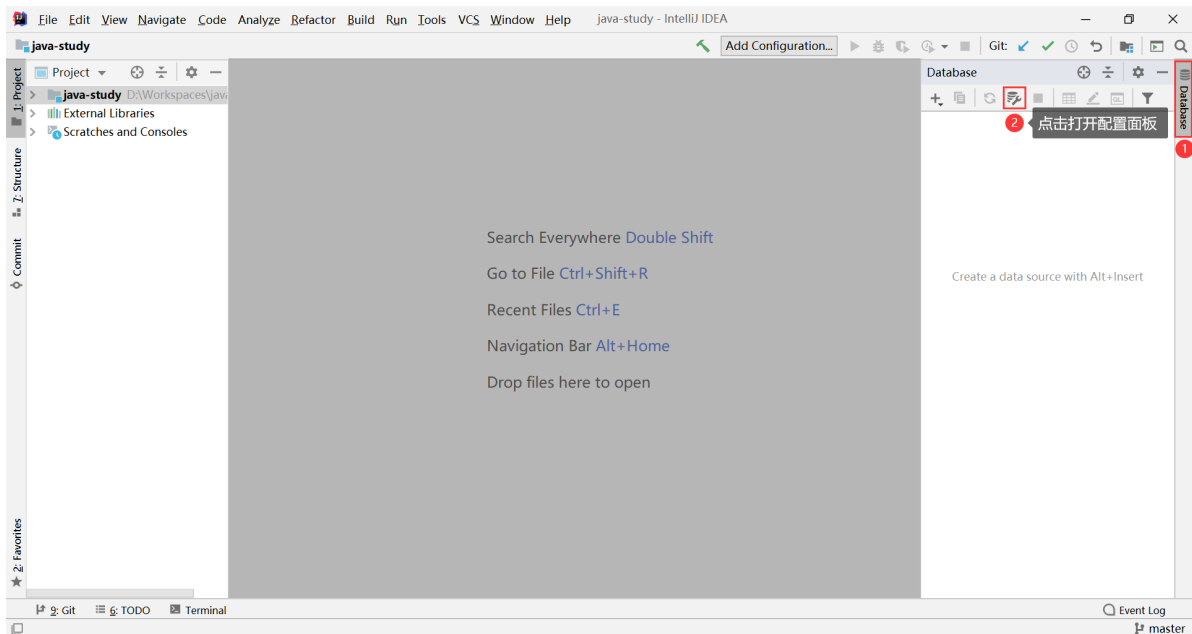
如果找不到面板的，点击 View -> Appearance -> Tool Window Bars



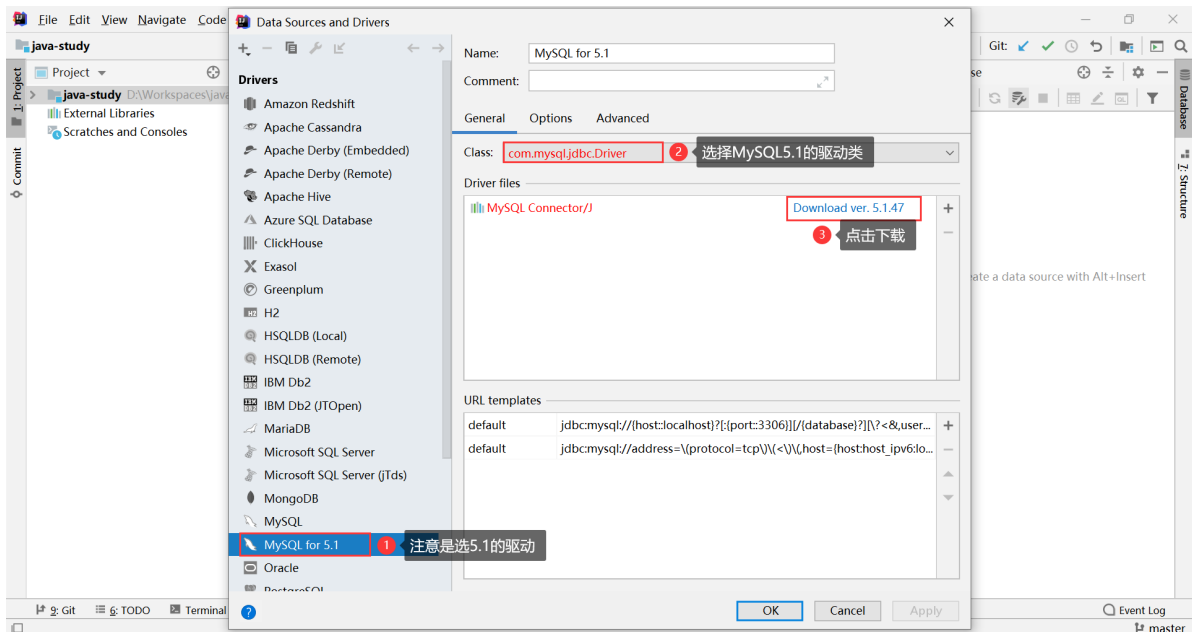
如果有面板，但是找不到数据库面板，点击 View -> Tool Windows -> Database



打开数据库GUI配置

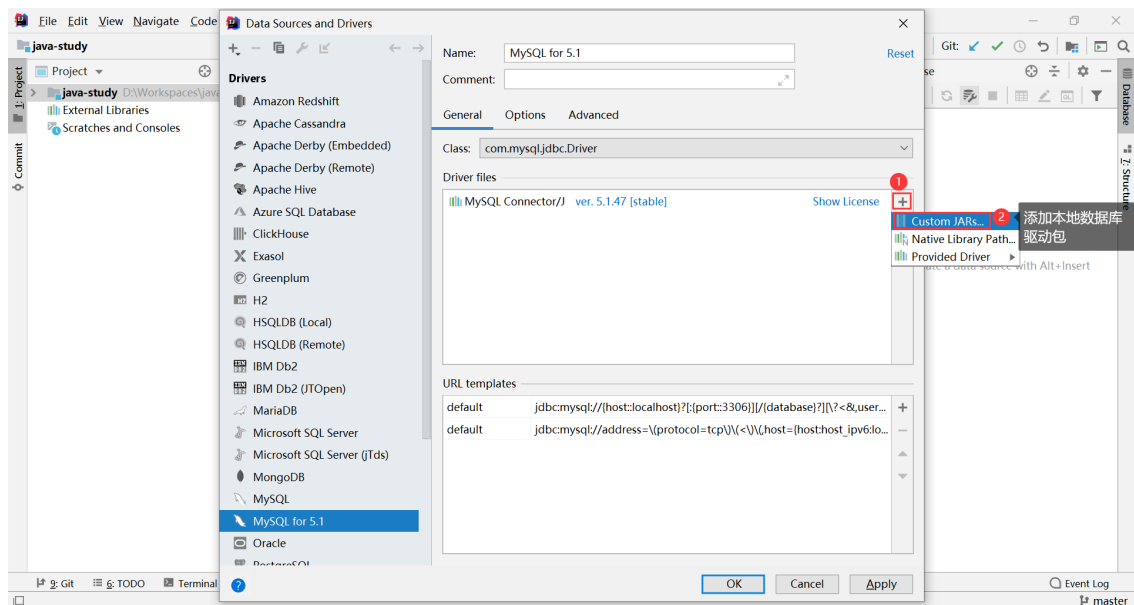


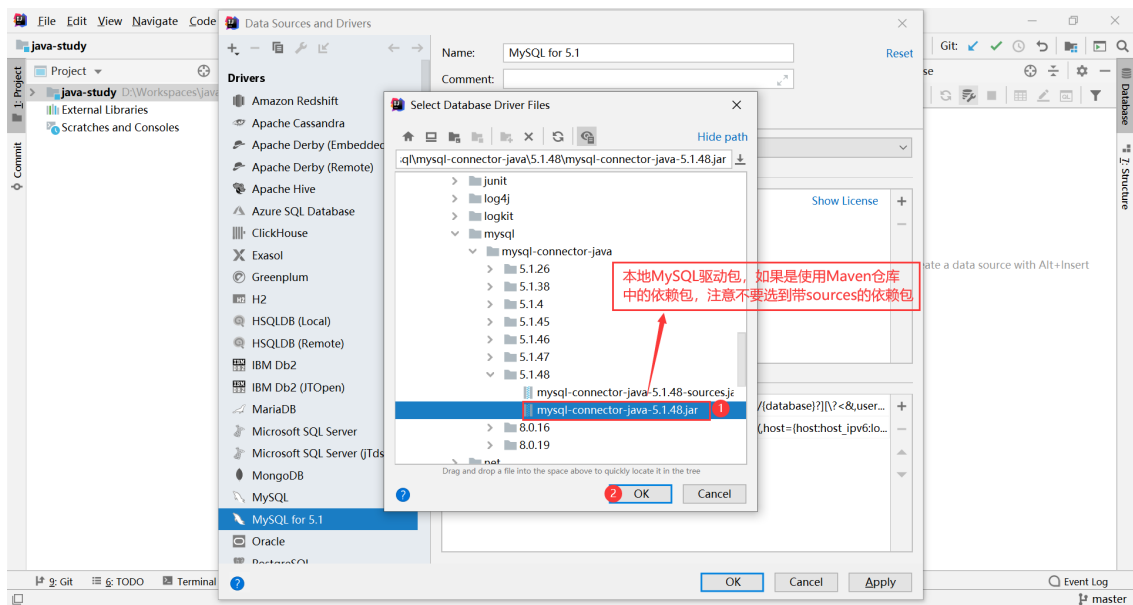
配置数据库驱动



点击Download后会自动下载，但因为是从国外网站下载数据库驱动包，可能会比较慢，或下载失败，可以采取如下方式解决：

1. 下载失败的，反复多下载几次。或是找个好一点的网络再加载。
2. 直接使用本地的数据库驱动包：

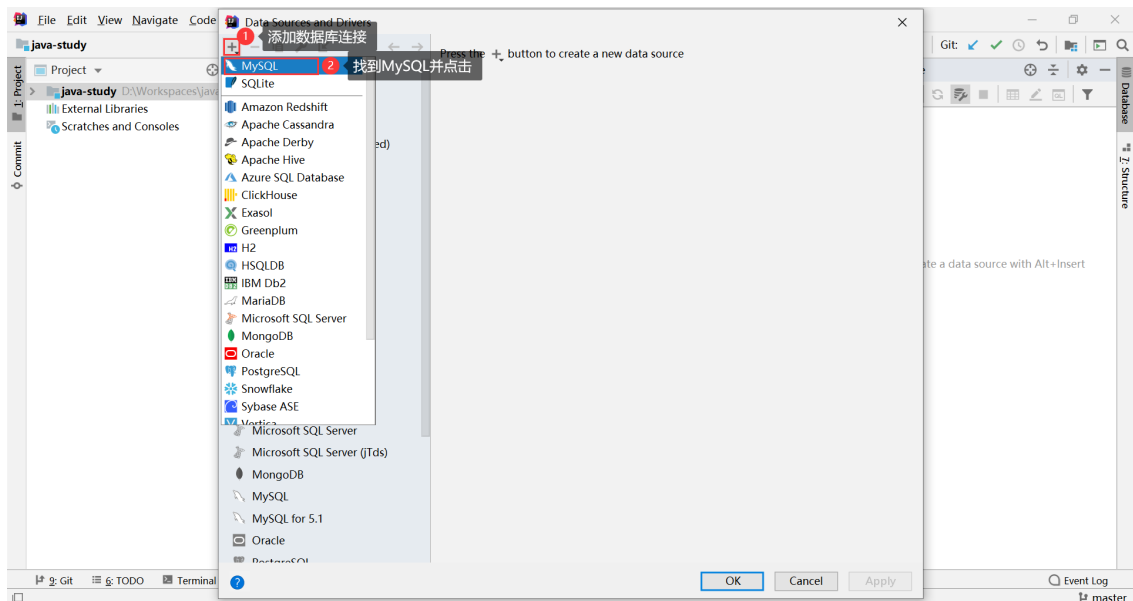




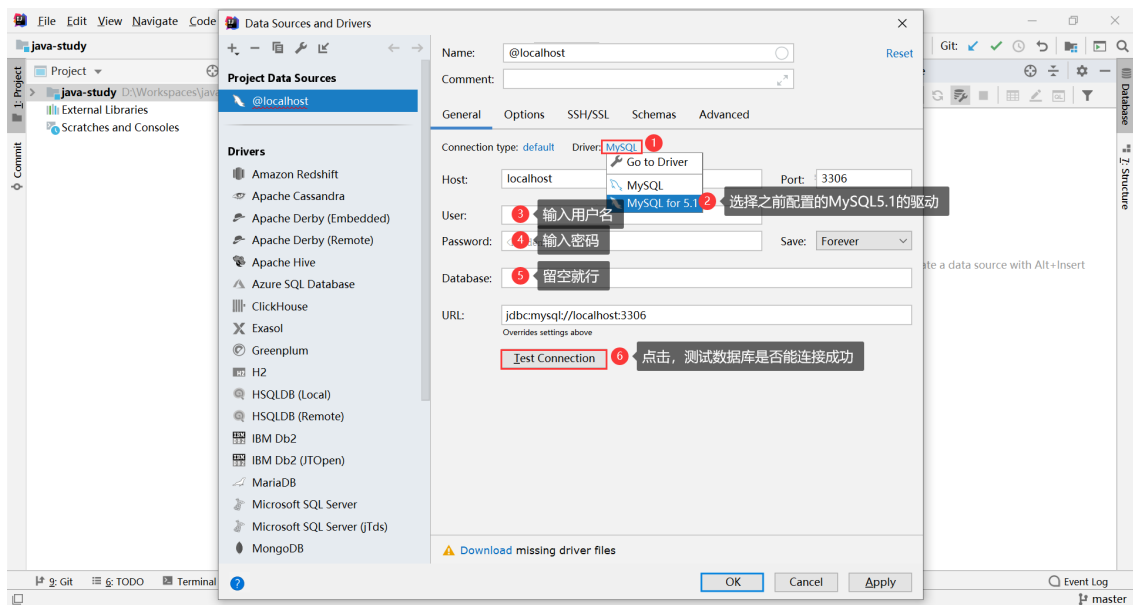
配置数据库连接

在配置好数据库驱动后，GUI客户端就可以通过驱动包来创建数据库连接了，但还需要手动配置数据库连接。

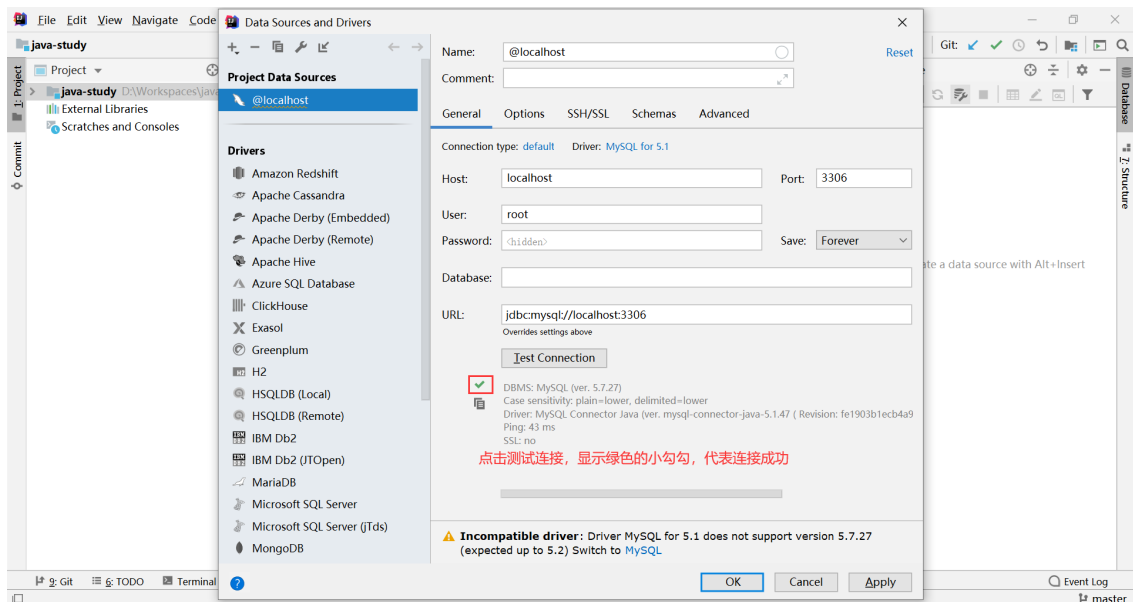
- 添加数据库连接：



- 配置数据库连接

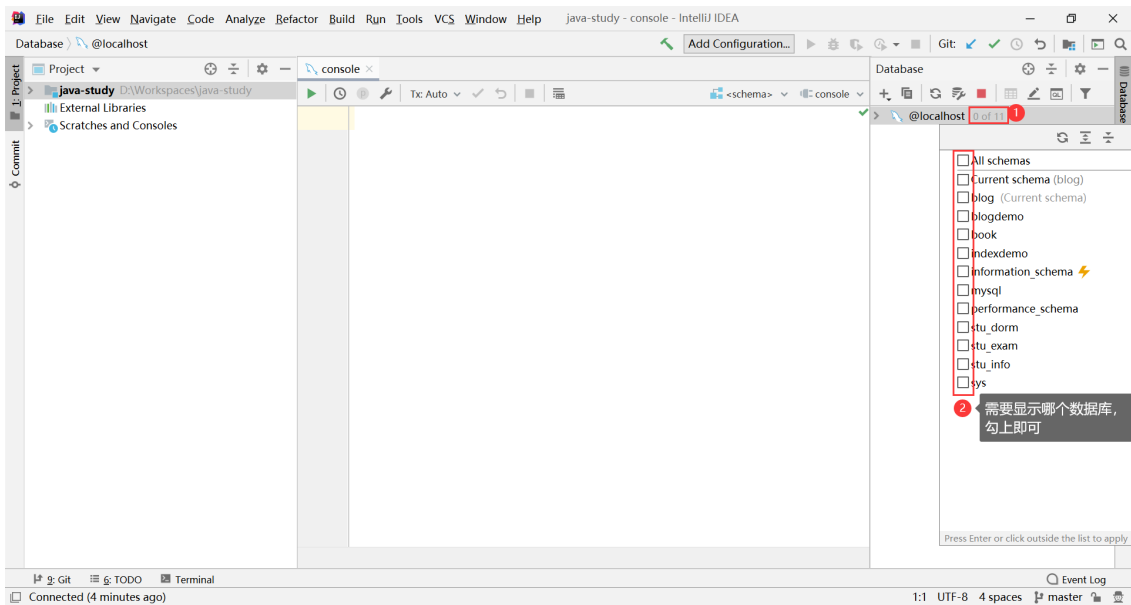


- 连接成功显示如下

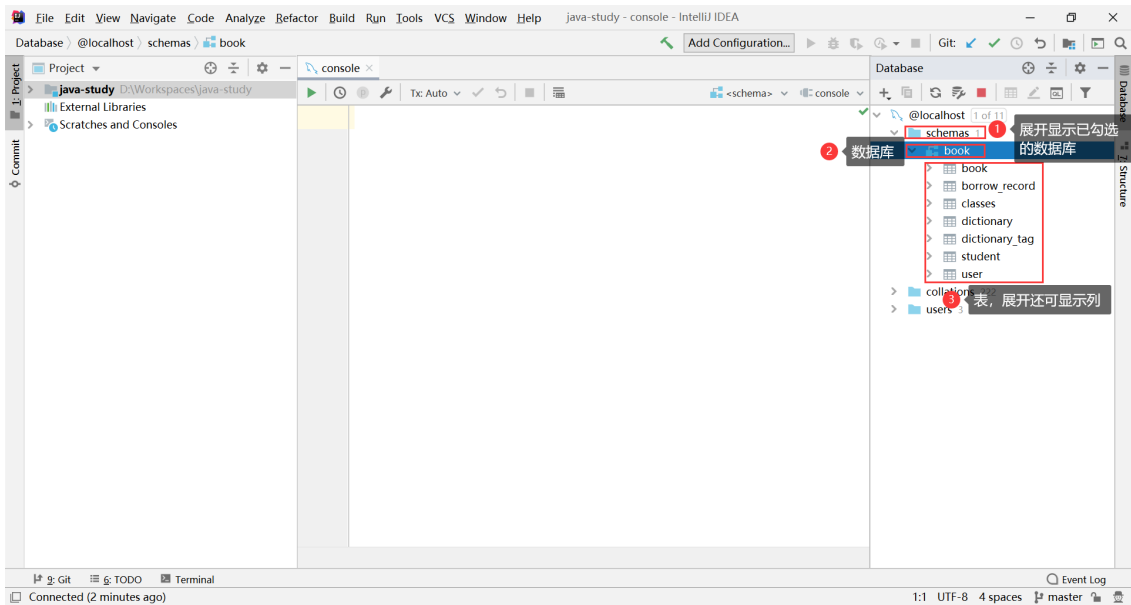


使用数据库GUI客户端

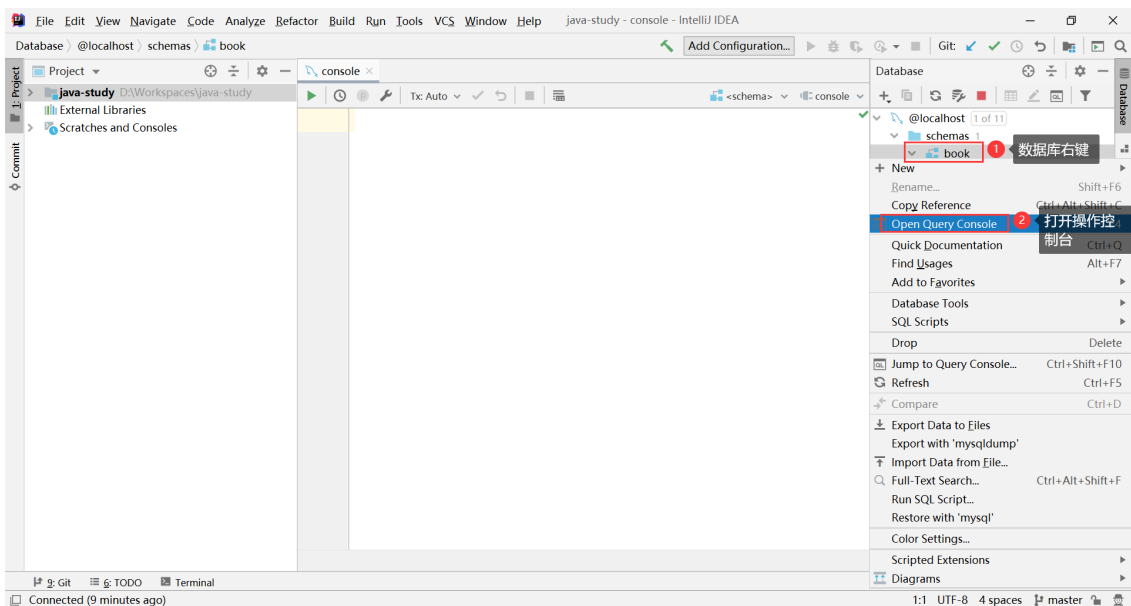
- 选择要显示的数据库

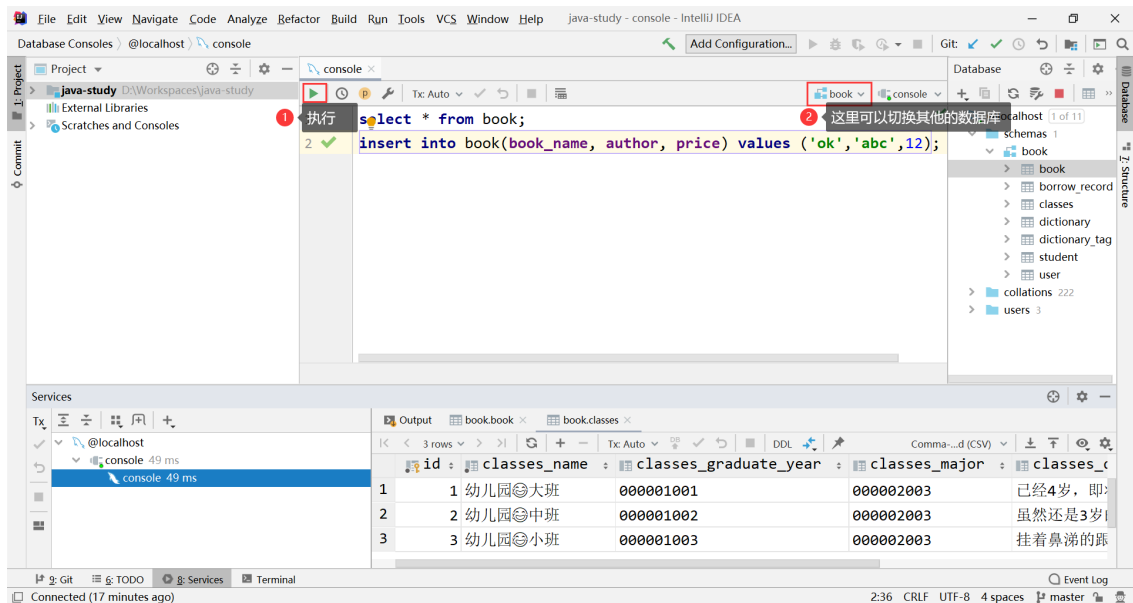


- 显示数据库、表及表字段



- 还可以打开数据库控制台执行 SQL 语句:





Web项目配置

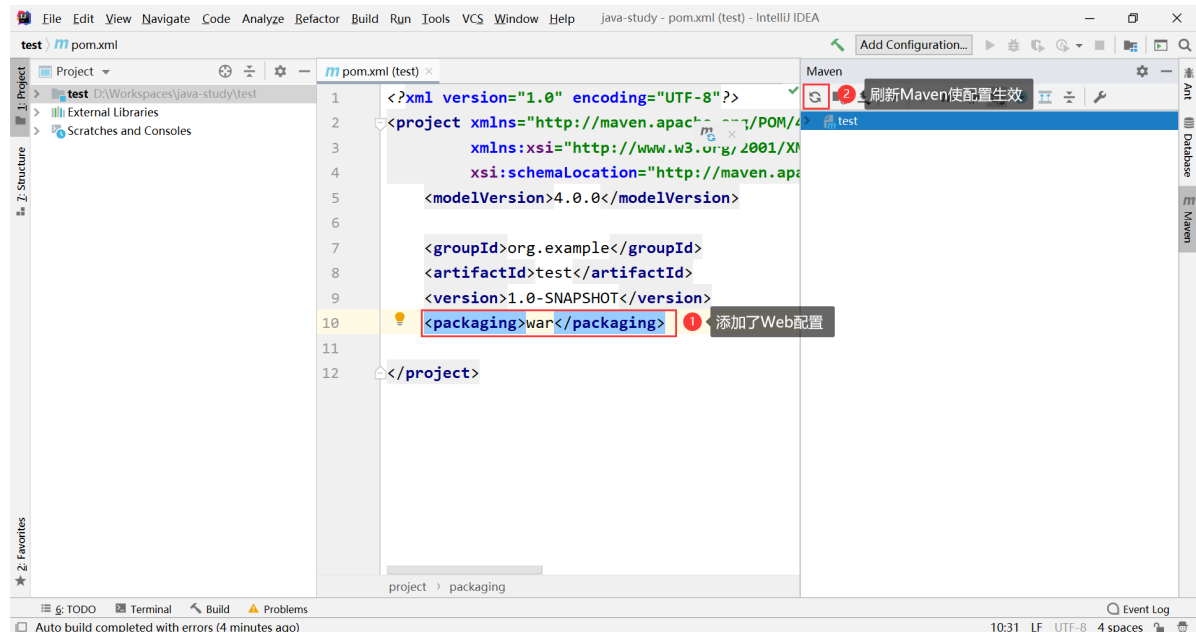
这里都是基于Maven的Web项目。

配置为Web项目

先在Maven的pom.xml配置文件中配置项目为Web项目：

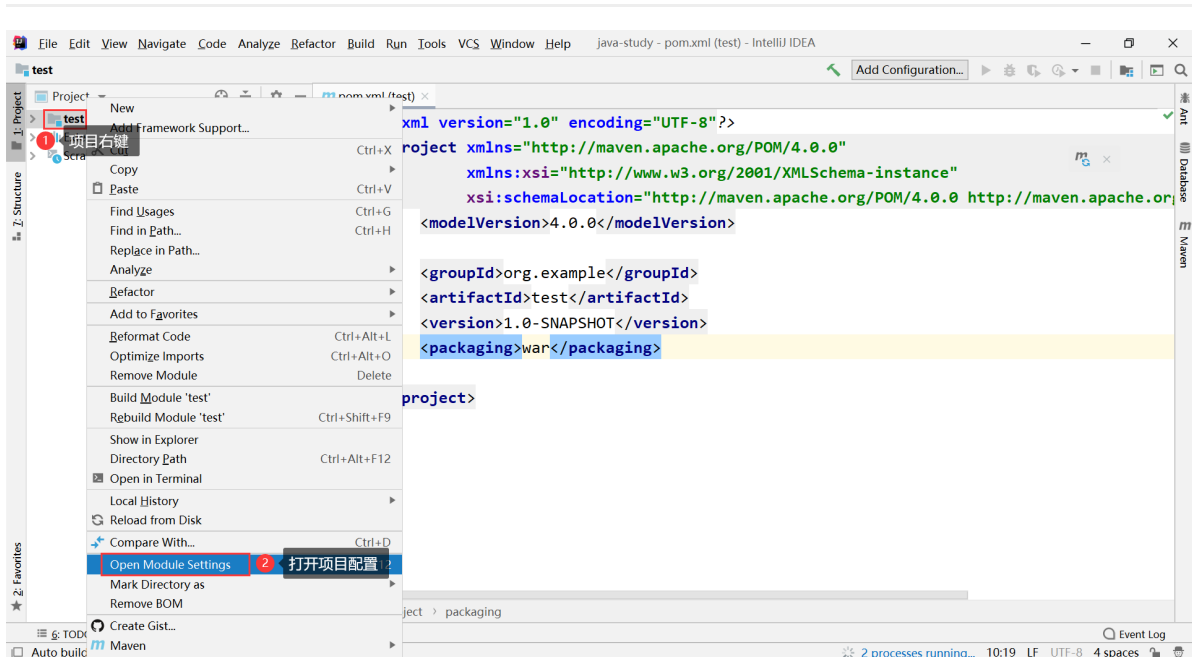
```
1 <packaging>war</packaging>
```

Maven项目的配置文件pom.xml修改以后，都需要在Maven面板中刷新一下，否则不能自动更新生效：

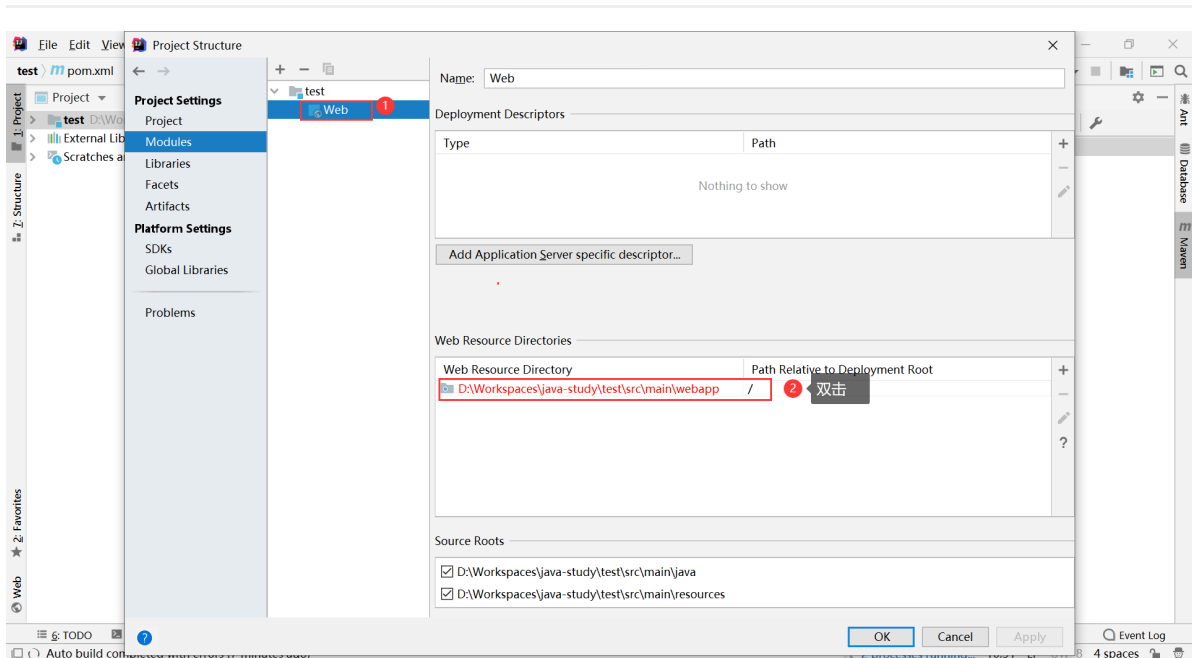


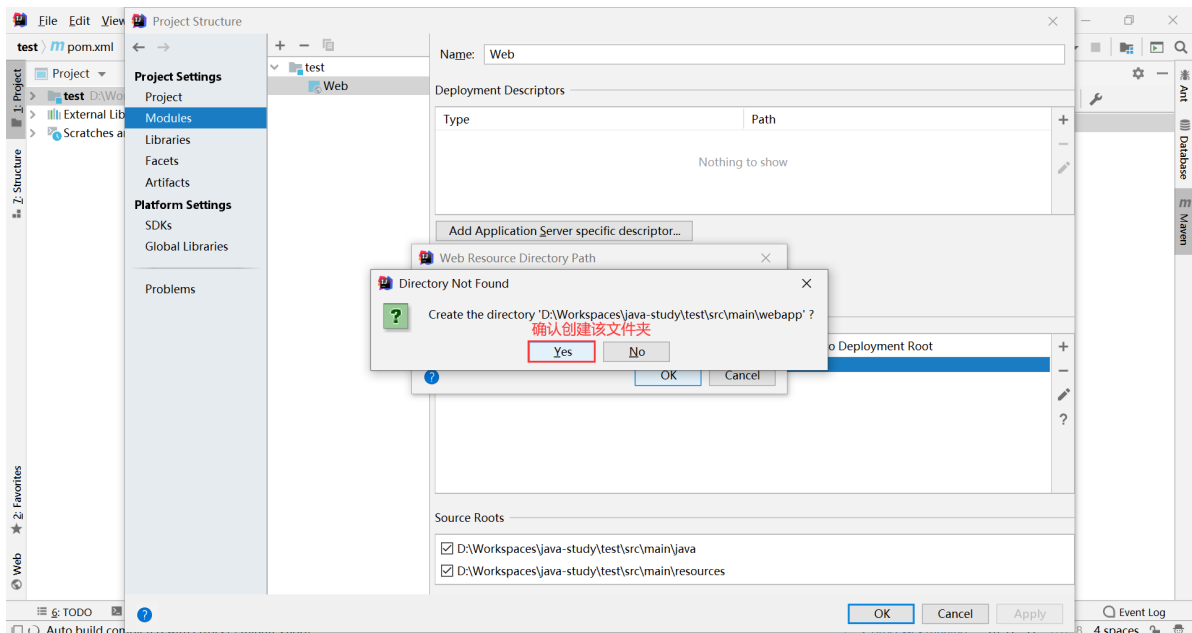
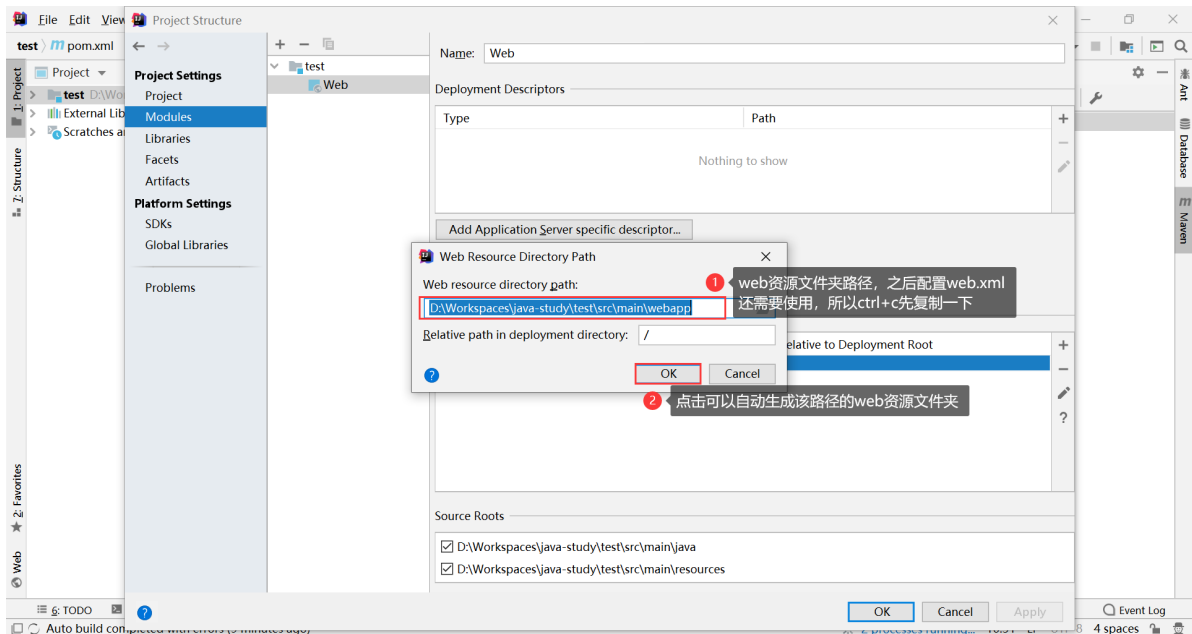
IDEA会自动生成项目的Web配置，但里边有内容还需要修改：

打开项目的配置

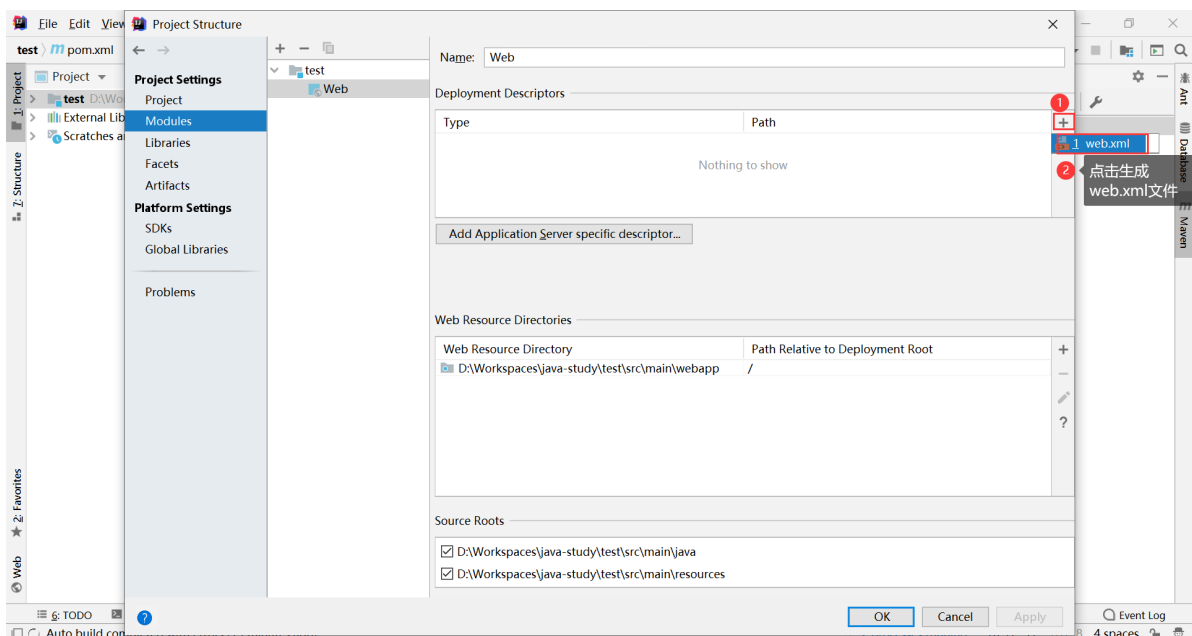


配置Web资源文件夹

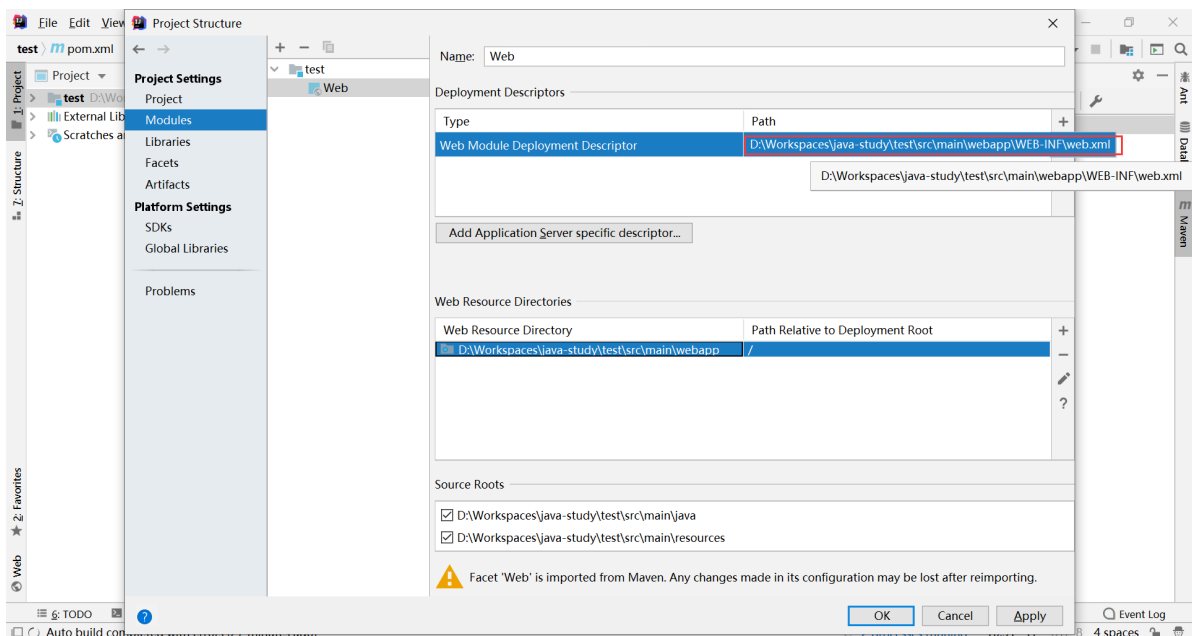
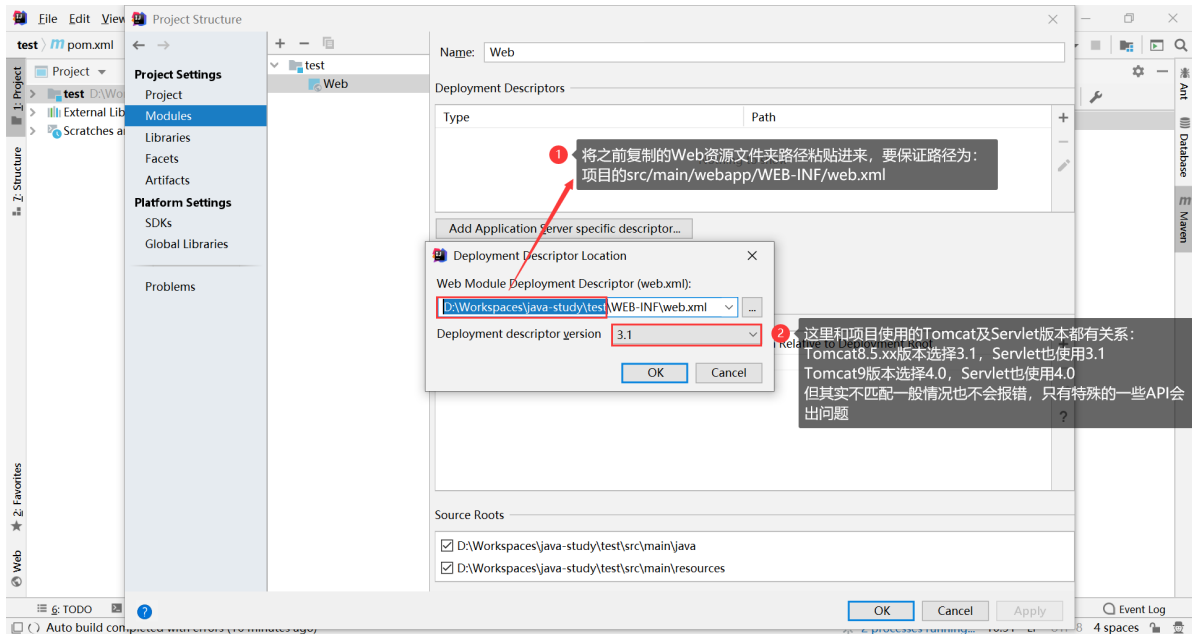




配置Web项目描述文件web.xml



注意web.xml一定要位于项目src/main/webapp/WEB-INF下：



IDEA中部署Web项目

配置编码

这里的编码是为 IDEA 中使用 Tomcat 时，打印日志设置的编码。如果不设置，默认 Tomcat 输出的日志可能为乱码。

打开 IDEA后，选择 Help -> Edit Custom VM Options...

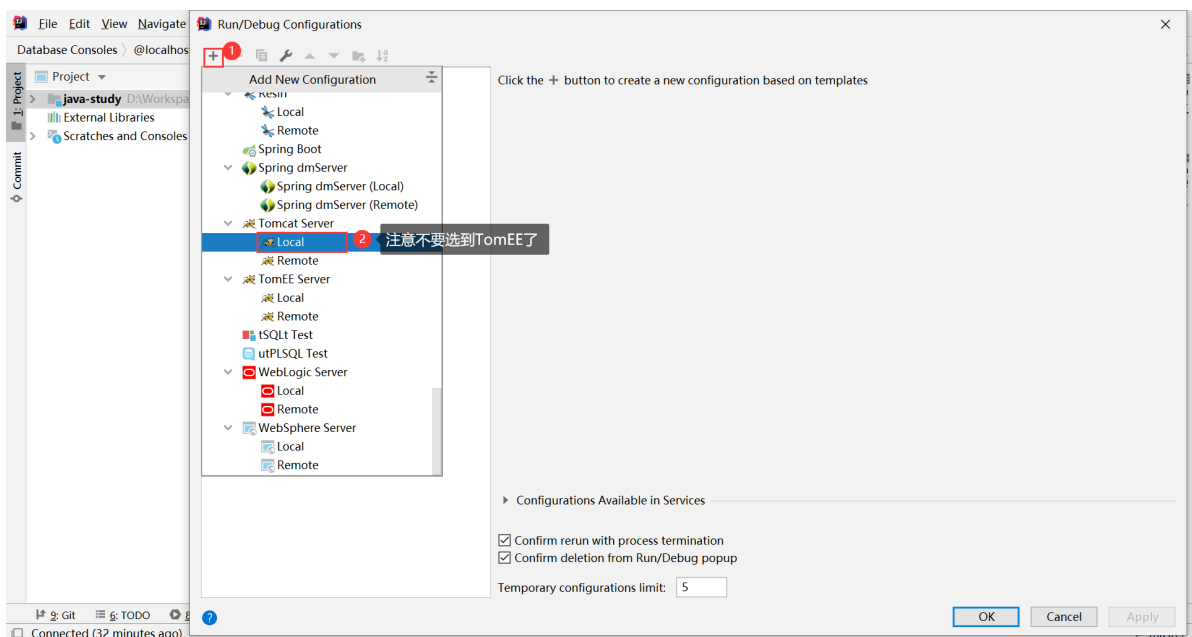
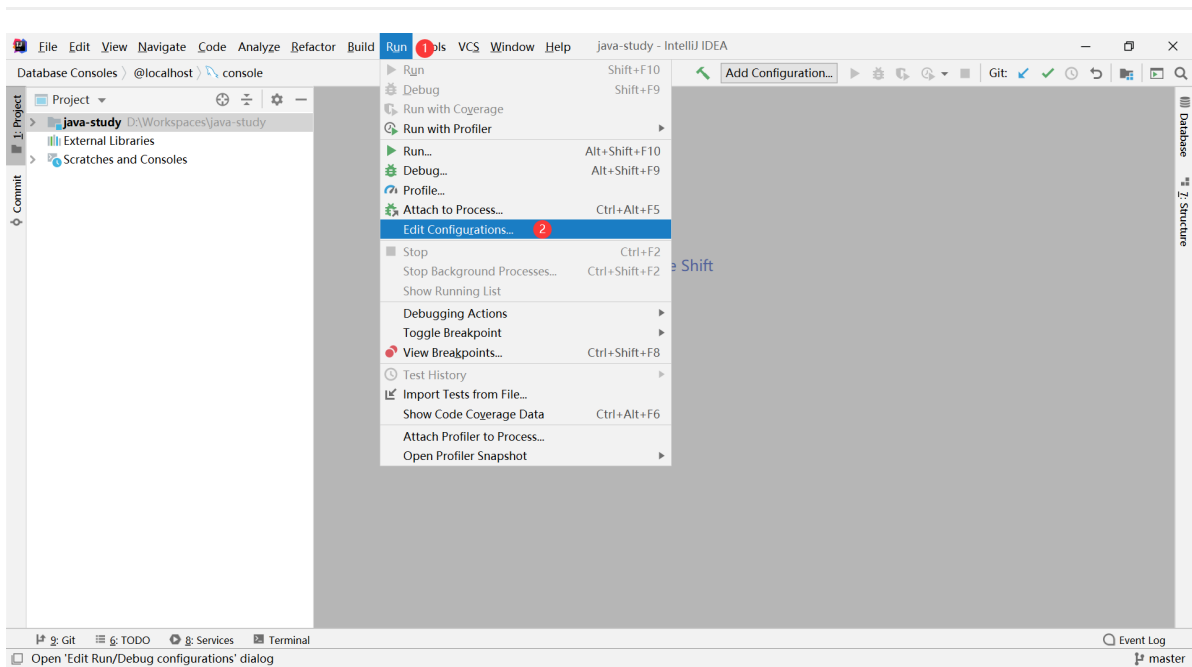
在文件最后回车换行，添加新的一行：

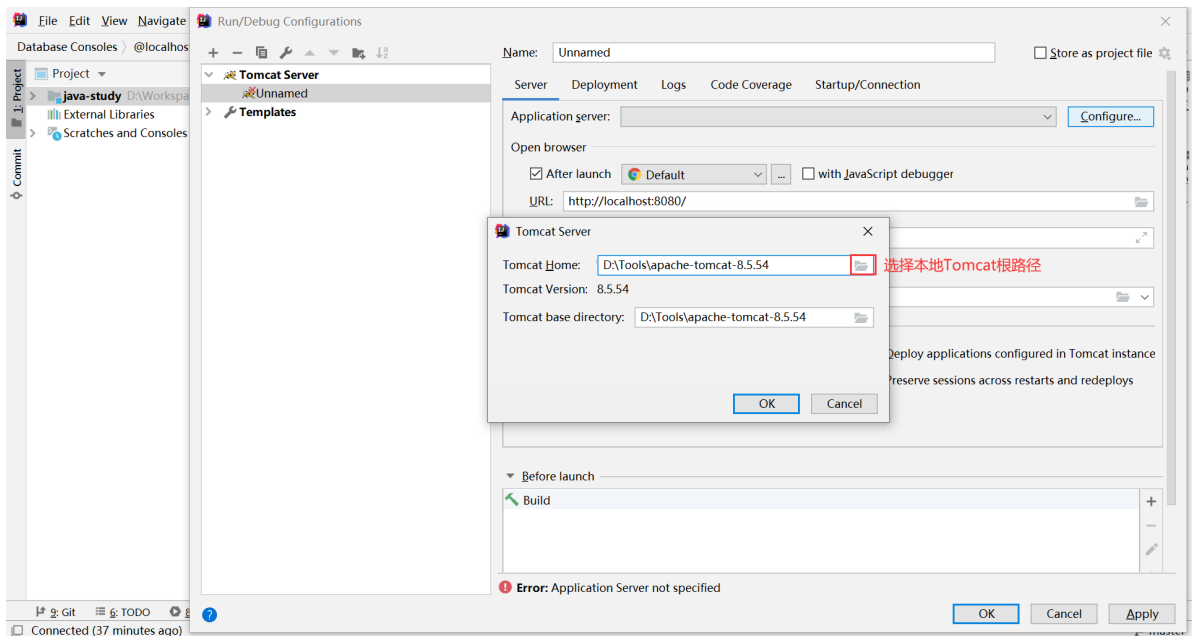
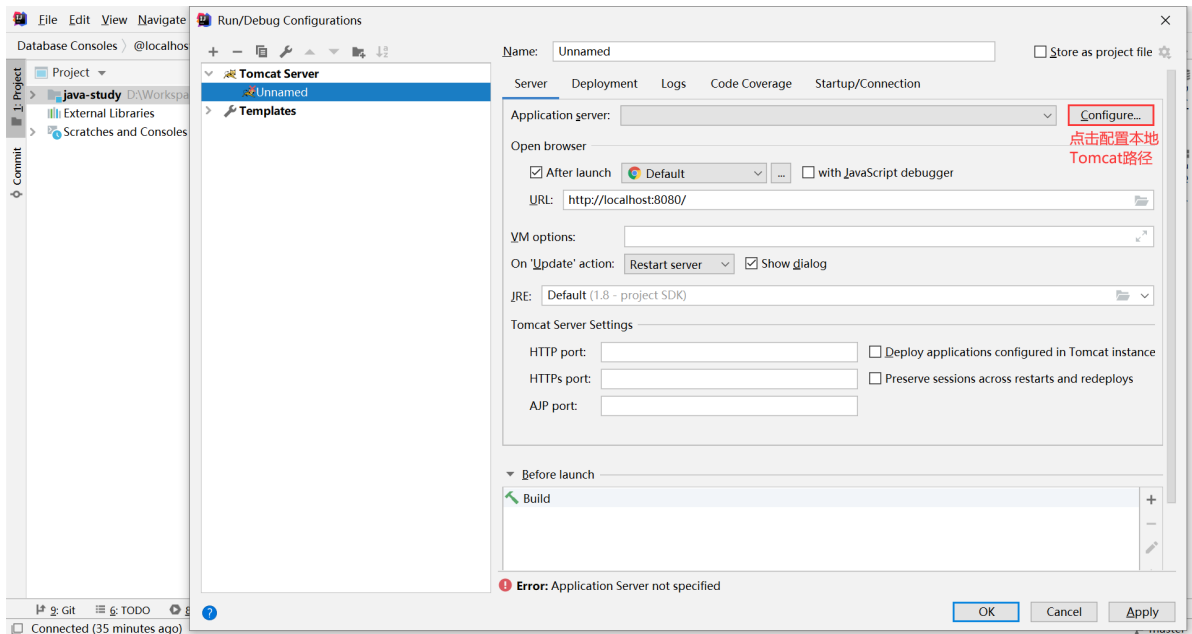
```
1 | -Dfile.encoding=UTF-8
```

配置好的内容如下：

```
idea64.exe.vmoptions
1 -Xms128m
2 -Xmx4096m
3 -XX:ReservedCodeCacheSize=240m
4 -XX:+UseConcMarkSweepGC
5 -XX:SoftRefLRUPolicyMSPerMB=50
6 -ea
7 -XX:CICompilerCount=2
8 -Dsun.io.useCanonPrefixCache=false
9 -Djdk.http.auth.tunneling.disabledSchemes=""
10 -XX:+HeapDumpOnOutOfMemoryError
11 -XX:-OmitStackTraceInFastThrow
12 -Djdk.attach.allowAttachSelf=true
13 -Dkotlinx.coroutines.debug=off
14 -Djdk.moduleillegalAccess.silent=true
15 -javaagent:C:\Users\Public\jetbrains\jetbrains-agent-v3.2.1.c46b.ed7=by https://zhile.io
16 -XX:MaxMetaspaceSize=1024m
17 -Dfile.encoding=UTF-8
```

部署前的准备：配置Tomcat

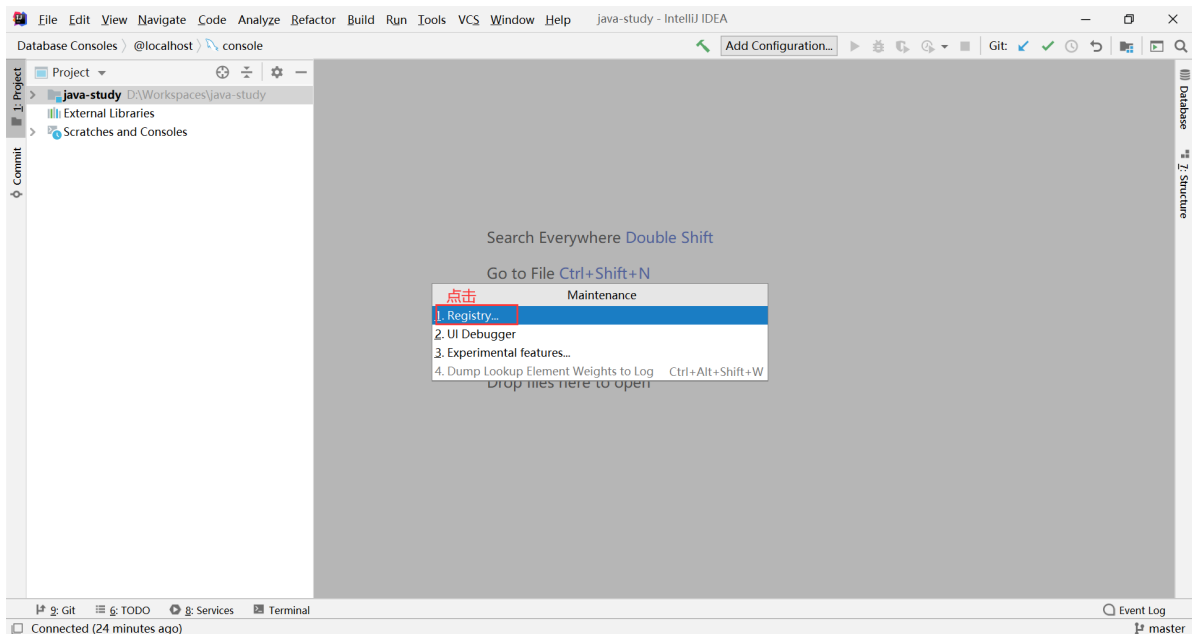




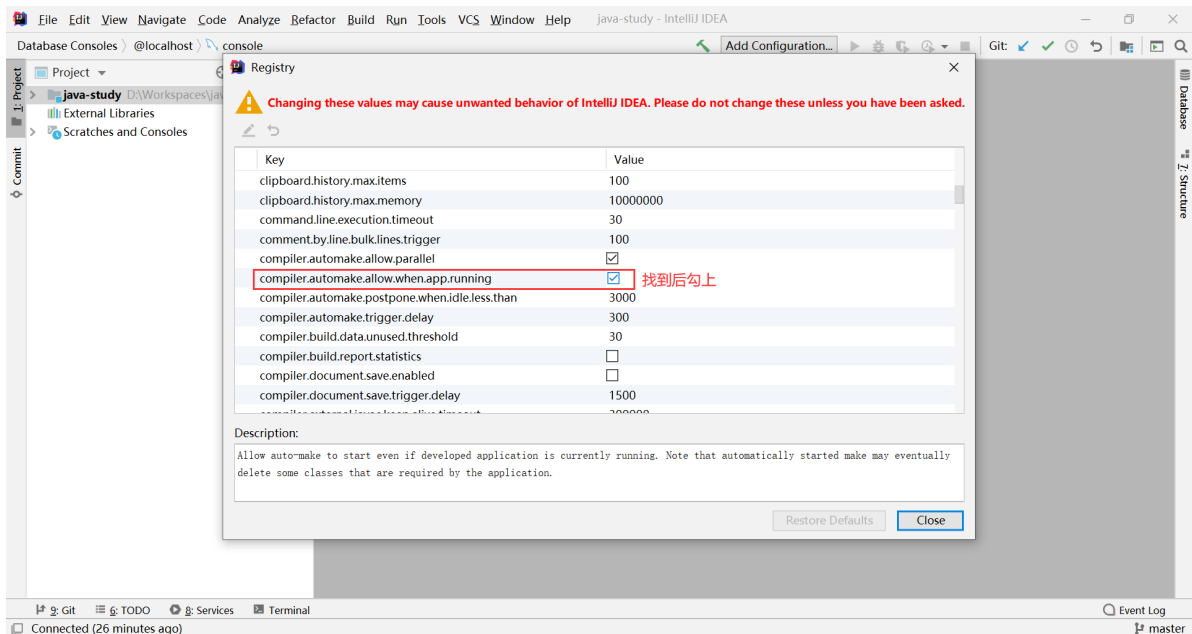
部署前的准备：Web项目运行时自动部署

为了让Web项目自动部署，需要如下配置：

点击 shift+ctrl+alt+/ 键，在弹出窗中点击Registry



在弹出的Registry选项中配置：



部署Web项目

在之前配置好的Tomcat中部署：Run -> Edit Configurations

