

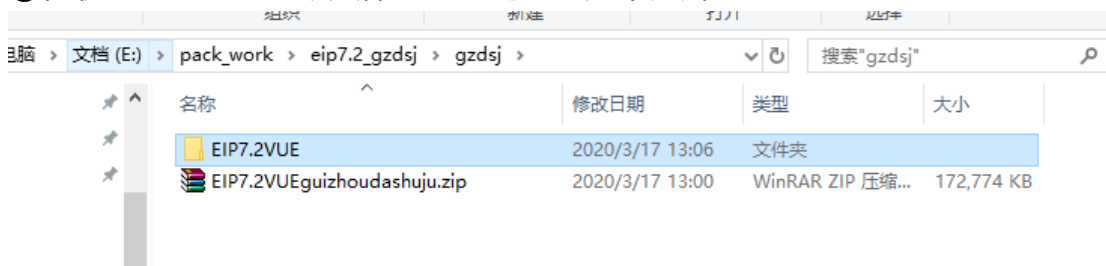
EIP7.2 开发环境单服务部署

(Eclipse 版)

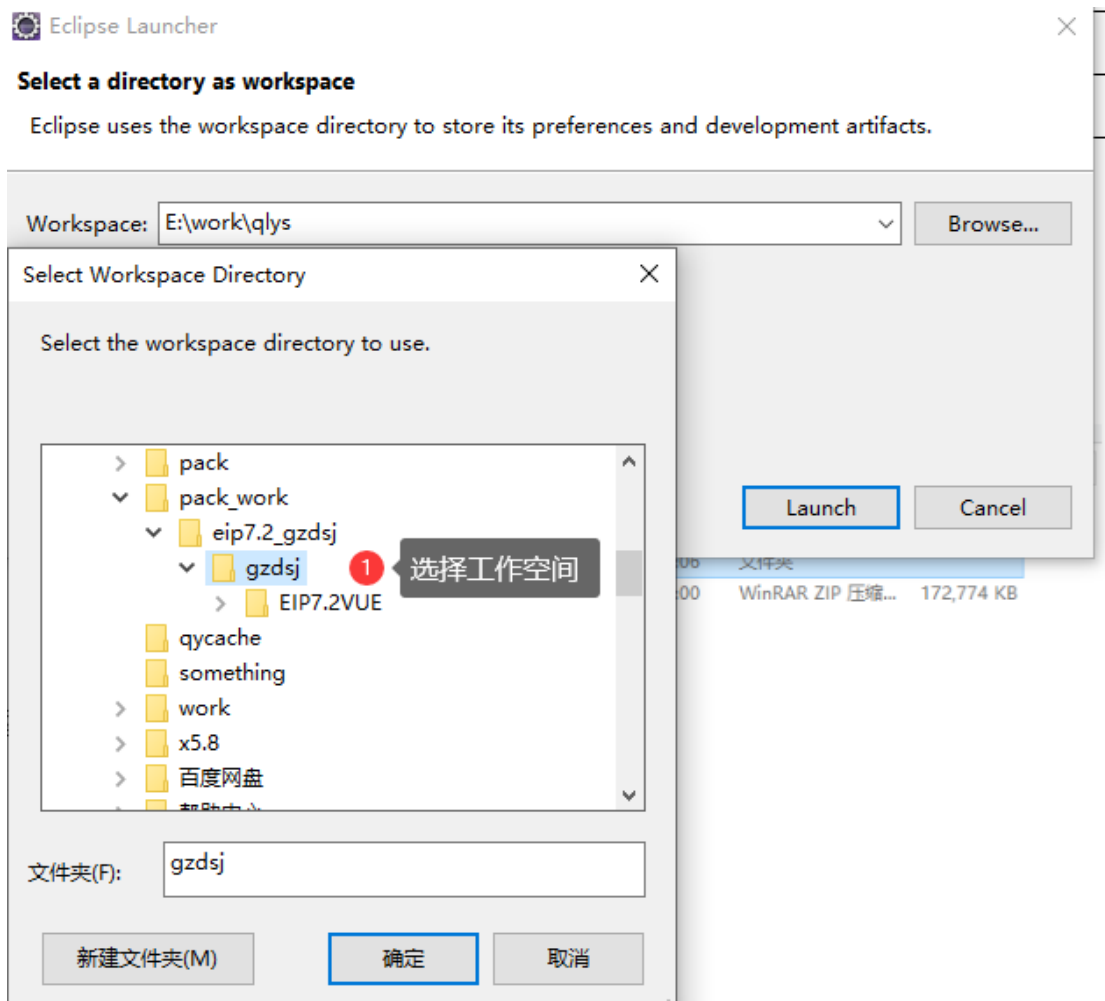
运行环境	JDK1.8
数据库	Mysql5.6 或 Mysql5.7 ORACLE 10 或 11G
操作系统	Windows server 或 linux。
应用服务器	Tomcat7.0.62 或 Tomcat8, nginx
中间件	ActiveMQ, Redis
浏览器	Google Chrome, Firefox 等等

1. 工作空间准备

①将收到的源码文件放具体工作空间文件夹中并解压

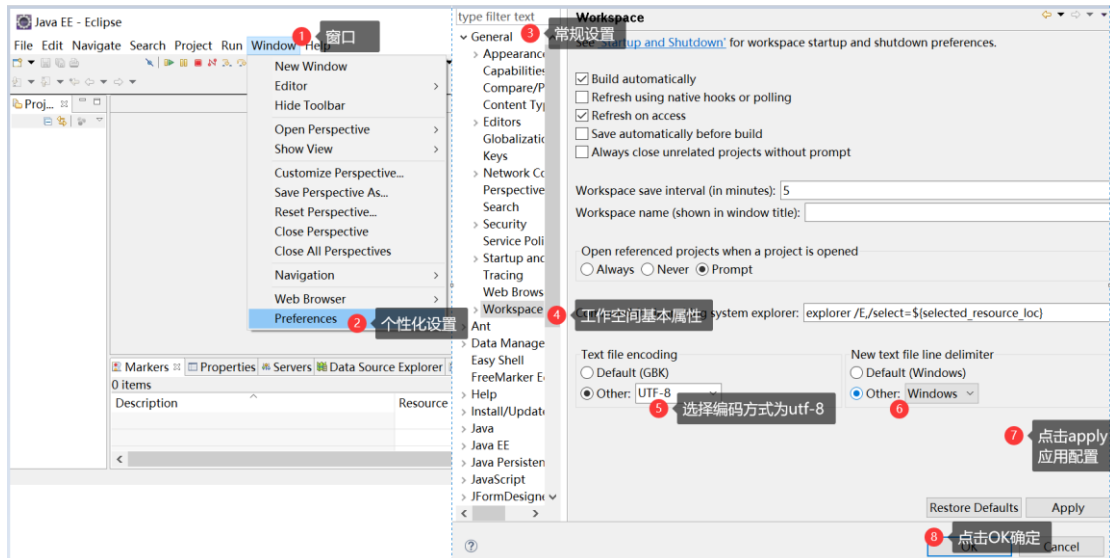


②启动 eclipse

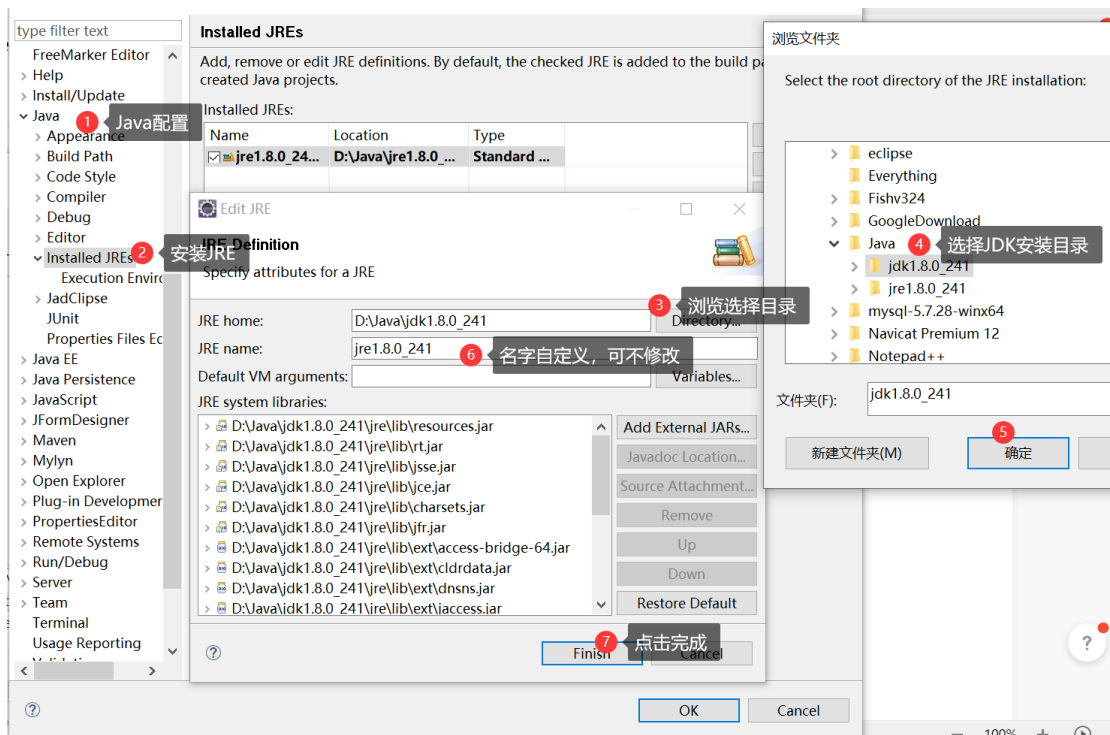


2. Eclipse 基本配置

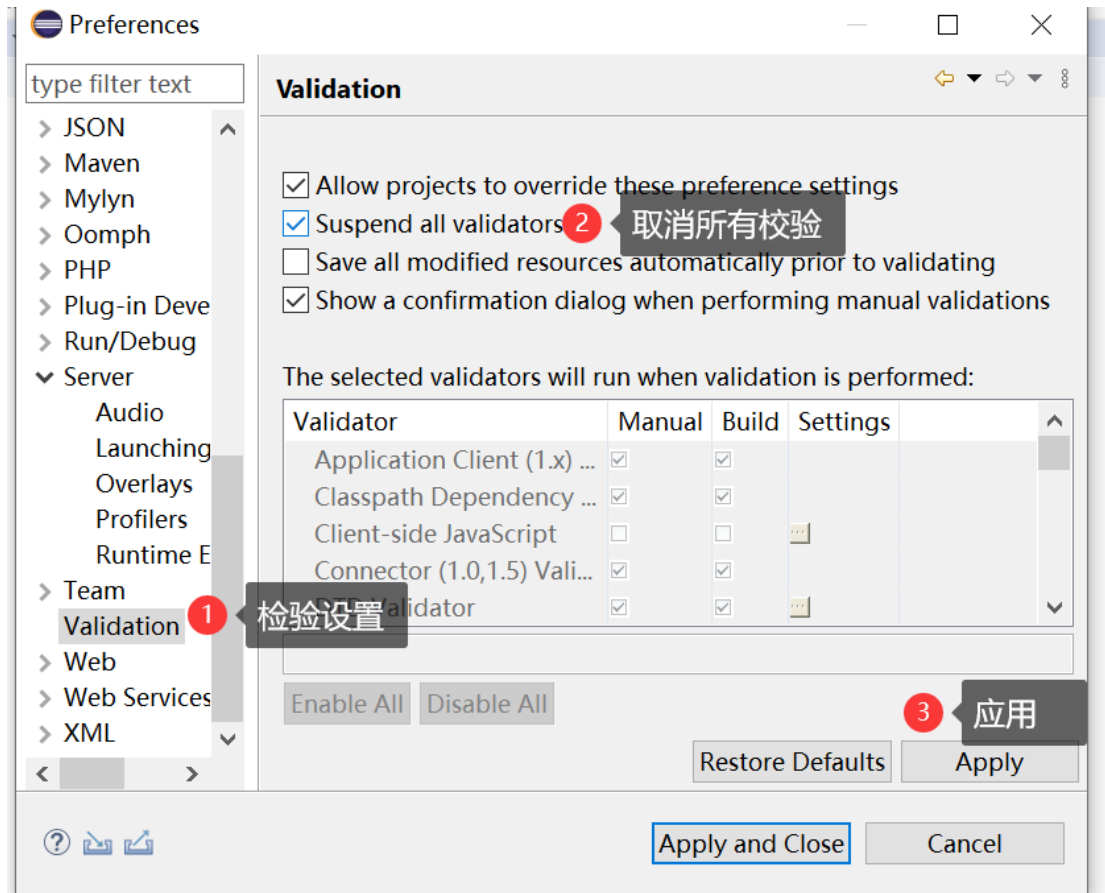
①配置编码规则



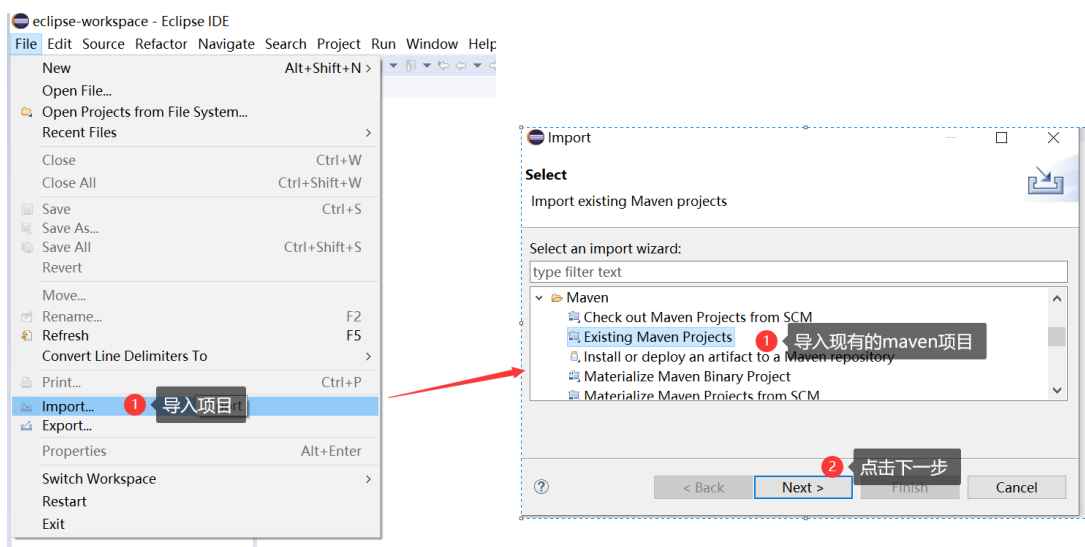
②配置 JAVA 环境

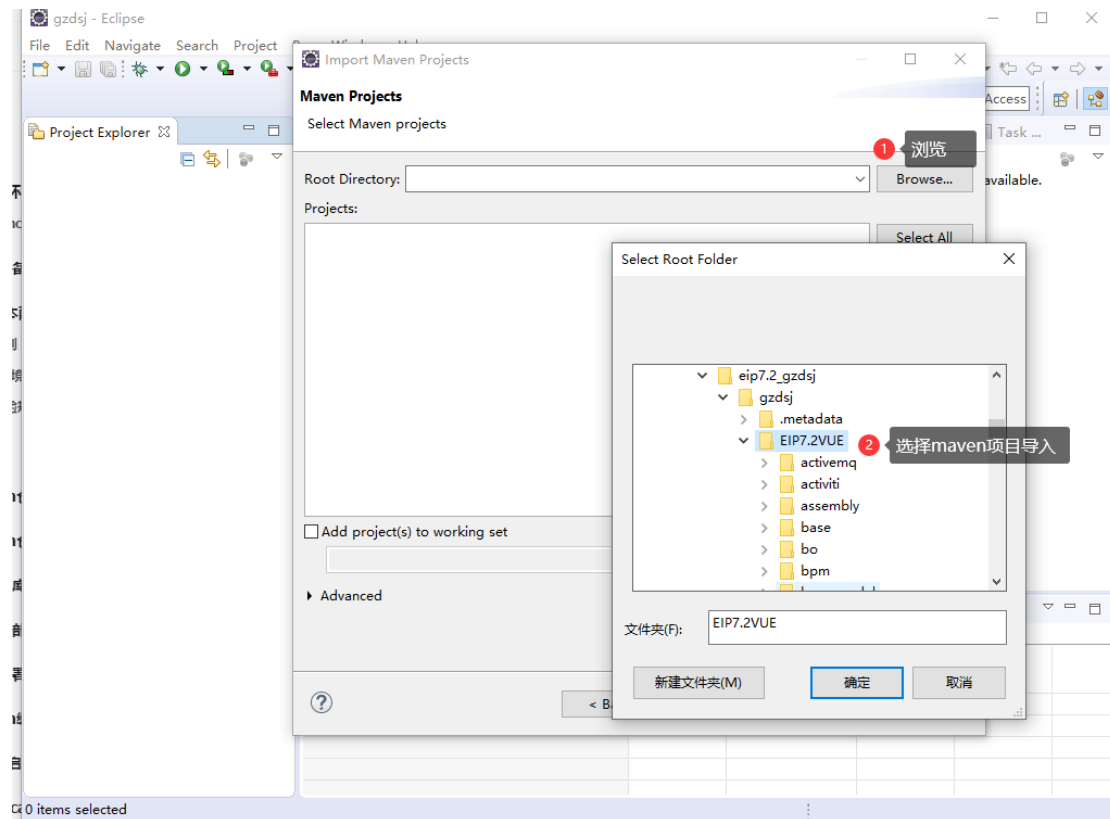


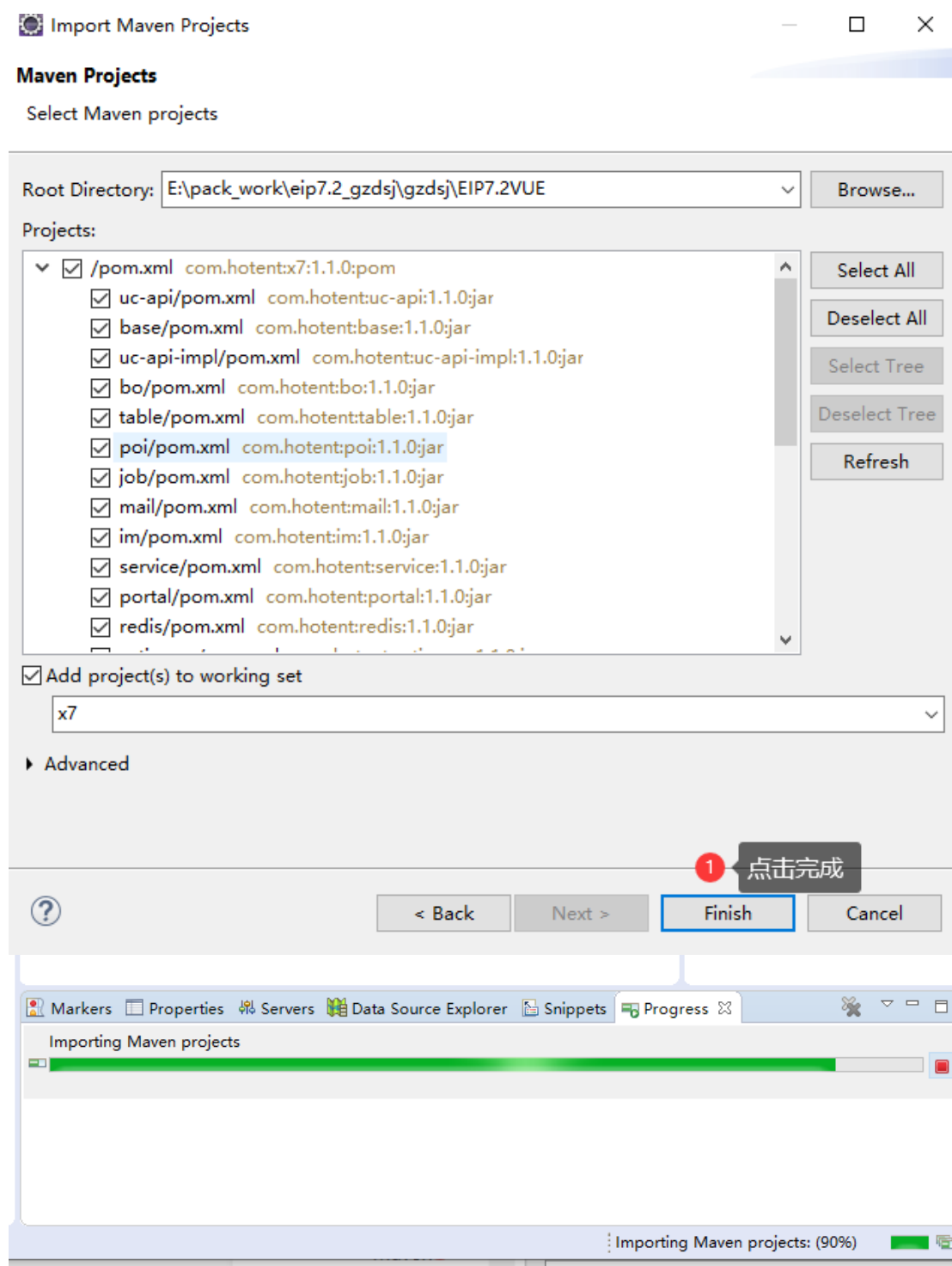
③去掉所有校验规则



3. 导入项目

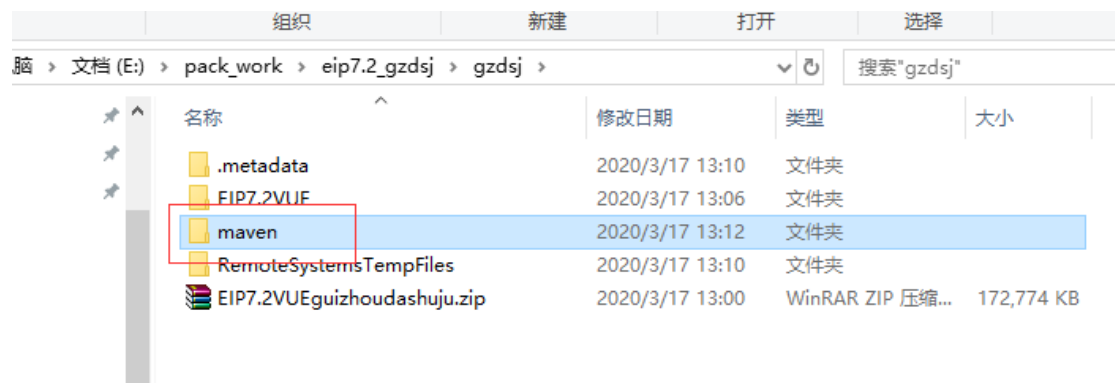




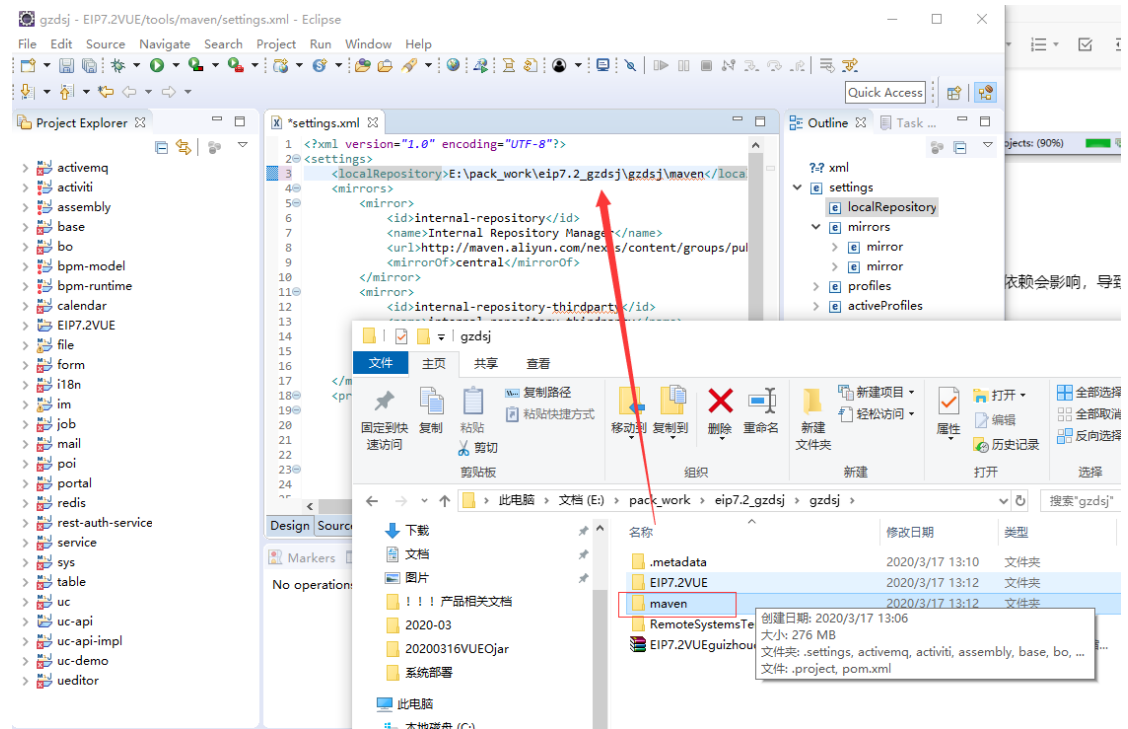


4. 修改 maven 仓库

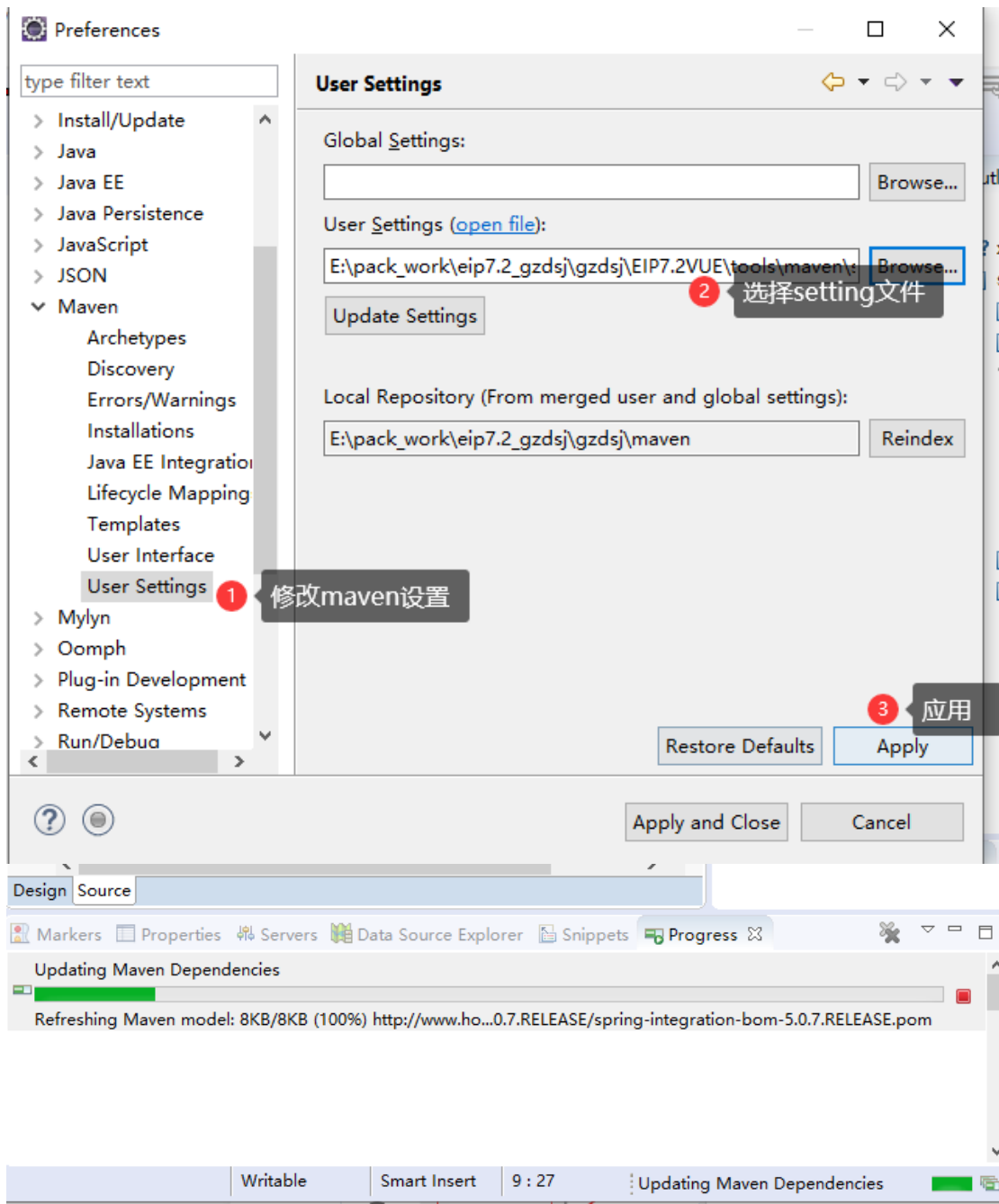
建议每个项目都新建一个文件夹作为 maven 仓库，以免不同项目的依赖会影响，导致后续问题产生。



修改 maven 的 settings.xml 文件



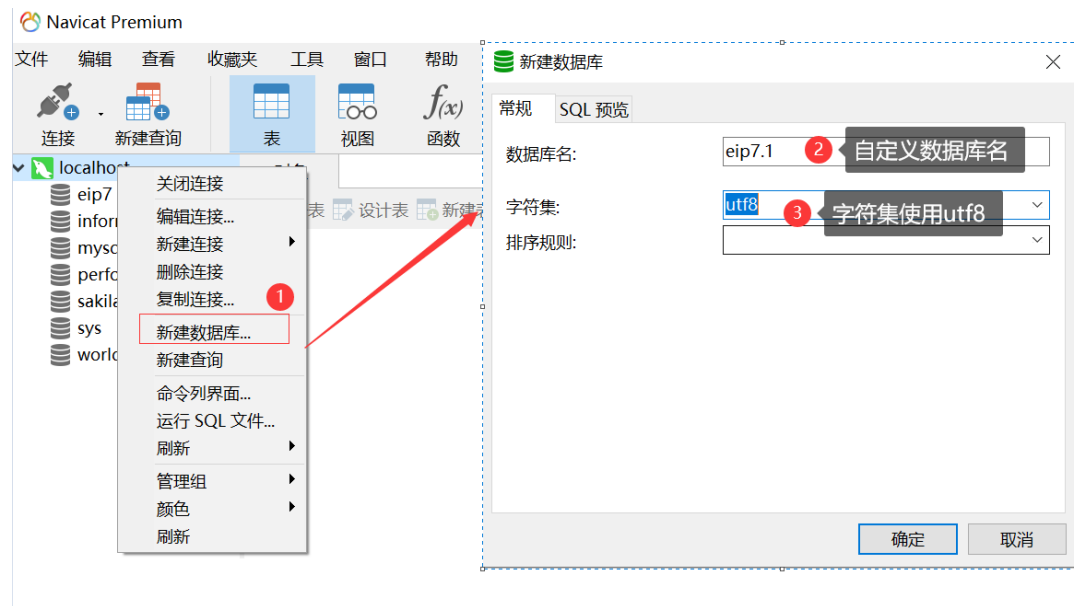
5. 配置 maven 仓库



6. 初始化数据库

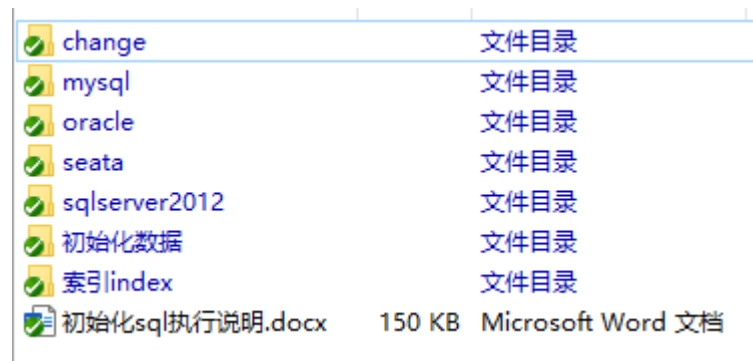
在我们提供的源码文件中找到 sql 文件夹，对应使用的数据库进行数据初始化，这里我用的时 mysql 数据库，选择 mysql 文件夹中的初始化脚本执

行，导入可使用数据库可视化工具协助。



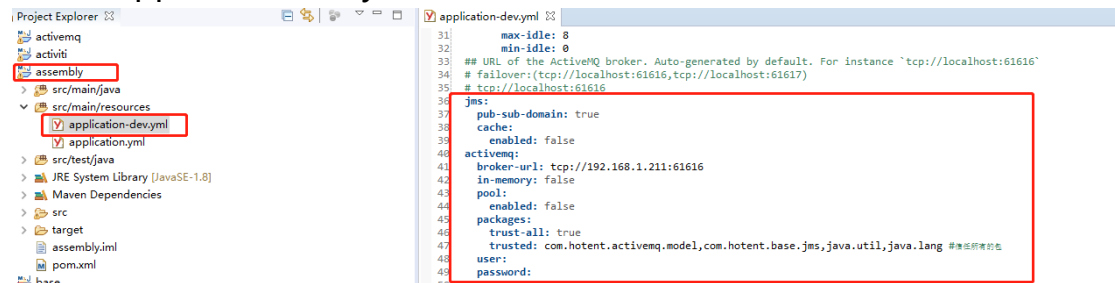
执行 sql 文件：

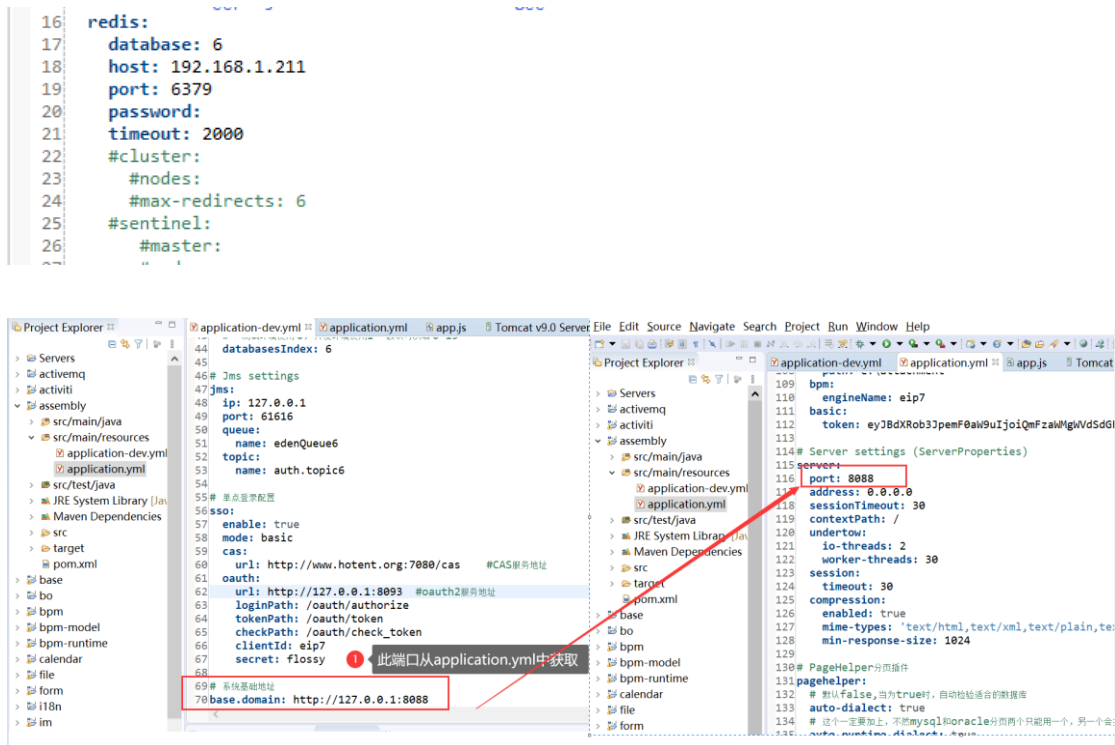
具体步骤请查看 [初始化 sql 执行说明 文档](#)



7. 修改单服务部署配置文件

①修改 application-dev.yml 文件，配置中间件信息及系统基础地址

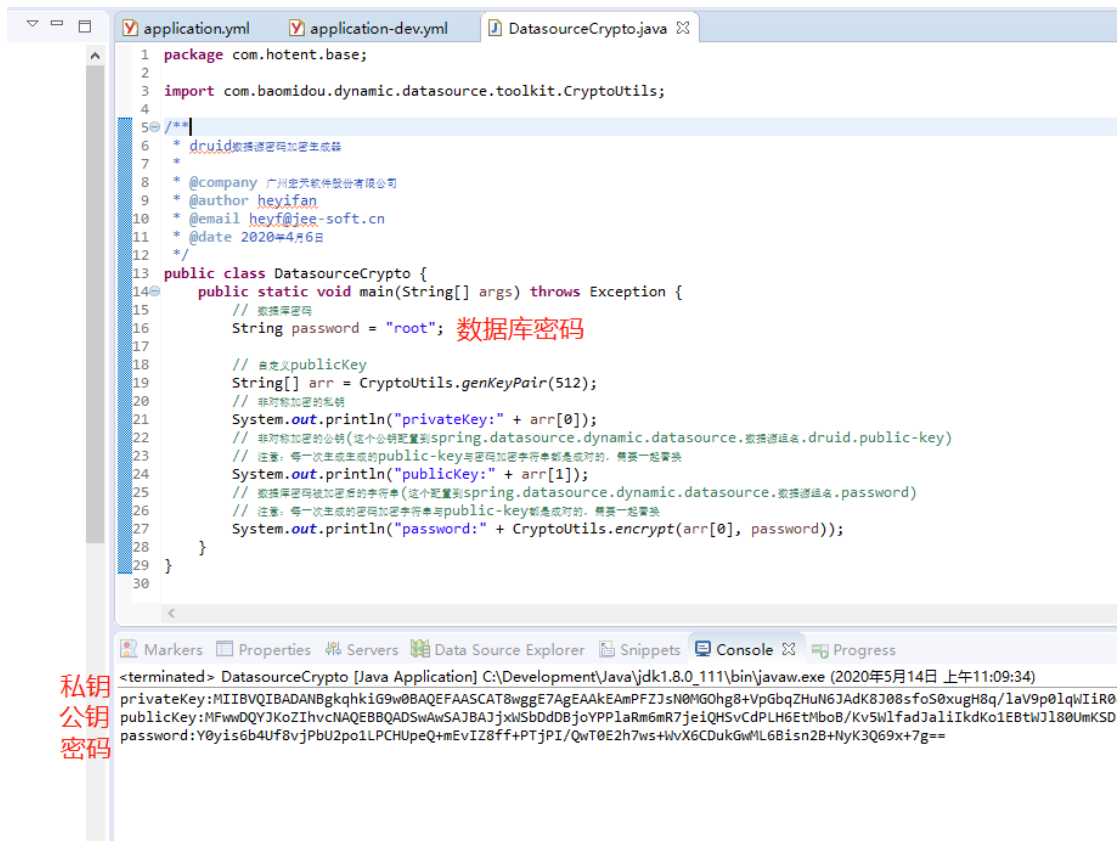




②修改 application.yml 文件，配置数据库信息



数据源连接需要使用数据，通过 DataSourceCrypto.java 类生成数据源的密码
加密数据及公钥数据



```

1 package com.hotent.base;
2
3 import com.baomidou.dynamic.datasource.toolkit.CryptoUtils;
4
5 /**
6  * druid数据库密码加密生成器
7  *
8  * @company 广州宏天软件股份有限公司
9  * @author heyifan
10  * @email heyifan@jag-soft.cn
11  * @date 2020年4月6日
12  */
13 public class DatasourceCrypto {
14     public static void main(String[] args) throws Exception {
15         // 数据库密码
16         String password = "root"; 数据库密码
17
18         // 自定义PublicKey
19         String[] arr = CryptoUtils.genKeyPair(512);
20         // 非对称加密的私钥
21         System.out.println("privateKey:" + arr[0]);
22         // 非对称加密的公钥(这个公钥配置到spring.datasource.dynamic.datasource.数据库连接.druid.public-key)
23         // 注意: 每一次生成生成的public-key与密码加密字符串都是成对的, 需要一起替换
24         System.out.println("publicKey:" + arr[1]);
25         // 数据库密码被加密后的字符串(这个配置到spring.datasource.dynamic.datasource.数据库连接.password)
26         // 注意: 每一次生成的密码加密字符串与public-key都是成对的, 需要一起替换
27         System.out.println("password:" + CryptoUtils.encrypt(arr[0], password));
28     }
29 }
30

```

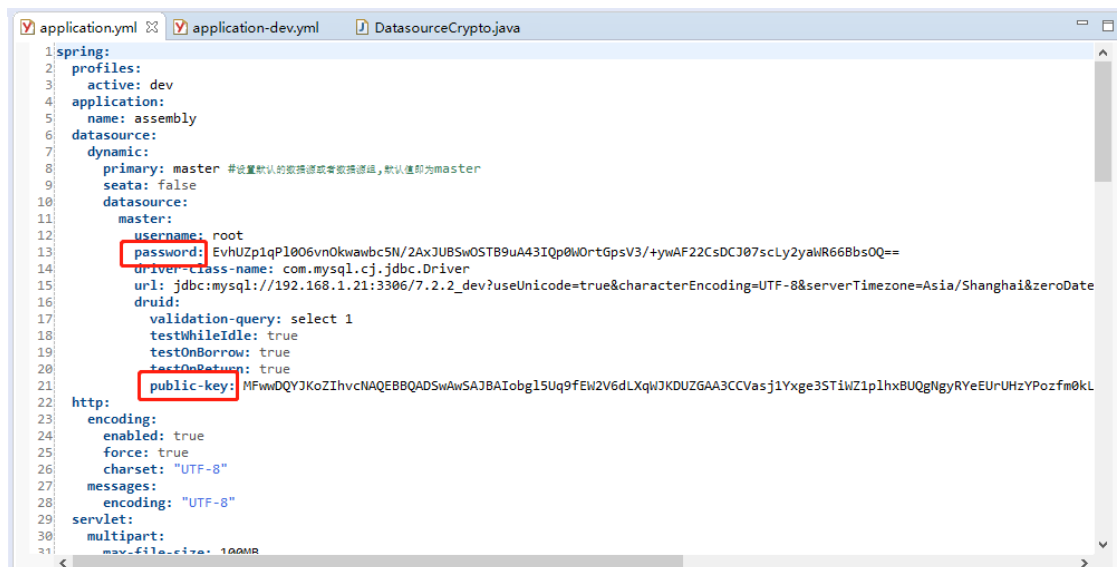
私钥
公钥
密码

```

<terminated> DatasourceCrypto [Java Application] C:\Development\Java\jdk1.8.0_111\bin\javaw.exe (2020年5月14日 上午11:09:34)
privateKey:MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAKEAmPFZ3sN0MG0hg8+Vp6bqZHuN6AdK8J08sfoS0xugh8q/1aV9p0lqWIIr0
publicKey:MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAljxwSbDdDBjoYPP1aRm6mR7jeiQHSvCdPLH6EtMboB/Kv5W1fadJa1iIkKo1EBtWJ180UmKSD
password:Y0yis6b4Uf8vjPbU2po1LPCHuPeQ+mEvIZ8ff+PTjPI/QwT0E2h7ws+lvX6CDukGwML6Bisn2B+NyK3Q69x+7g==

```

将公钥及密码替换至配置文件



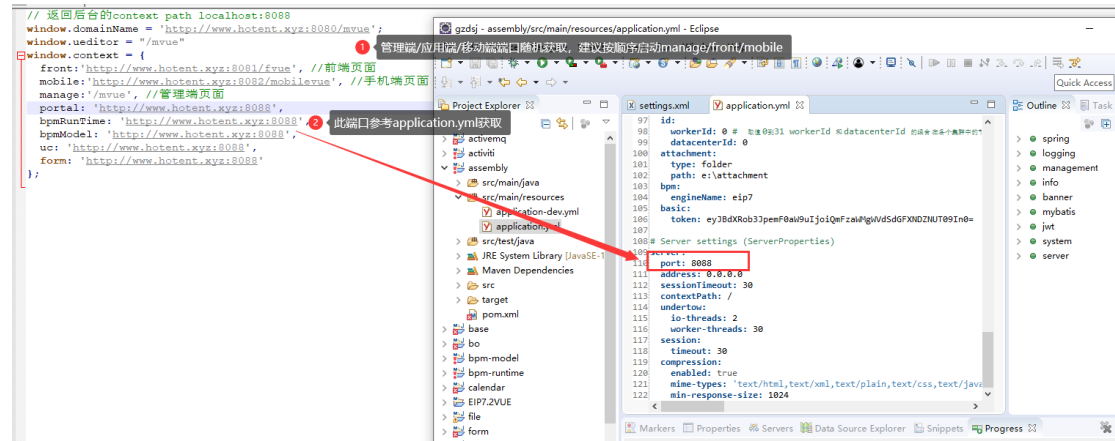
```

1 spring:
2   profiles:
3     active: dev
4   application:
5     name: assembly
6   datasource:
7     dynamic:
8       primary: master #设置默认的数据源或者数据源组, 默认值为master
9     seata: false
10    datasource:
11      master:
12        username: root
13        password: EvhUZp1qP1006vn0kwawbc5N/2AxJUBSw0STB9uA43IQp0W0rtGpsV3/+yWAF22CsDCJ07scLy2yahR66BbsOQ==
14        driver-class-name: com.mysql.cj.jdbc.Driver
15        url: jdbc:mysql://192.168.1.21:3306/7.2.2_dev?useUnicode=true&characterEncoding=UTF-8&serverTimezone=Asia/Shanghai&zeroDate
16        druid:
17          validation-query: select 1
18          testWhileIdle: true
19          testOnBorrow: true
20          testOnReturn: true
21          public-key: MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAljxwSbDdDBjoYPP1aRm6mR7jeiQHSvCdPLH6EtMboB/Kv5W1fadJa1iIkKo1EBtWJ180UmKSD
22      http:
23        encoding:
24          enabled: true
25          force: true
26          charset: "UTF-8"
27        messages:
28          encoding: "UTF-8"
29      servlet:
30        multipart:
31          max-file-size: 100MB

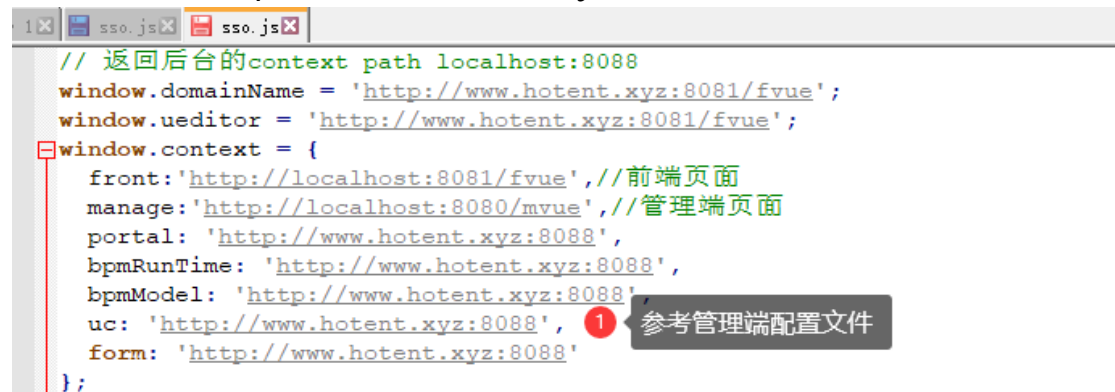
```

8. 修改前端部署配置文件

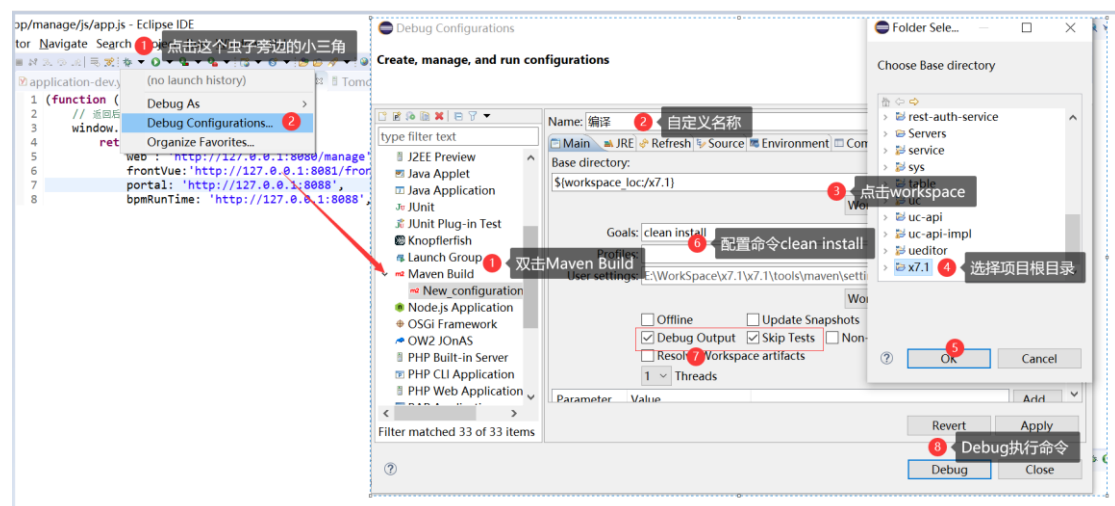
①...\web\manage\public 路径下找到 sso.js 文件, 编辑修改



②....\web\front\public 路径下找到 sso.js 文件，编辑修改



9. 配置 maven 编译命令



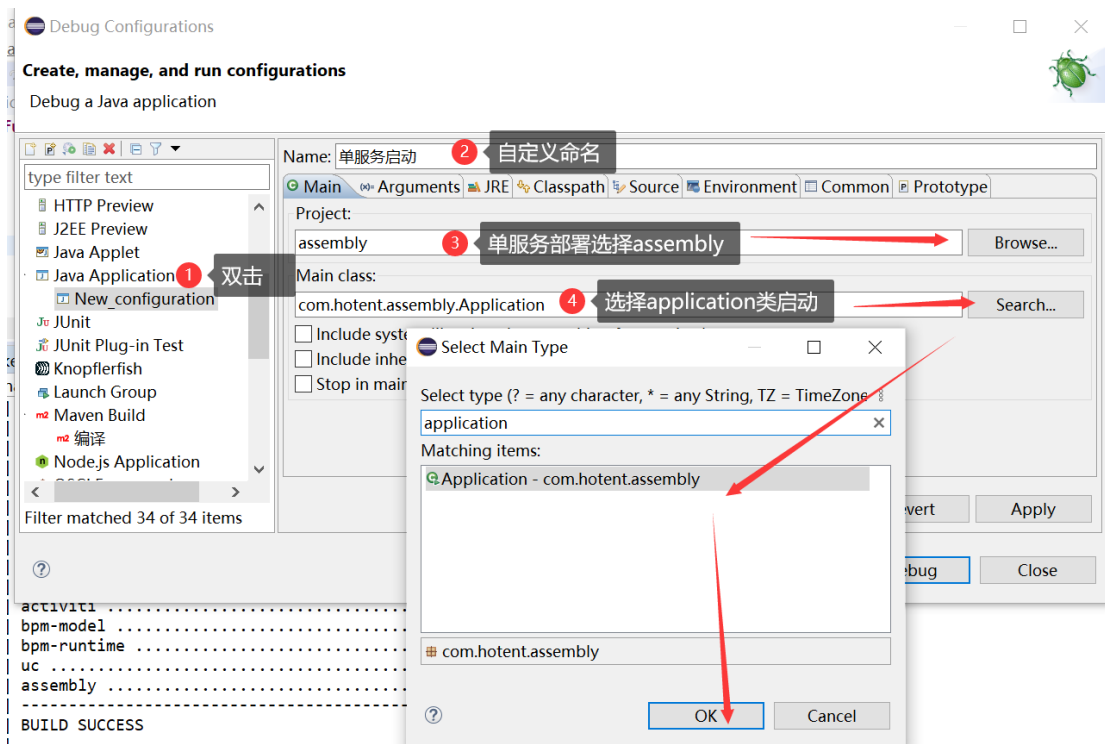
控制台可查看编译日志，如有报错，请排查错误原因。

```

Markers Properties Servers Data Source Explorer Snippets Console Progress
<terminated> 编译 [Maven Build] D:\Java\jdk1.8.0_241\bin\javaw.exe (2020年3月17日 上午12:11:06)
[INFO] mail ..... SUCCESS [ 2.577 s]
[INFO] im ..... SUCCESS [ 1.778 s]
[INFO] service ..... SUCCESS [ 2.109 s]
[INFO] activemq ..... SUCCESS [ 1.208 s]
[INFO] calendar ..... SUCCESS [ 2.244 s]
[INFO] ueditor ..... SUCCESS [ 1.517 s]
[INFO] portal ..... SUCCESS [ 5.250 s]
[INFO] rest-auth-service ..... SUCCESS [ 0.766 s]
[INFO] form ..... SUCCESS [ 6.780 s]
[INFO] bpm ..... SUCCESS [ 23.011 s]
[INFO] activiti ..... SUCCESS [ 8.306 s]
[INFO] bpm-model ..... SUCCESS [ 2.667 s]
[INFO] bpm-runtime ..... SUCCESS [ 5.730 s]
[INFO] uc ..... SUCCESS [ 11.548 s]
[INFO] assembly ..... SUCCESS [02:48 min]
[INFO] -----
[INFO] BUILD SUCCESS ① 编译成功
[INFO] -----
[INFO] Total time: 05:09 min
[INFO] Finished at: 2020-03-17T00:16:21+08:00
[INFO] -----

```

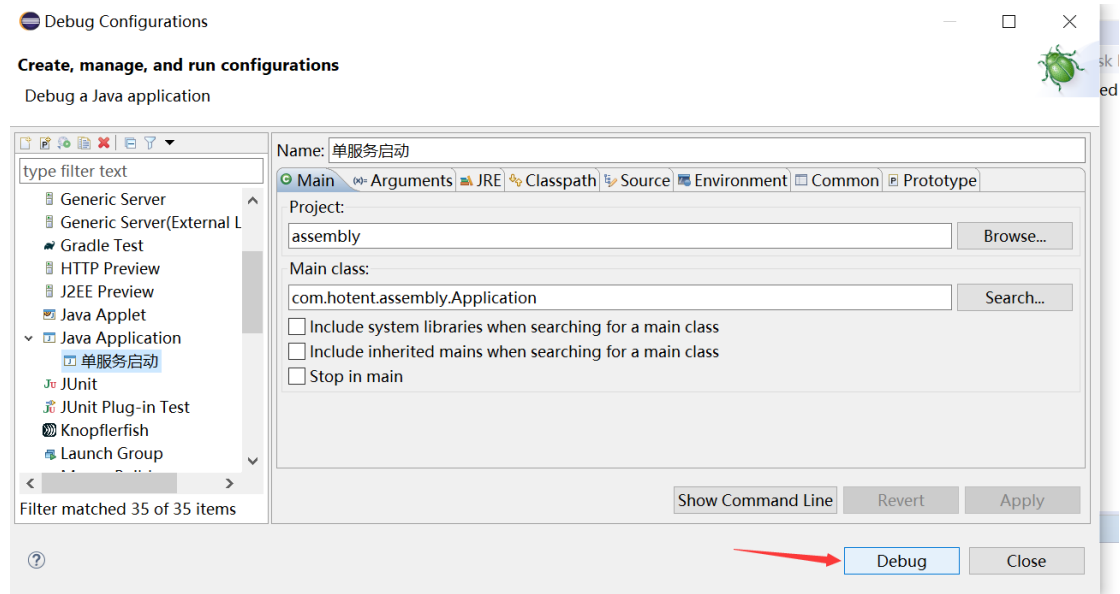
10. 配置服务启动命令



11. 启动项目

①启动服务

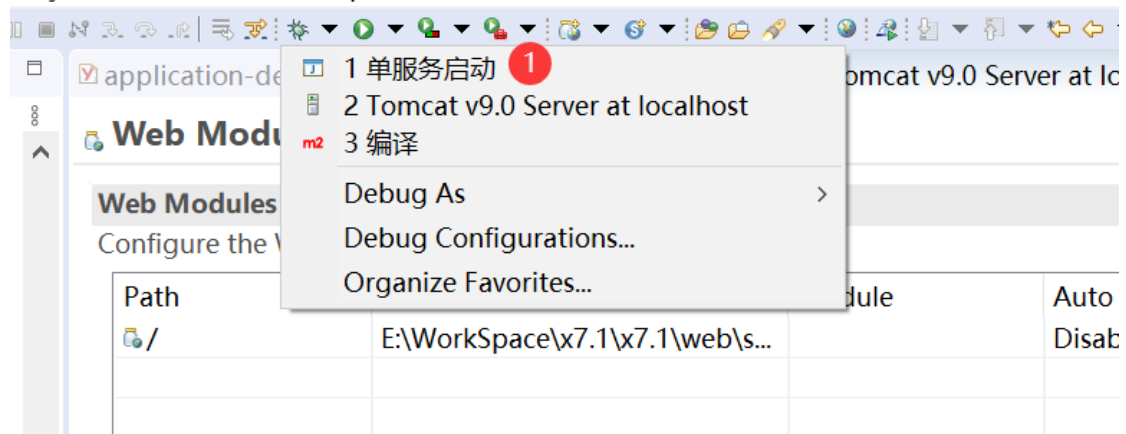
注：需保证 ActiveMQ\redis 等中间件都已启动



后续可通过小虫子启动

t localhost - Eclipse IDE

Project Run Window Help



③启动前端

注：提前安装 node.js，网上搜索 node.js 进入官网，通常下载稳定版，下载安装即可，安装成功后继续以下操作

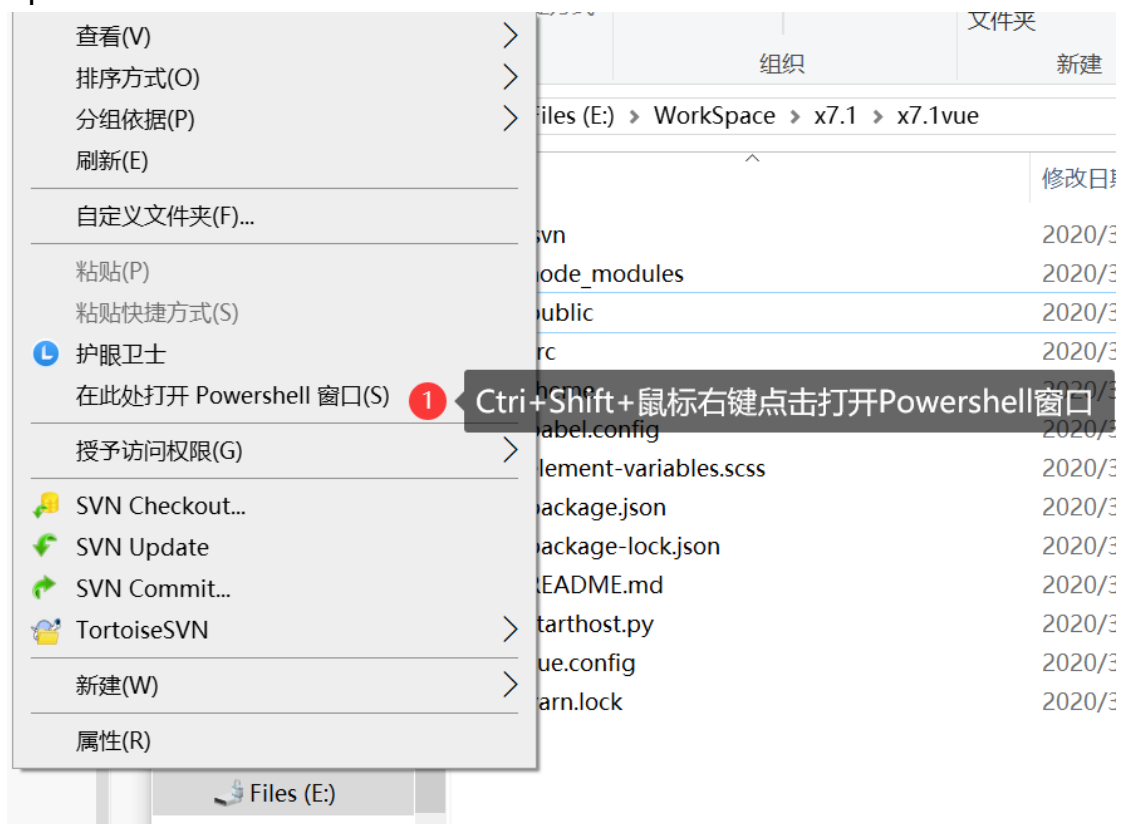


进入前端文件夹，Ctrl+Shift+鼠标右键进入命令窗口，依次输入以下命令：

npm install

npm run build

npm run serve



④浏览器访问

到达这一步，你就部署成功啦，现在试试浏览器访问看看。