**部署手册**

1. JDK安装
   1. Windows
2. 安装文件版本

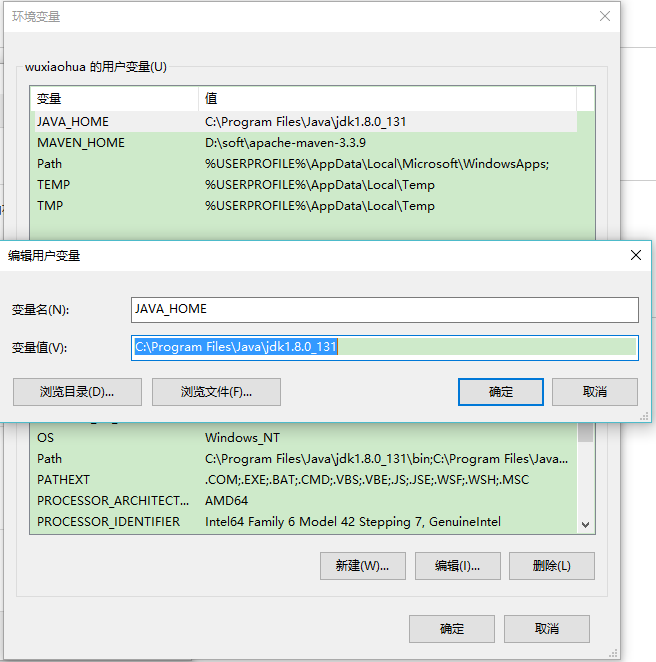


1. 安装步骤（略）

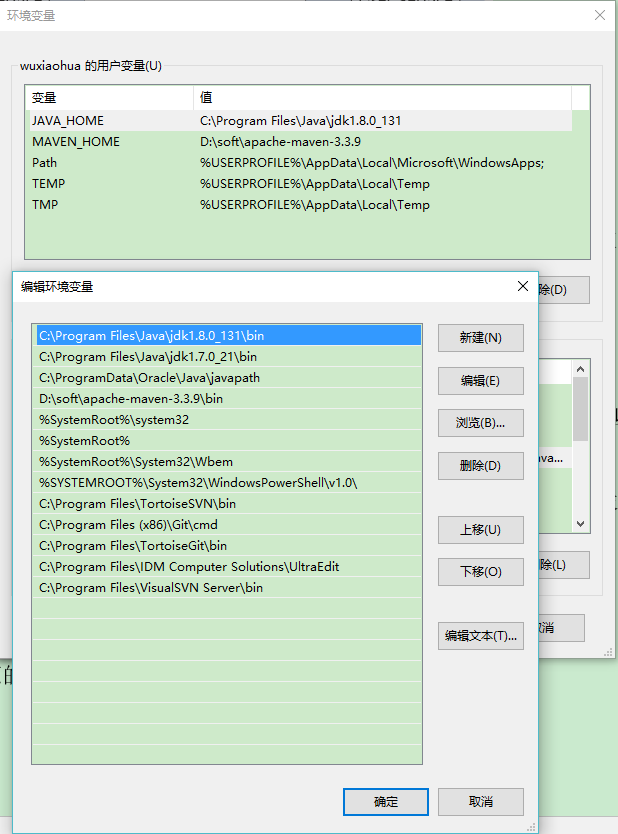
直接点击运行安装文件即可

1. 环境变量

配置JAVA\_HOME



配置PATH



* 1. Linux

1. 安装版本



1. 安装步骤

下载JDK（Linux版）安装文件：jdk-7u21-linux-x64.tar.gz，拷贝到Linux Server目录：/opt

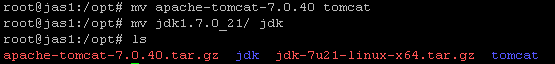


解压jdk-7u21-linux-x64.tar.gz至当前目录

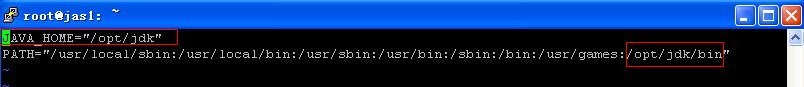




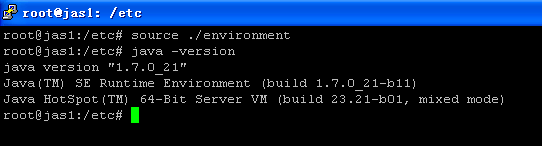
重命名解压文件jdk1.7.0\_21为jdk



配置JDK系统环境变量



验证JDK是否安装成功

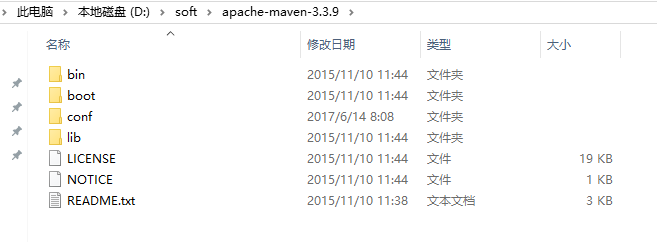


1. Maven安装（Windows）
2. 安装版本

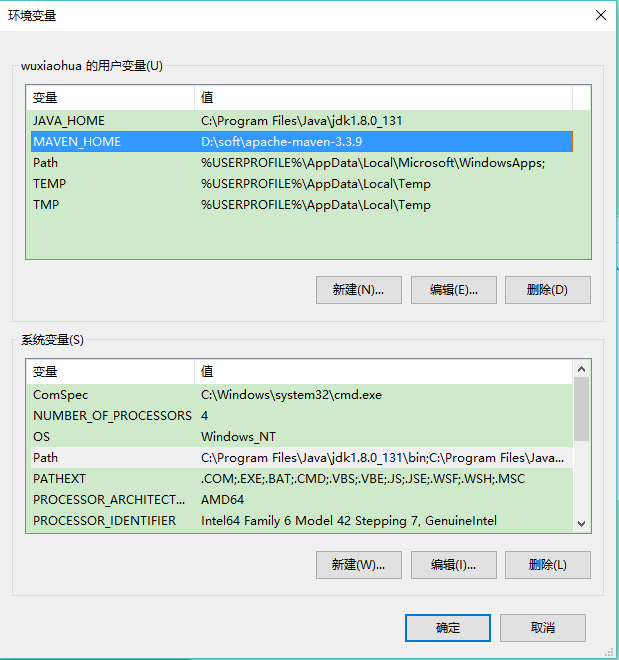


1. 安装步骤（略）

直接解压即可



1. 环境变量



1. 配置文件

Maven工具的默认配置文件路径MAVEN\_HOME/conf/settings.xml，一般需要对其内容进行修改，修改的内容如下

参考esf-document\settings\maven-conf\settings.xml

SVN获取该参考配置文件：

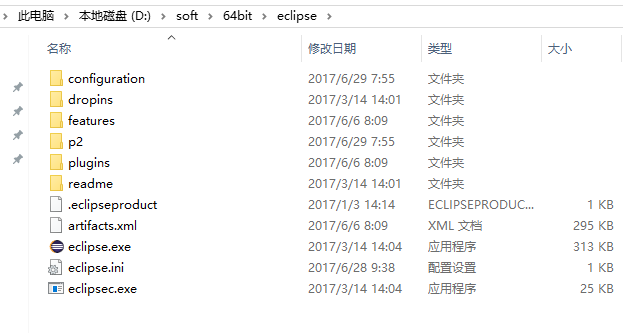
<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/settings/maven-conf/settings.xml>

1. Eclipse安装配置（Windows）
   1. 安装
2. 安装版本



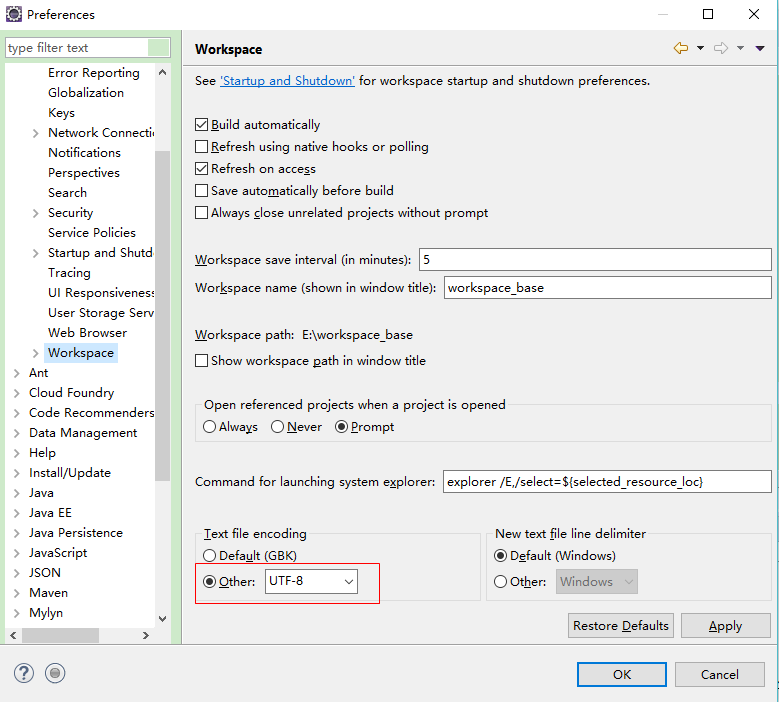
1. 安装步骤

直接解压即可



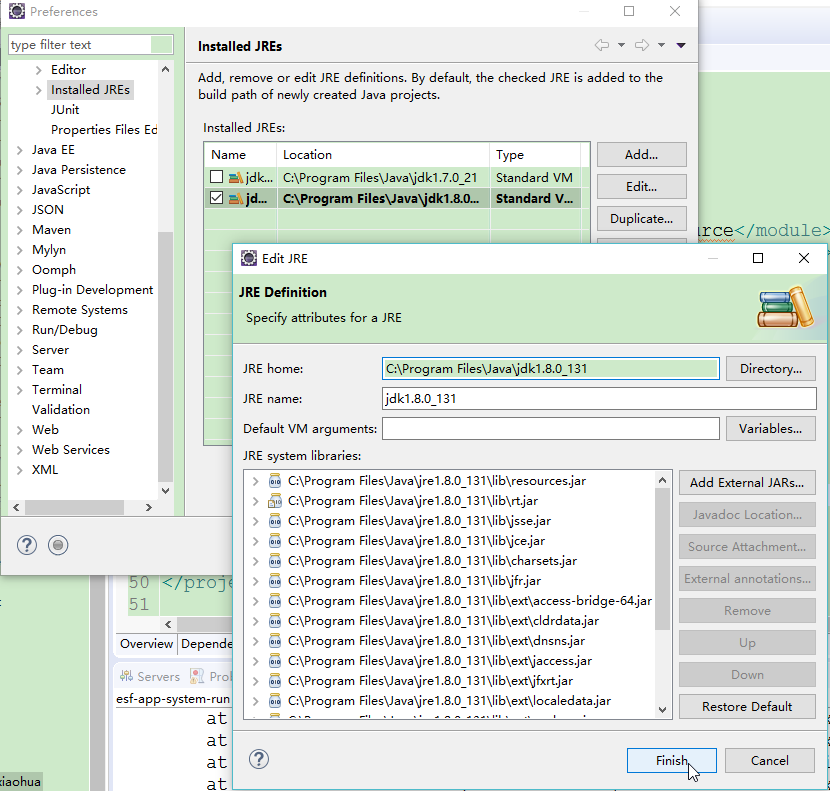
* 1. Eclipse配置字符集

【Window】【Preferences】【General】【Workspace】



* 1. Eclipse配置JDK

【Window】【Preferences】【Java】【Installed JREs】



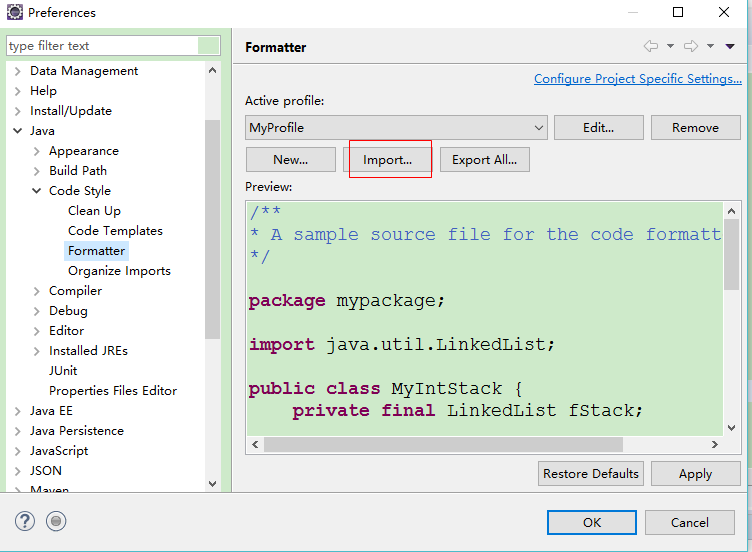
* 1. Eclipse配置格式化模板

【Window】【Preferences】【Java】【Code Style】【Formatter】【Import】

导入模板文件：esf-document\settings\eclipse\ide\_java\_style

SVN获取该文件：

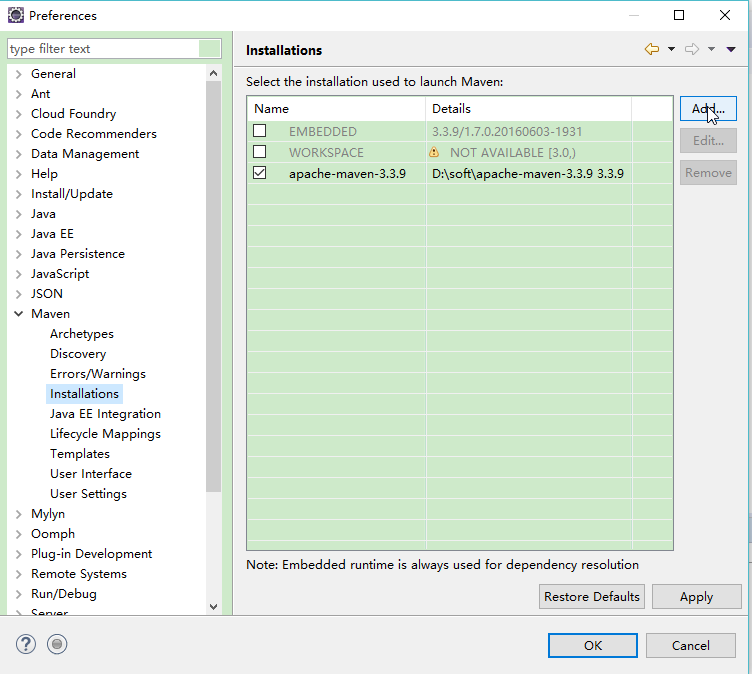
https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF\_CODE/ESF\_CLOUD/esf-document/settings/eclipse/ide\_java\_style



* 1. Eclipse配置Maven

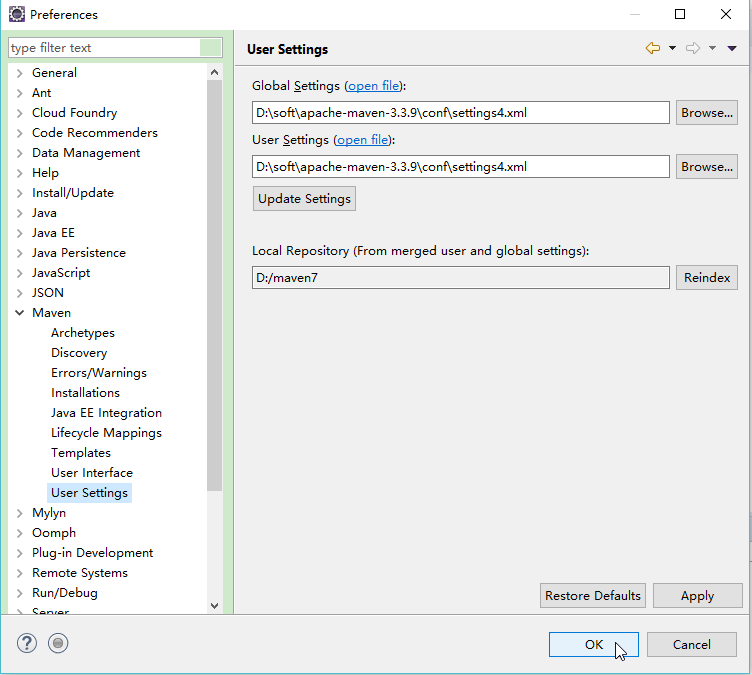
使用前面步骤安装过的Maven目录

【Window】【Preferences】【Maven】【Installations】



使用前面步骤安装过的Maven目录下的配置文件

【Window】【Preferences】【Maven】【User Settings】



配置文件可参考：

https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF\_CODE/ESF\_CLOUD/esf-document/settings/maven-conf/settings.xml

* 1. Eclipse配置lombok

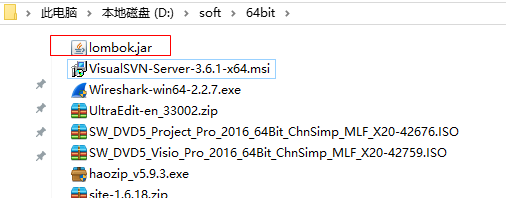
1. 插件版本

lombok 1.16.18：

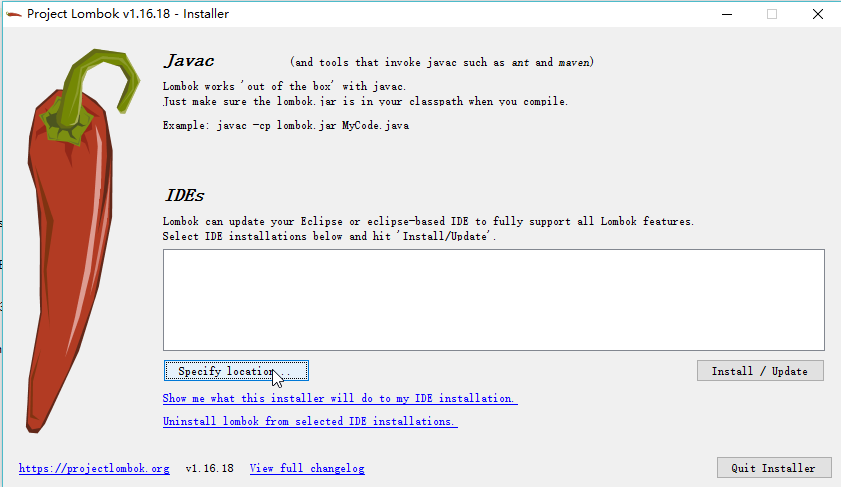
esf-document\libs\eclipse-plugins\lombok.jar

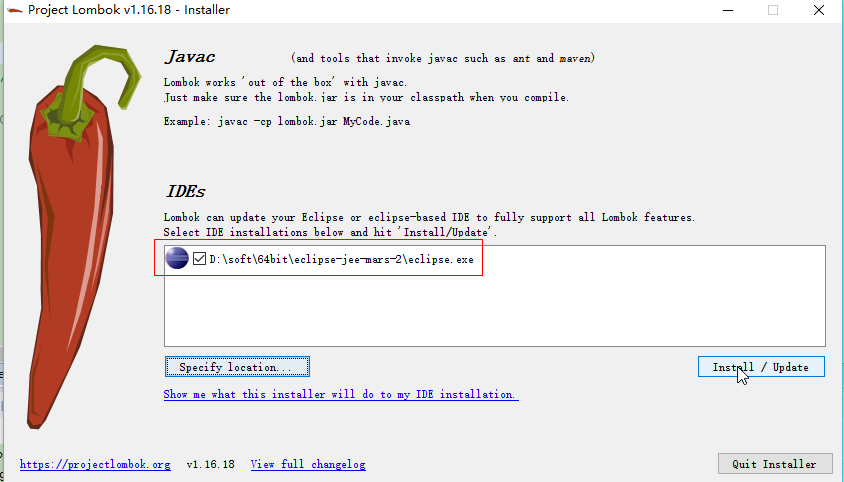
1. 安装步骤

双击lombok.jar运行jar文件

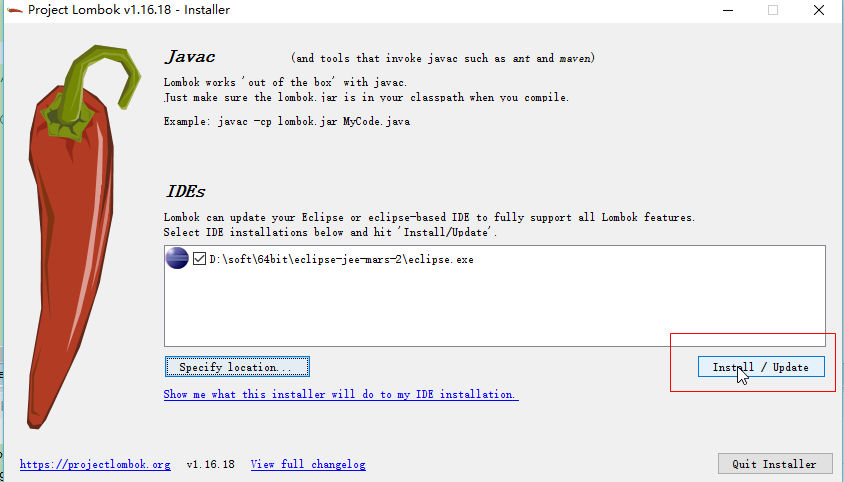


点击【Specify location】,设置为本地Eclipse解压目录

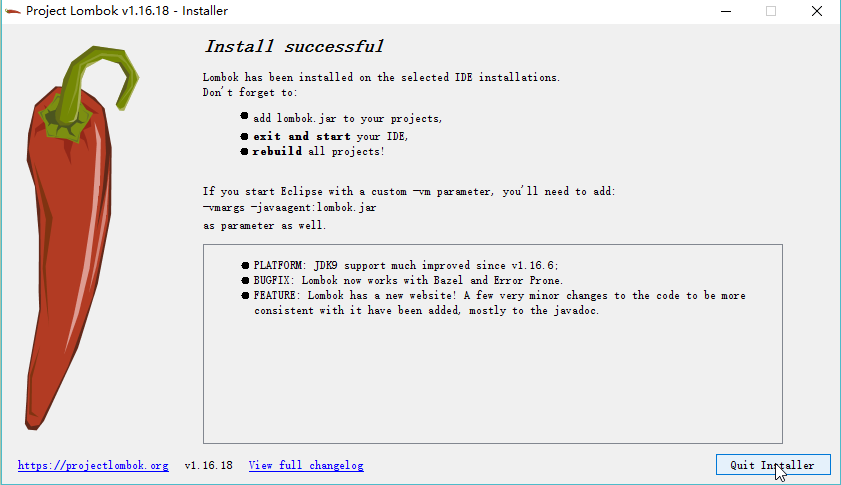




点击【Install/Update】



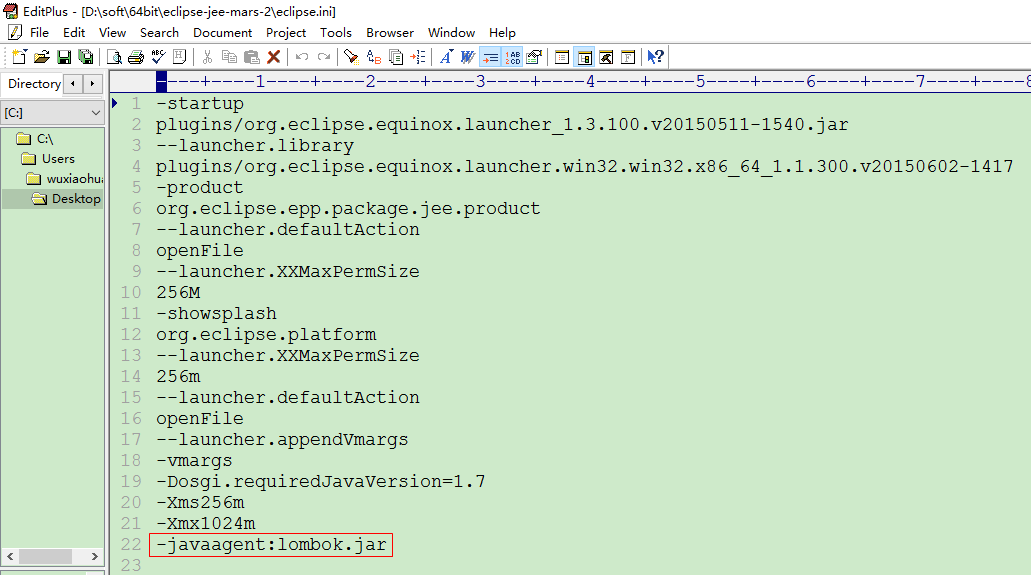
点击【Quit Installer】完成



回到本地Eclipse解压目录，安装成功会多出lombok.jar文件



【eclipse.ini】配置文件会添加如下配置



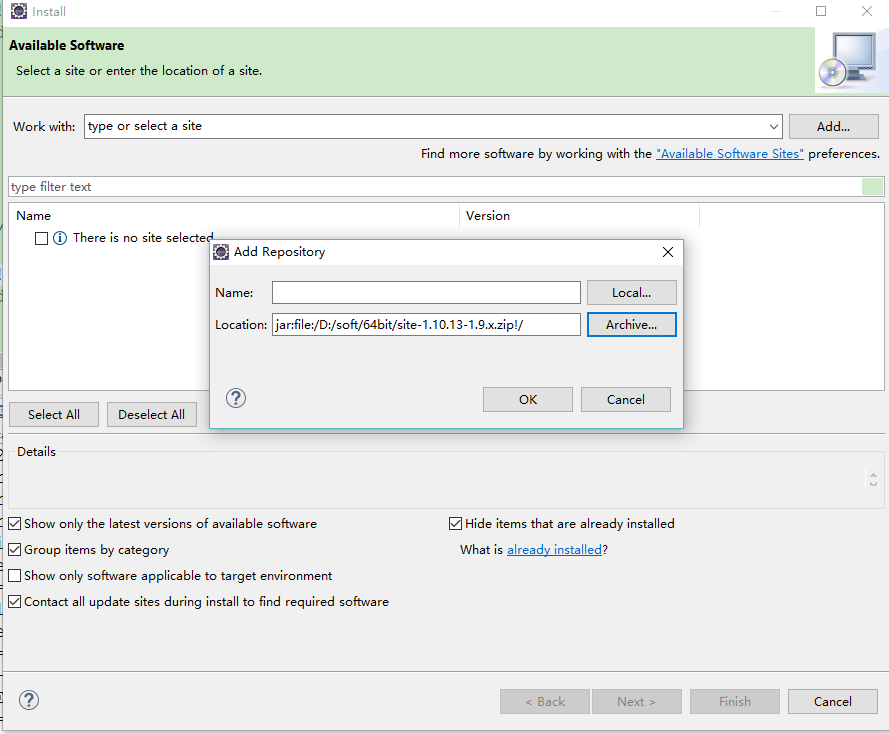
* 1. Eclipse配置SVN插件

1. 插件版本



1. 安装步骤

【Help】【Install New Software】【Add】



【Select All】【Next】即可

1. SVN客户端安装
2. 版本



1. 安装步骤

直接点击运行即可

1. 配置hosts
   1. Windows

修改文件C:\Windows\System32\drivers\etc\hosts

内容参考esf-document\settings\linux-etc-hosts\hosts

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/settings/linux-etc-hosts/hosts.txt>

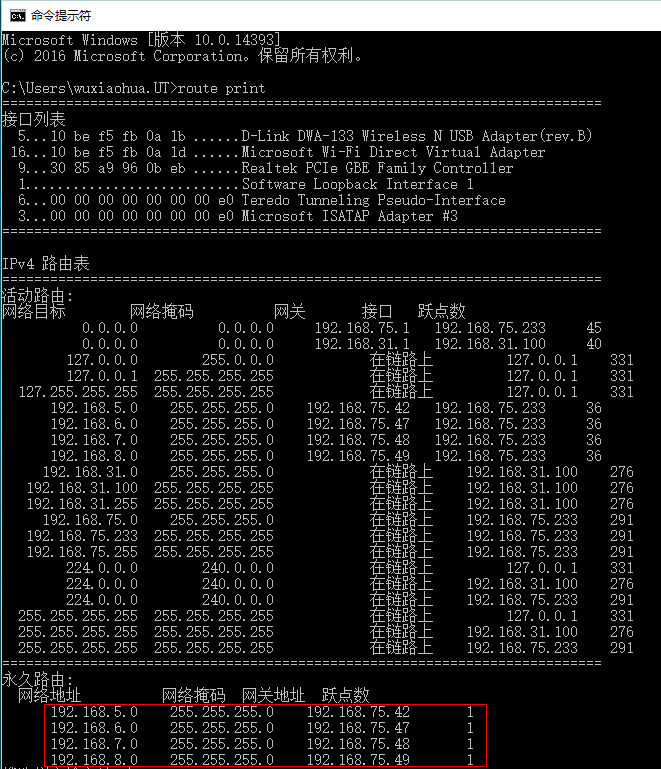
配置其他网段路由：

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/settings/linux-etc-hosts/routerun.bat>

以管理员身份双击运行【routerun.bat】

进入【cmd】【route print】查看路由是否添加成功



* 1. Linux

修改文件vi /etc/hosts

内容参考esf-document\settings\linux-etc-hosts\hosts

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/settings/linux-etc-hosts/hosts.txt>

1. 配置本地依赖库
   1. 加密库（可选）

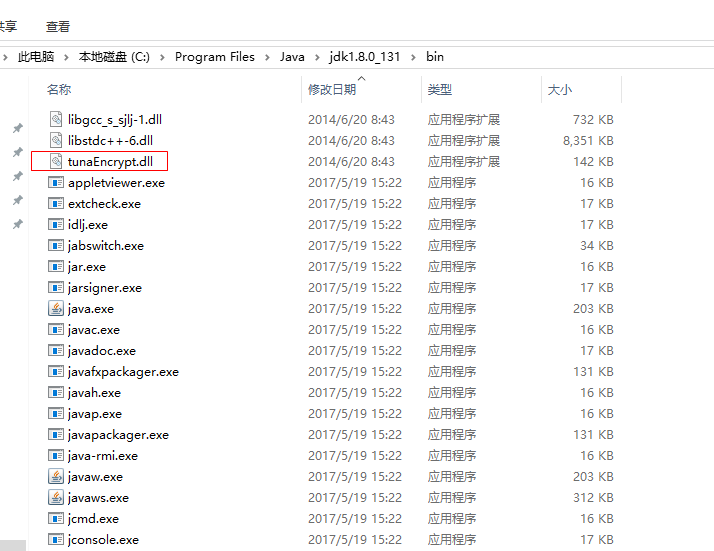
esf-encrypt项目依赖加密解密库

参考文件esf-document\libs\encrypt-libs

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/libs/encrypt-libs/win32>

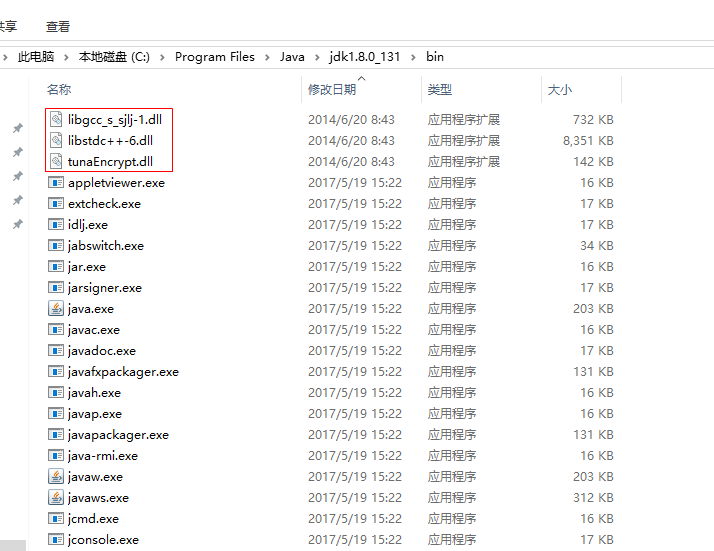
Win 32bit的配置，加密库放置JAVA\_HOME\bin路径



Win 64bit的配置，加密库放置JAVA\_HOME\bin路径

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/libs/encrypt-libs/win64>



Linux 64bit的配置，加密库放置/lib路径

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/libs/encrypt-libs/Linux64>



* 1. ffmpeg（可选）

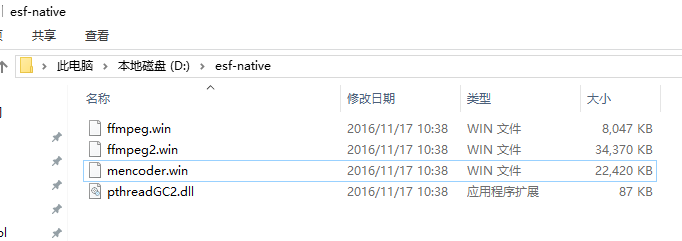
esf-fs-resource项目依赖ffmpeg库

参考文件esf-document\libs\ffmpeg-libs

SVN获取该文件：

<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/libs/ffmpeg-libs/win32>

Win32、Win64的配置，ffmpeg放置路径D:/esf-native



Linux 64bit的配置，ffmpeg放置路径/usr/local/esf-native

SVN获取该文件：

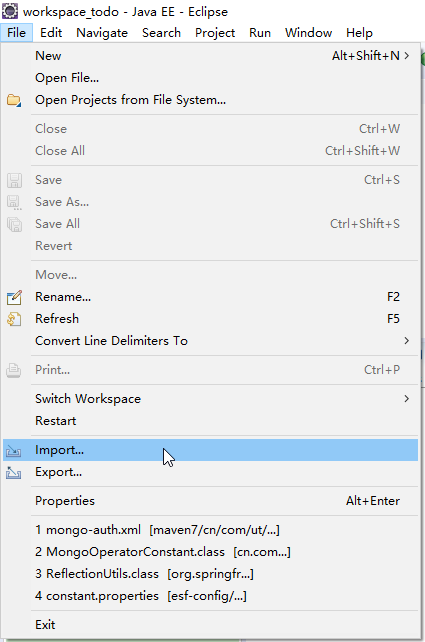
<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_CODE/ESF_CLOUD/esf-document/libs/ffmpeg-libs/Linux64>

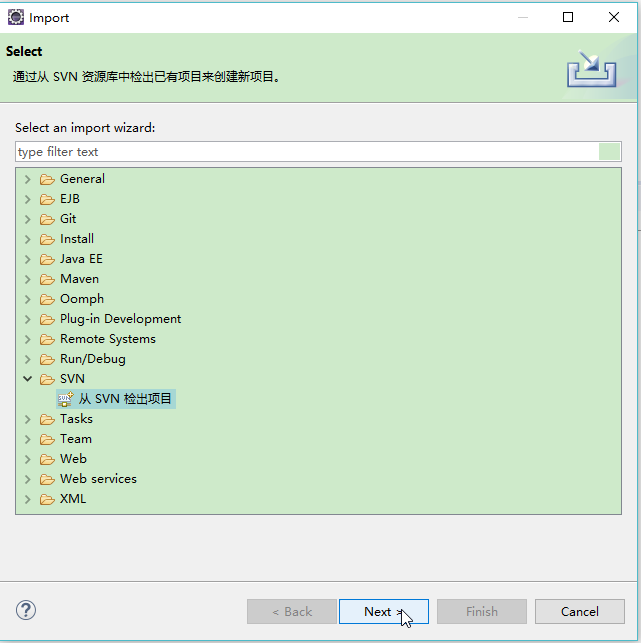


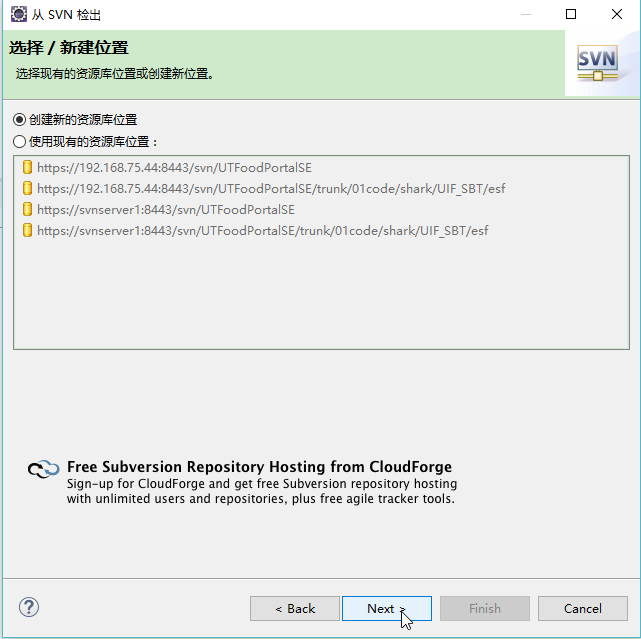
1. 项目构建
   1. eclipse svn checkout（获取源码）

SVN账号密码请咨询管理员

进入【File】【Import】【SVN】

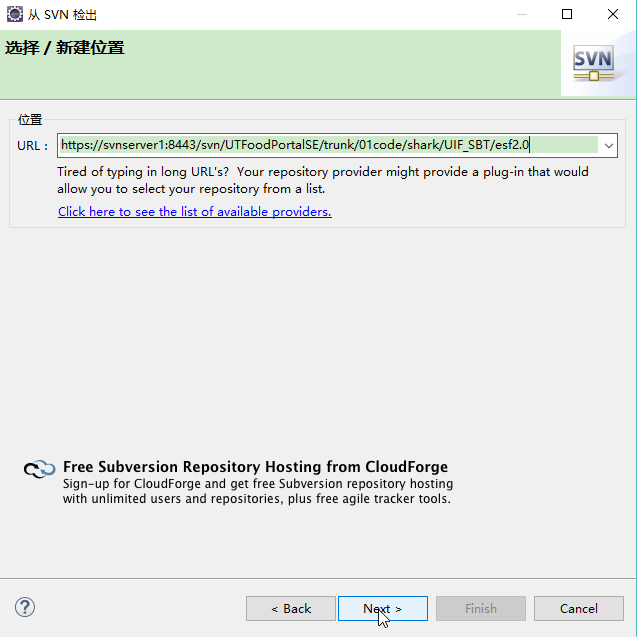


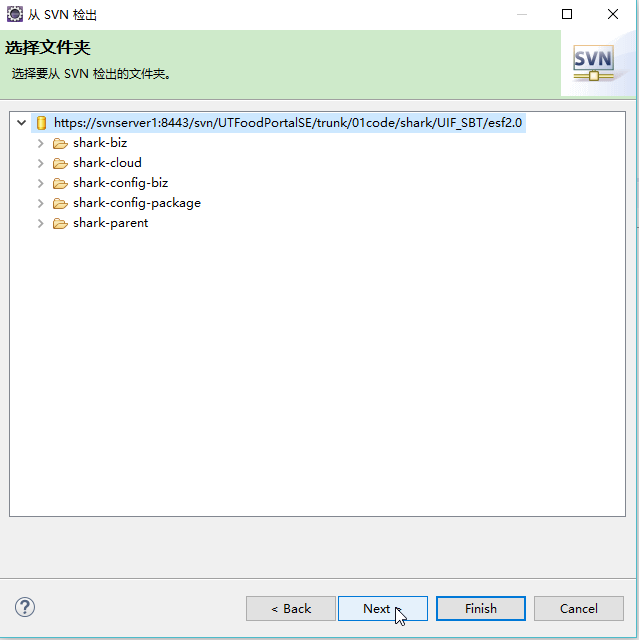


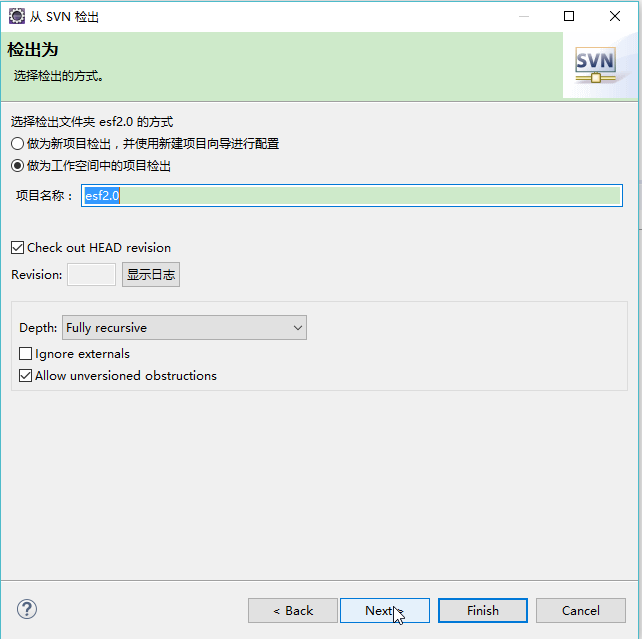


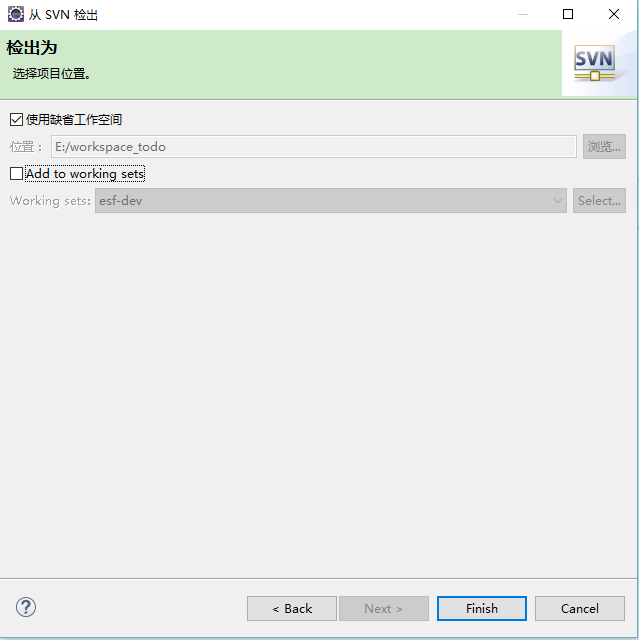
从如下SVN位置检出项目：

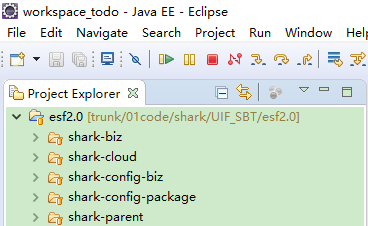
<https://svnserver1:8443/svn/UTFoodPortalSE/trunk/01code/shark/UIF_SBT/esf2.0>





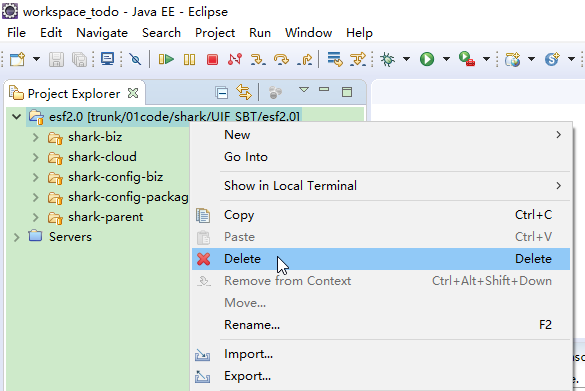


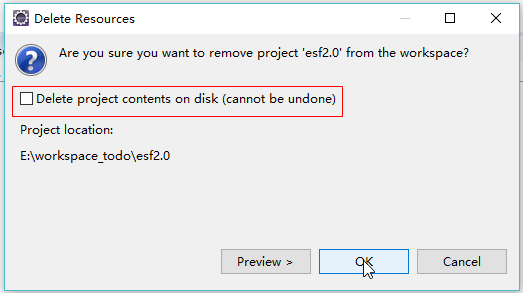




* 1. import maven project（导入maven工程）

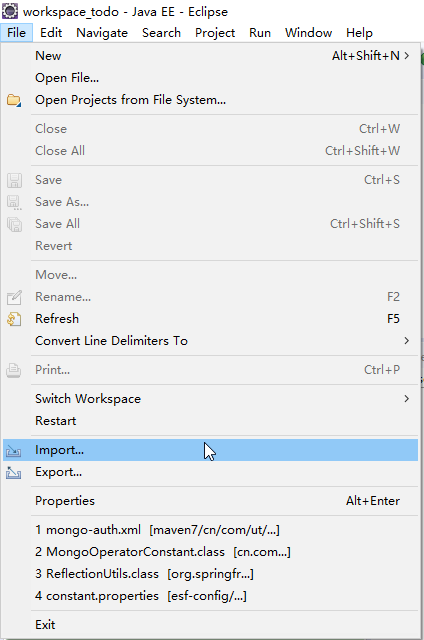
从当前Eclipse工作空间，删除上一步刚检出的工程目录esf2.0，以及子目录，但不要删除物理文件，下一步将以Maven方式重新导入这些工程文件进Eclipse工作空间

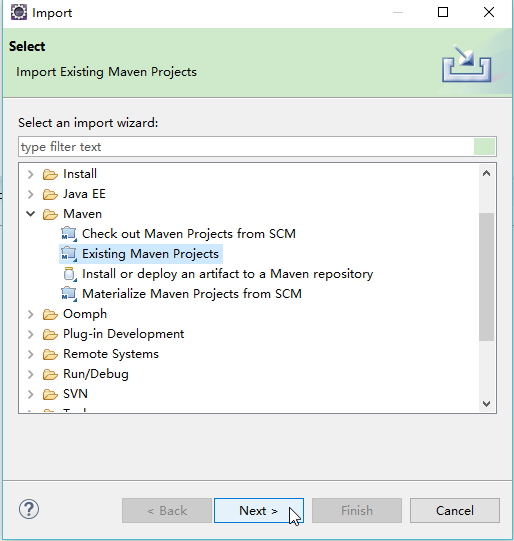


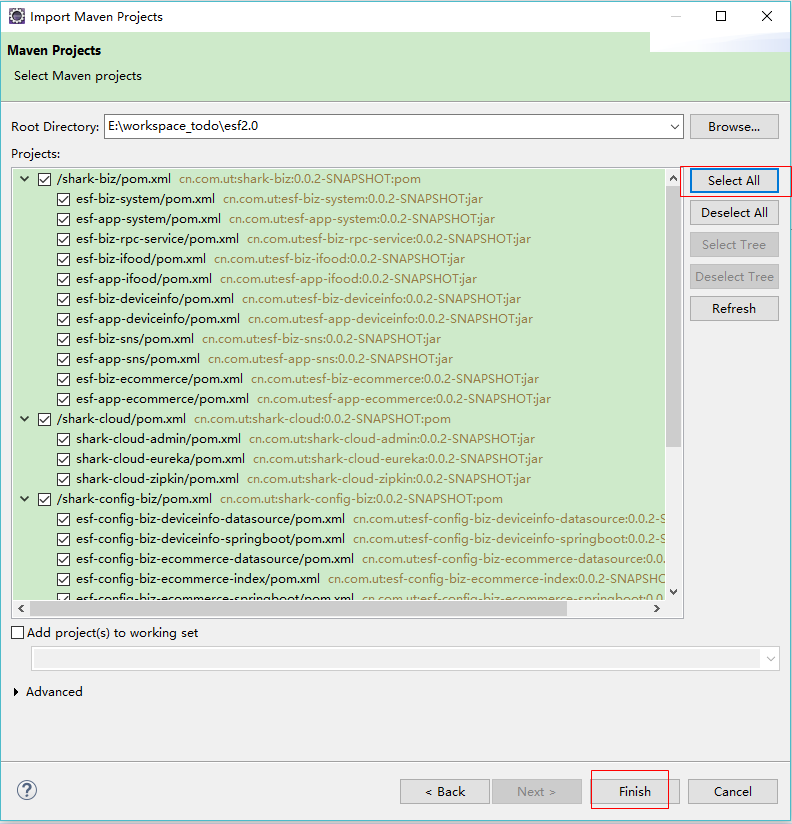


重新以Maven方式，导入之前检出的工程文件，等待Maven构建完成，这可能需要花费一点时间

进入【File】【Import】【Existing Maven Projects】【Next】

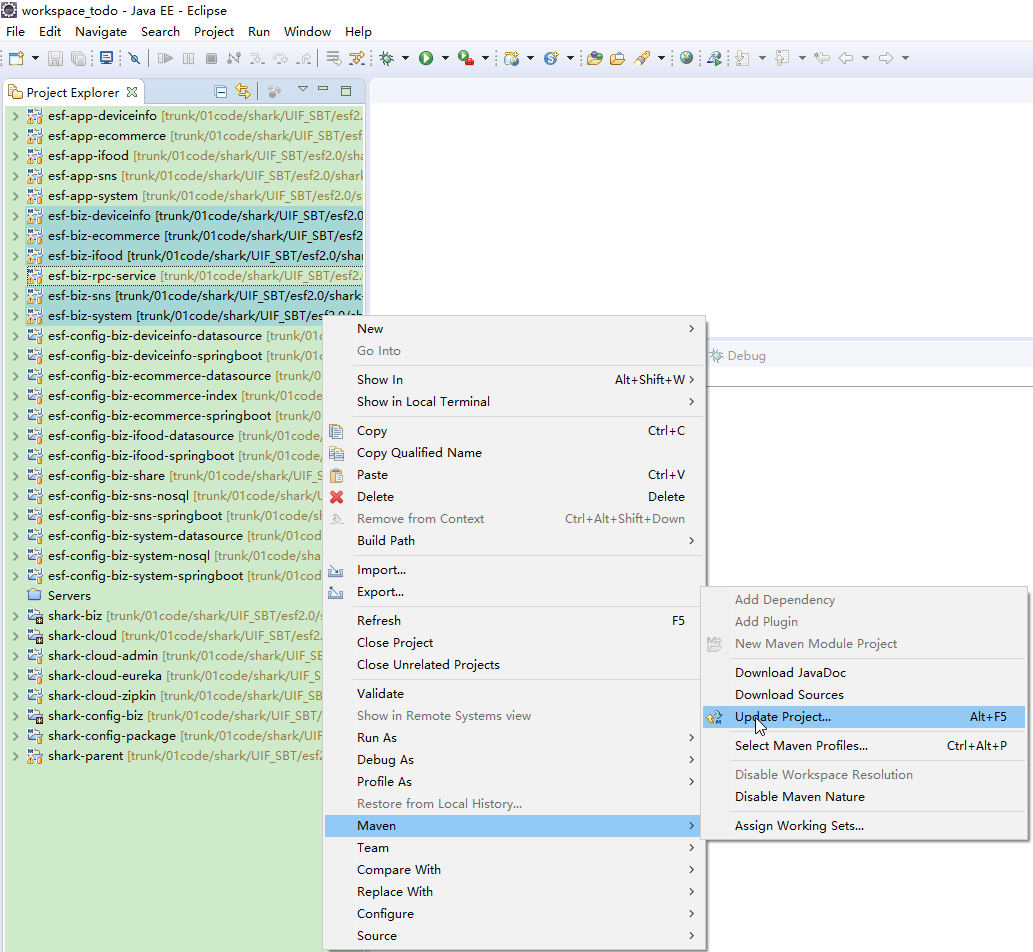


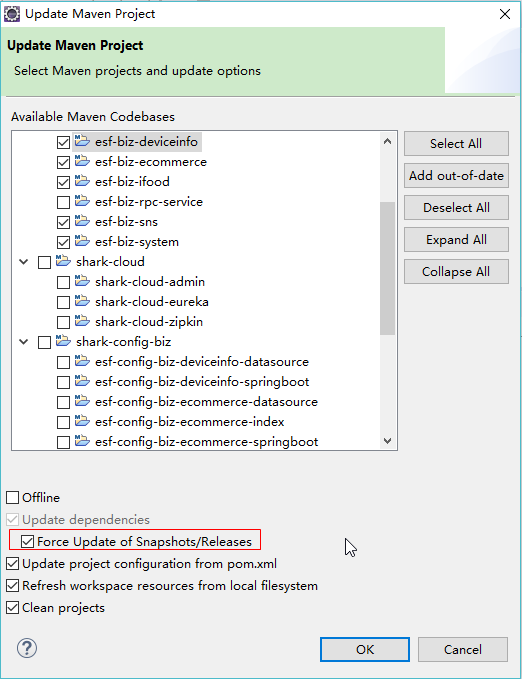




导入的Maven项目如果出现编译错误，请尝试如下步骤解决

选择编译出错项目、鼠标右键、【Maven】【Update Project】





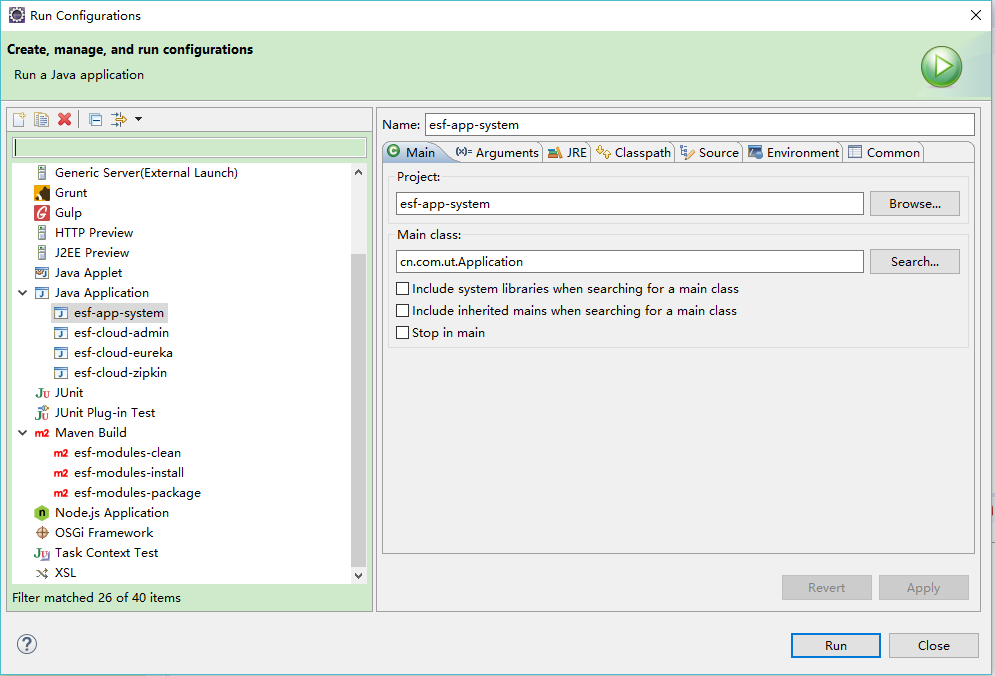
如果编译仍然不通过，一般是Maven远程仓库地址错误引起下载jar包失败，请检查Maven配置文件

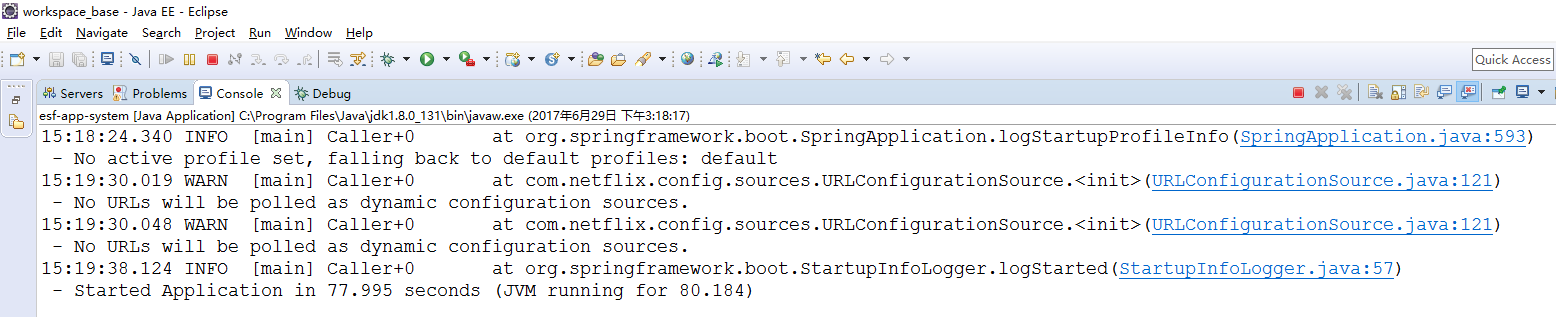
* 1. 项目部署
     1. Eclipse开发调试

在Eclipse进行开发时，一般以Debug模式或Run模式运行Spring Boot应用的main方法即可。

以esf-app-system项目为例

选择该项目、鼠标右键、【Run Configurations】【Run】

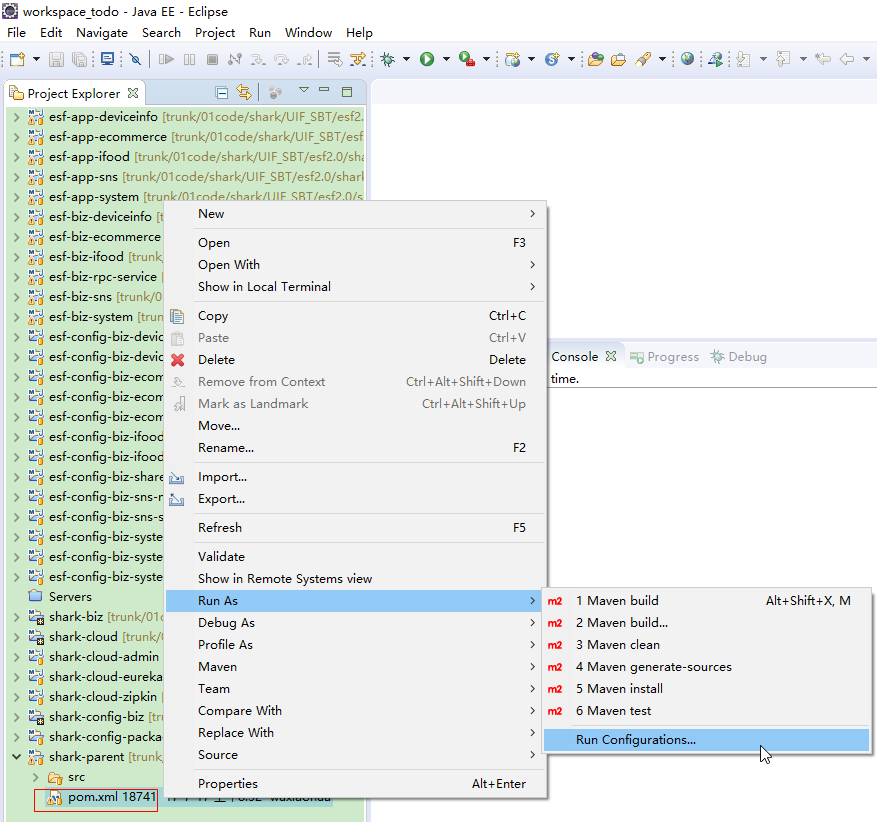


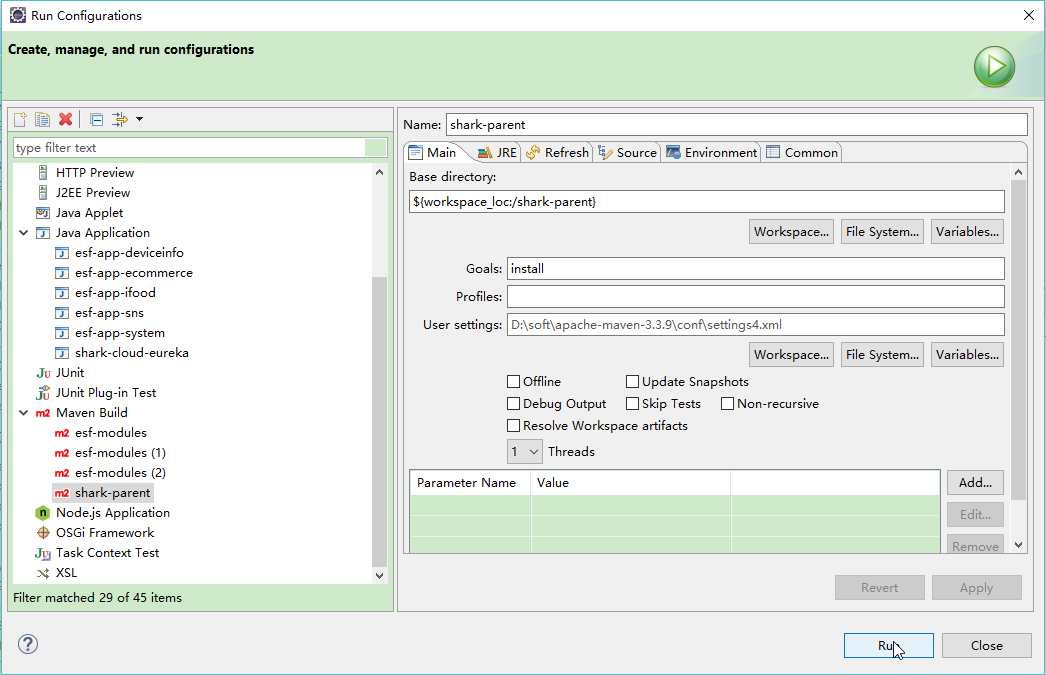


* + 1. Eclipse Maven打包运行

由于应用启动依赖的项目比较多，一般使用聚合项目将依赖项目一并安装至本地仓库。

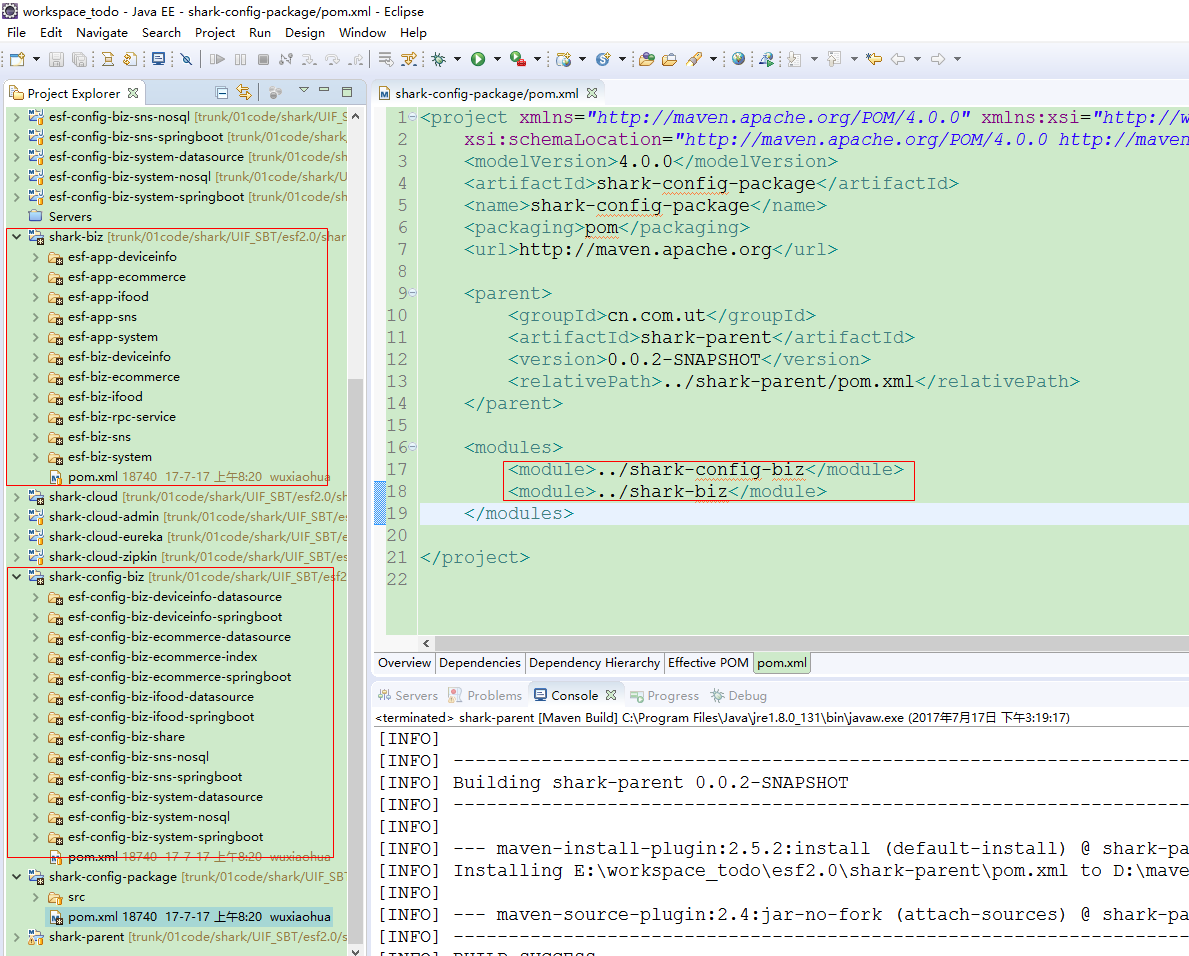
第1步，父项目shark-parent安装至本地仓库，shark-parent配置各子项目的版本，以及maven插件的配置等



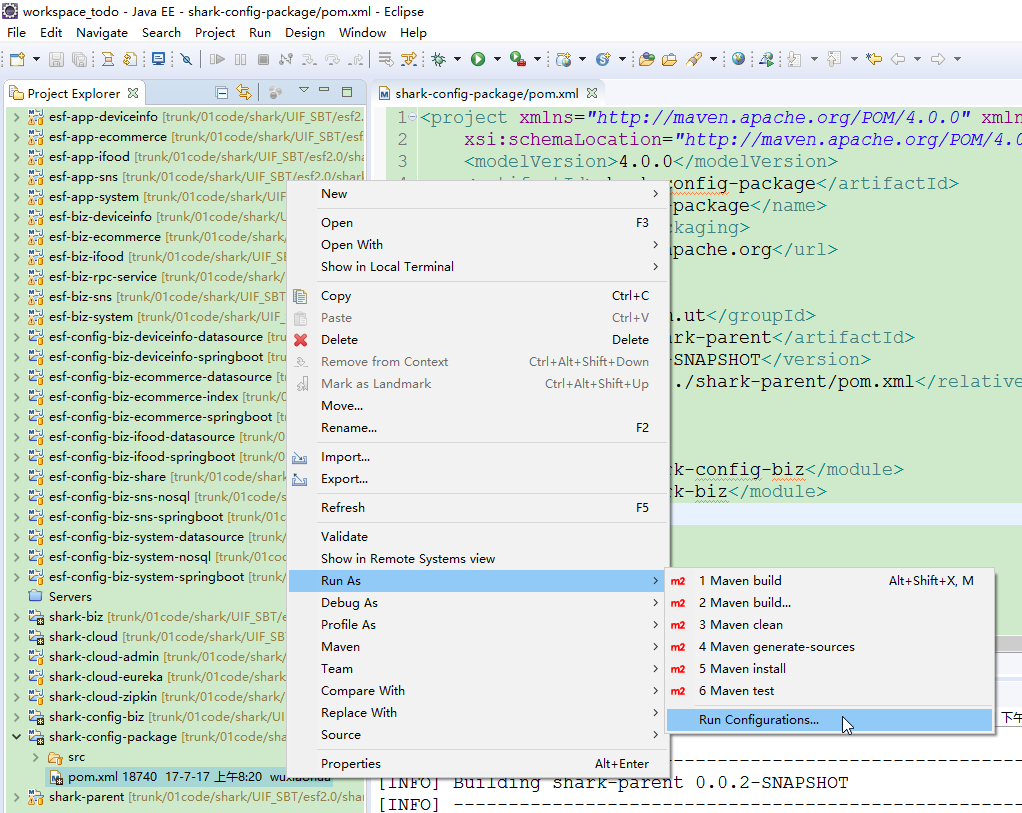


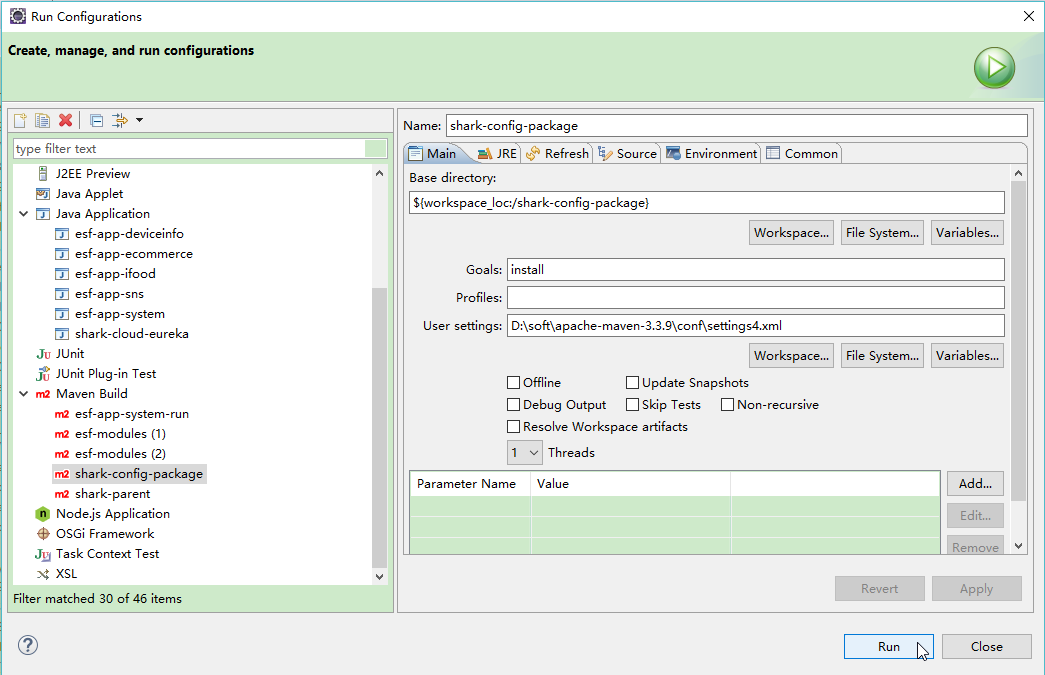
第2步，聚合项目shark-config-package安装全部依赖项目至本地仓库，shark-config-package的作用就是管理应用的全部依赖项目，使用Maven聚合关系。

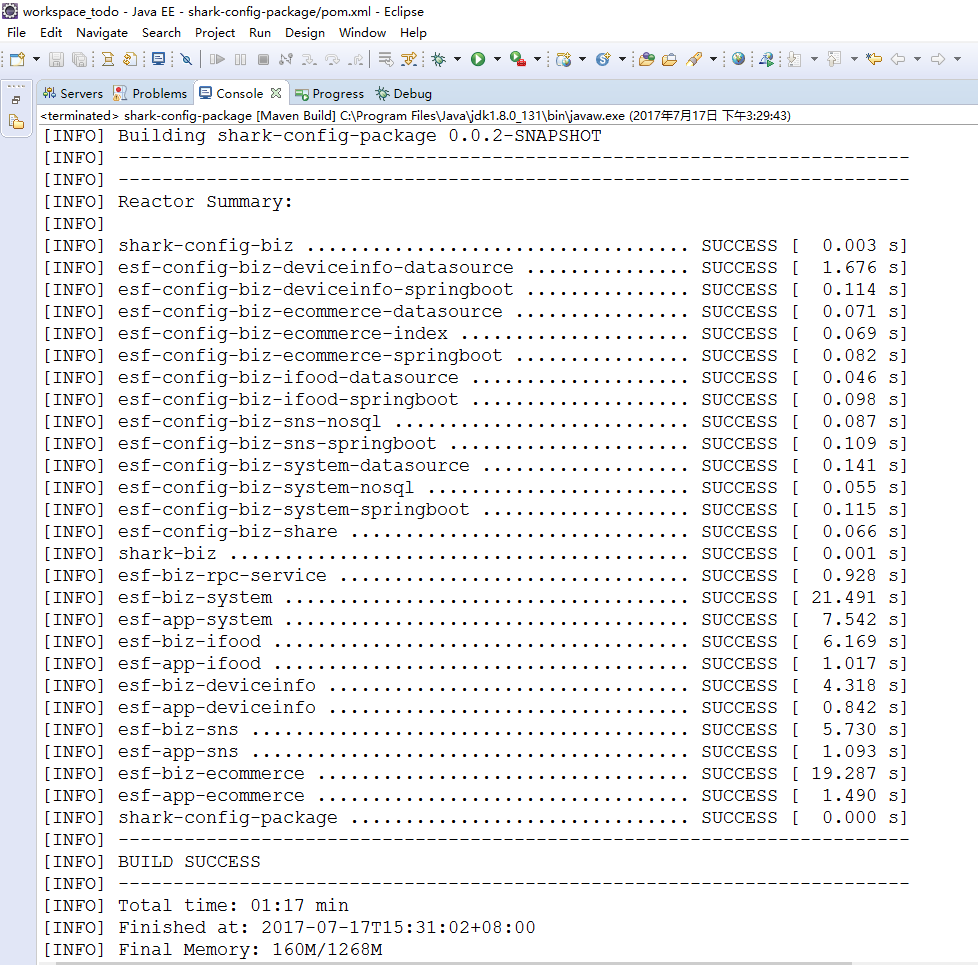
例如，要对应用esf-biz-\*和esf-app-\*进行打包，使用shark-config-package聚合esf-biz-\*和esf-app-\*依赖的全部项目



选中shark-config-package、右键、【Run As】【Run Configurations】

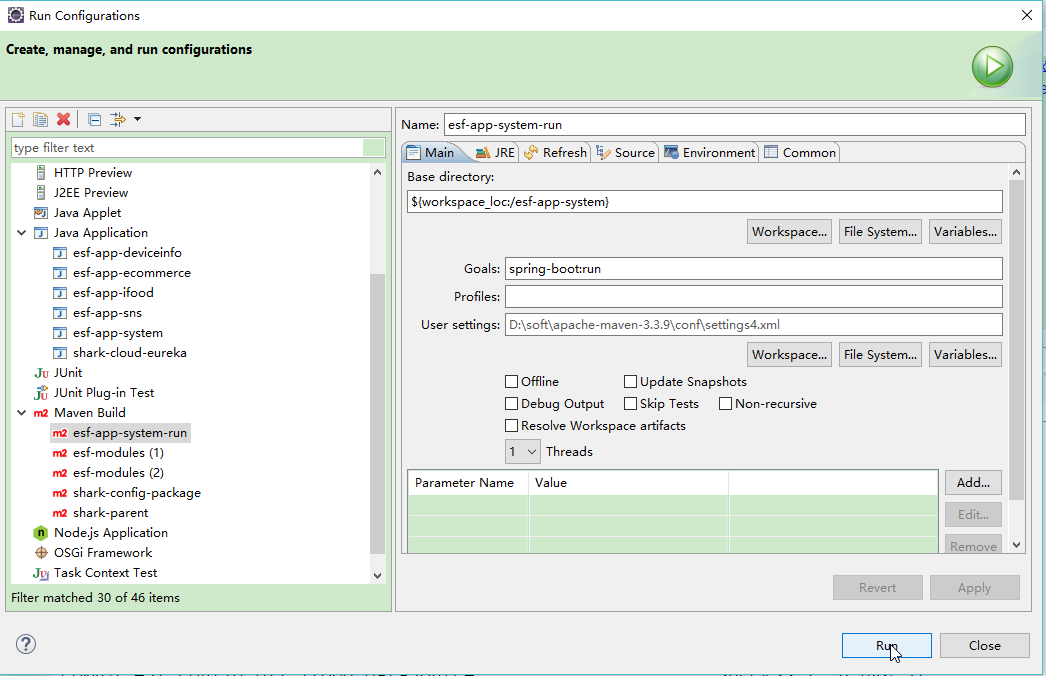


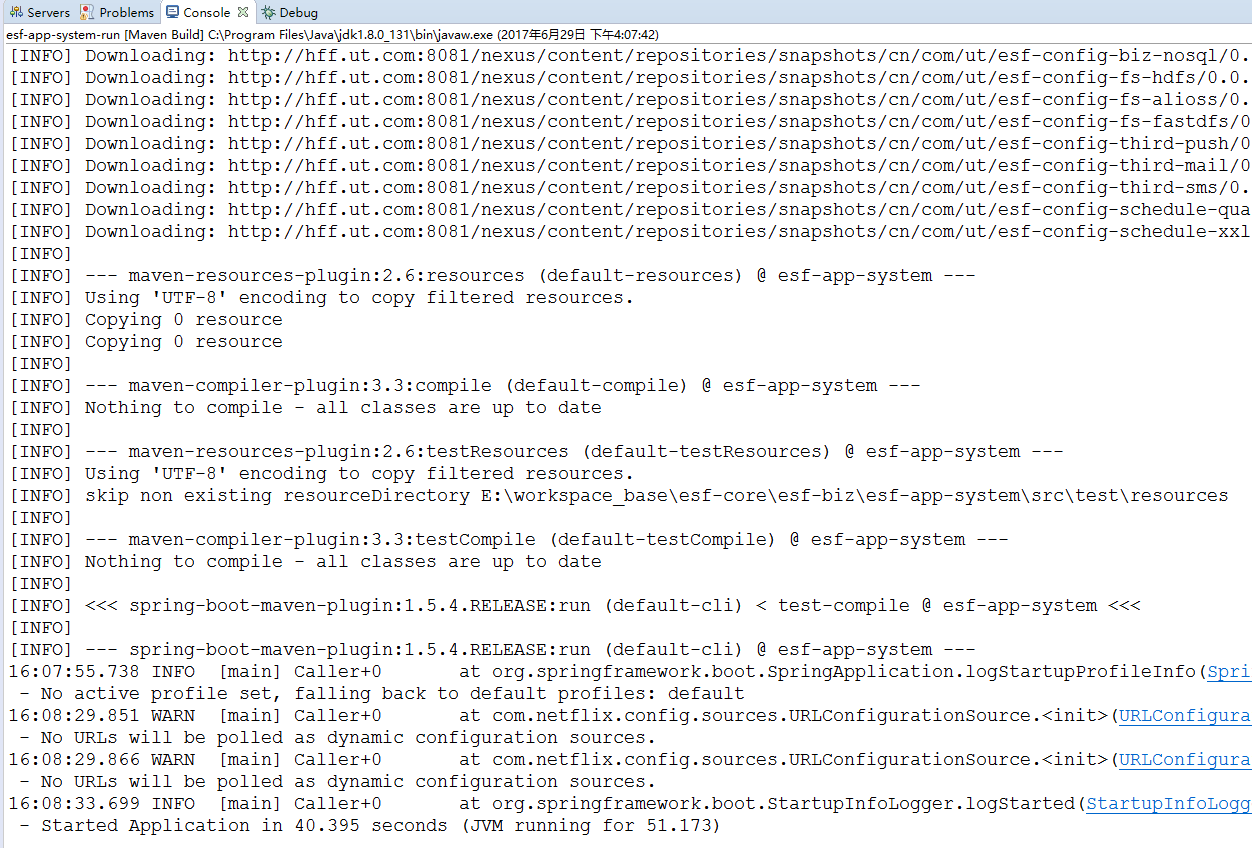




Maven插件运行应用esf-app-system

选中esf-app-system、右键、【Run As】【Run Configurations】

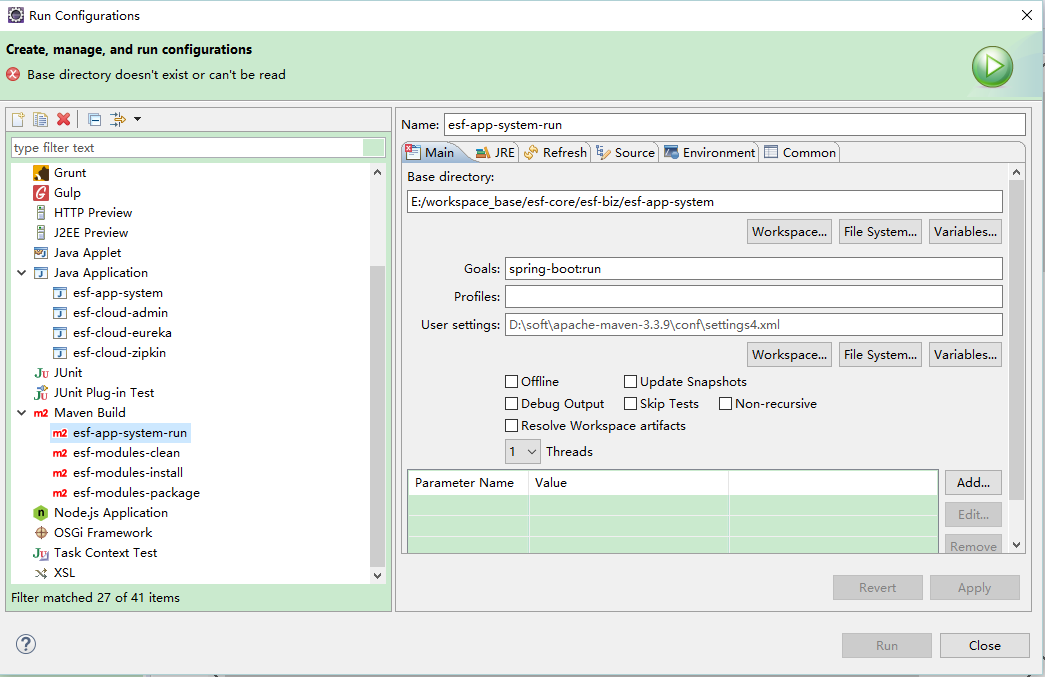


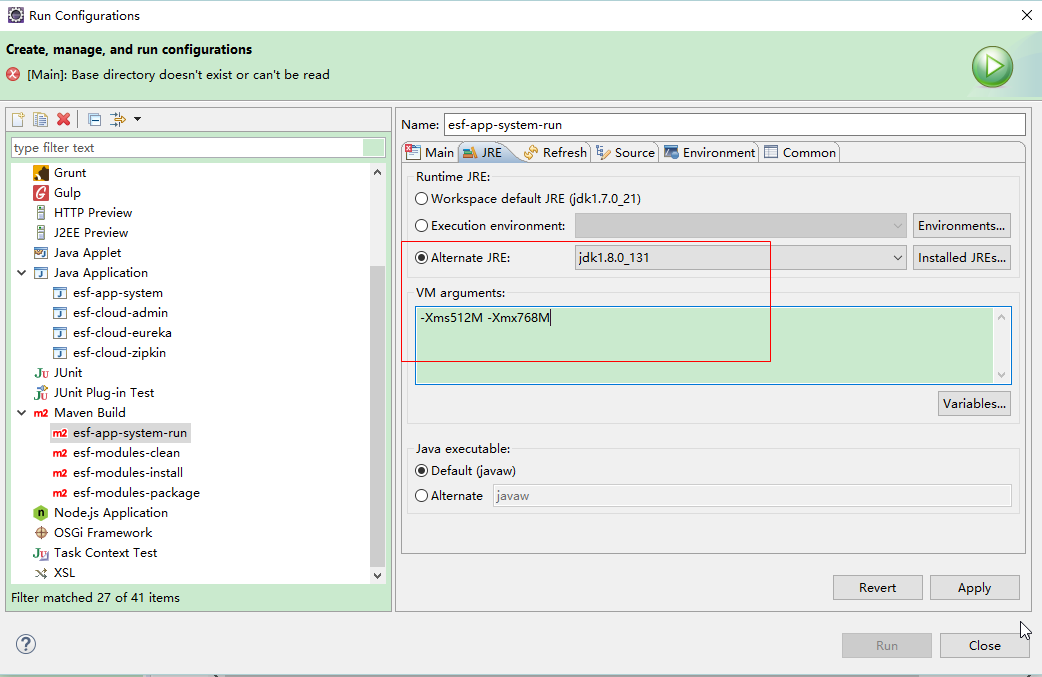


* + 1. 可能遇到的问题

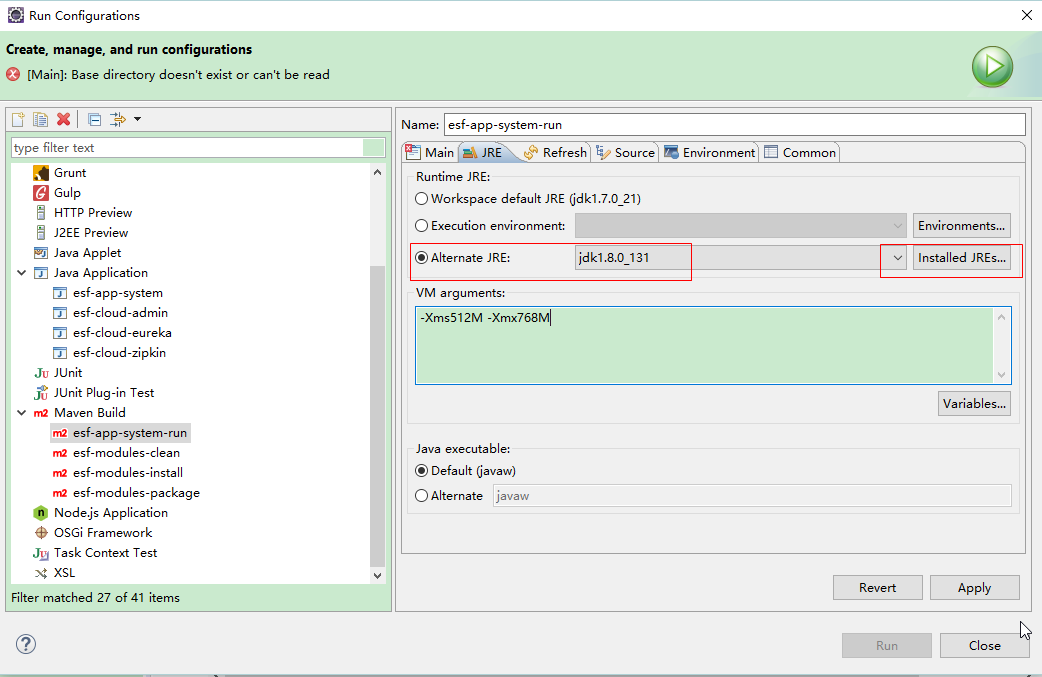
在使用Maven打包、部署可能出现【OutOfMemoryError】、【Java Heap Error】类似情况的错误。请尝试下面几点解决办法。

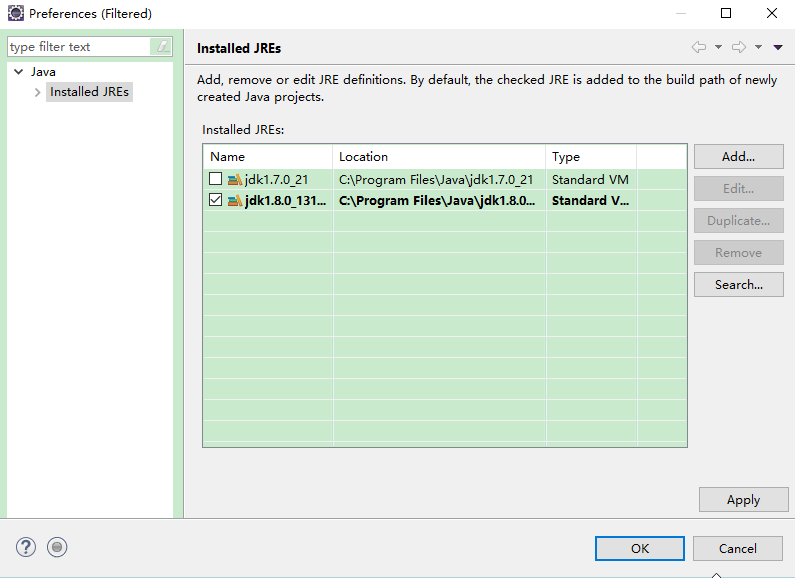
第1点，调整虚拟机vm内存分配参数，例如

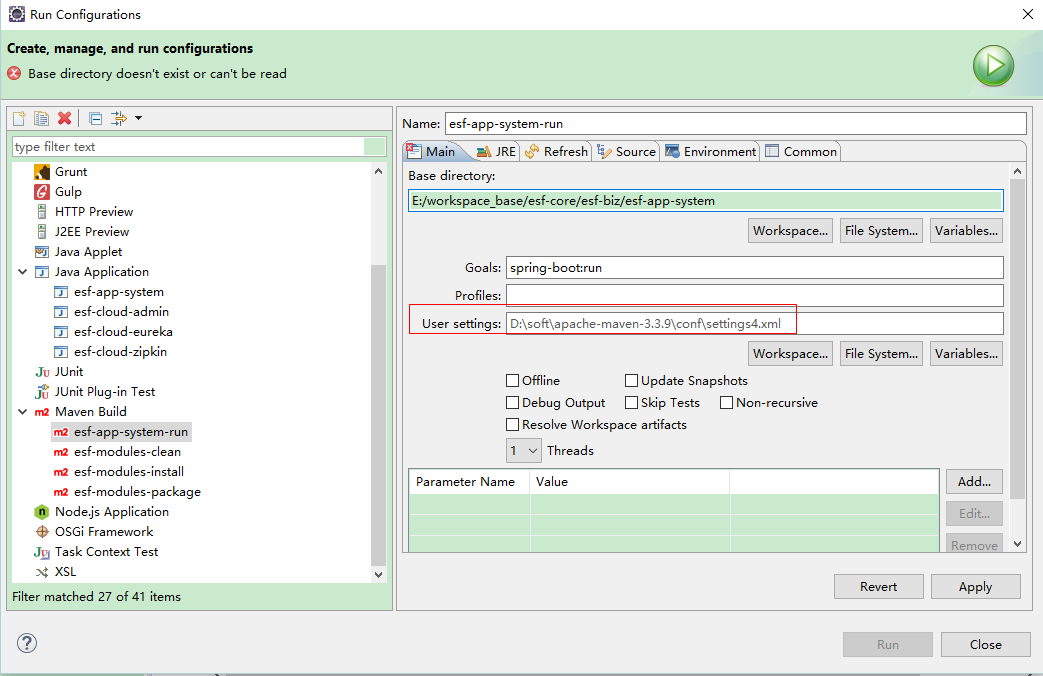


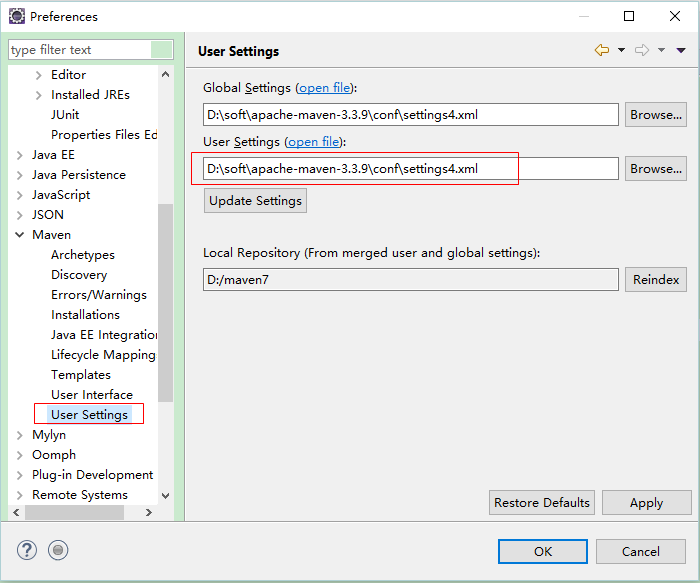


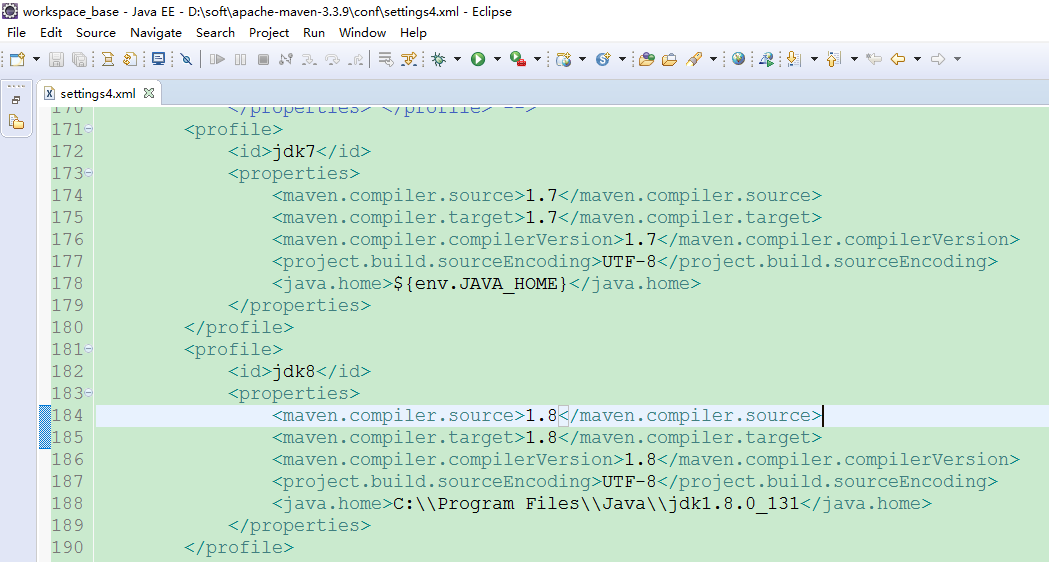
第2点，确保【run Configurations】配置里【JRE】版本和【Main】【User Settings】maven配置文件中指定的jre版本保存一致，例如

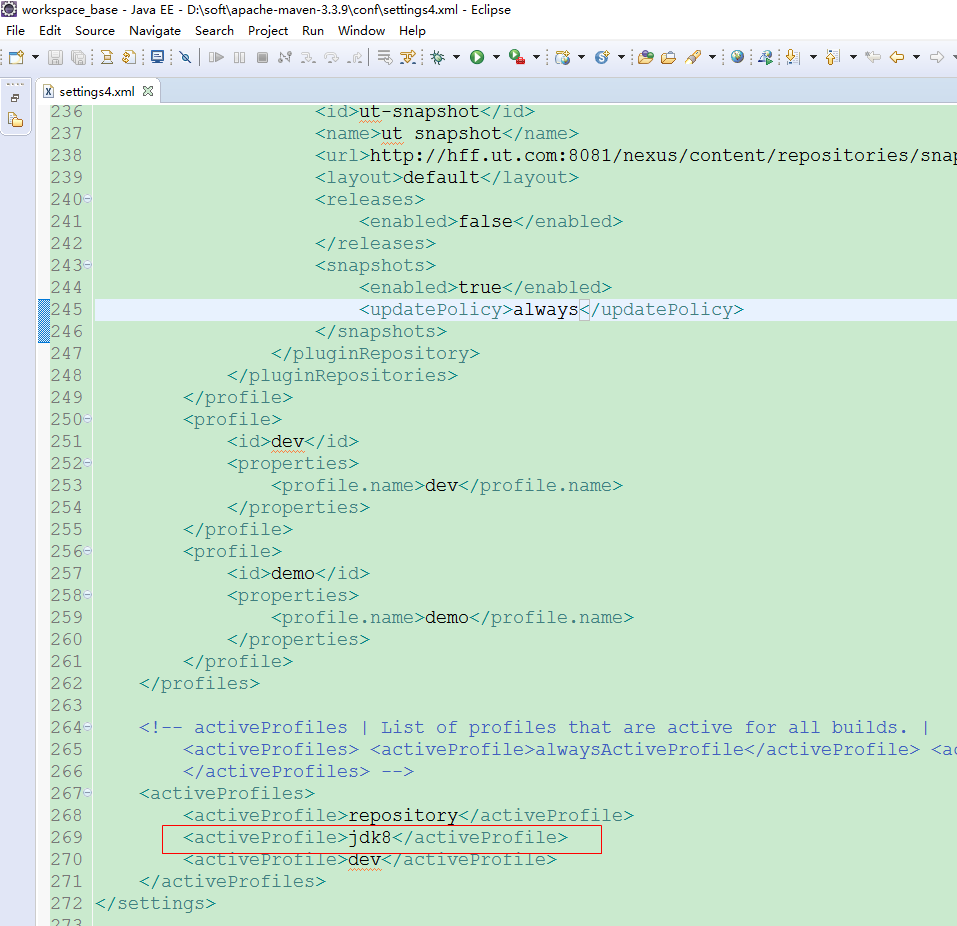








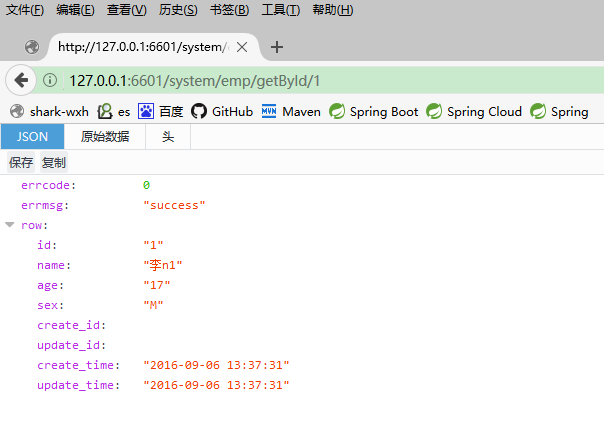




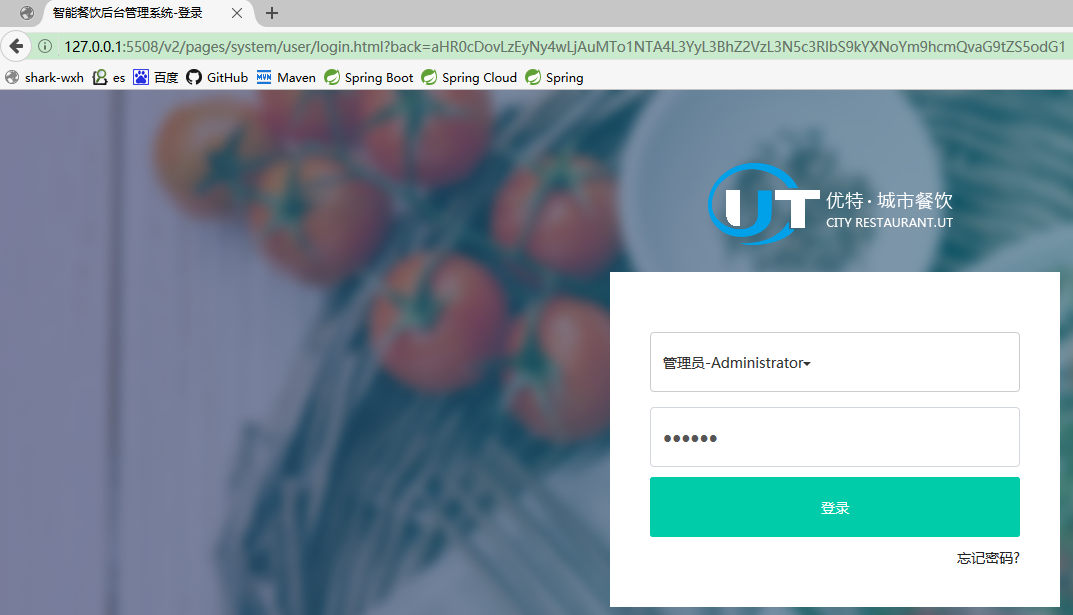
* + 1. 运行是否成功验证

无nginx反向代理，直接访问restful接口

访问<http://127.0.0.1:6601/system/emp/getById/1>



有Nginx反向代理，访问前端页面<http://127.0.0.1:5508>





Nginx的配置这里不作详细介绍，请参考配置文件

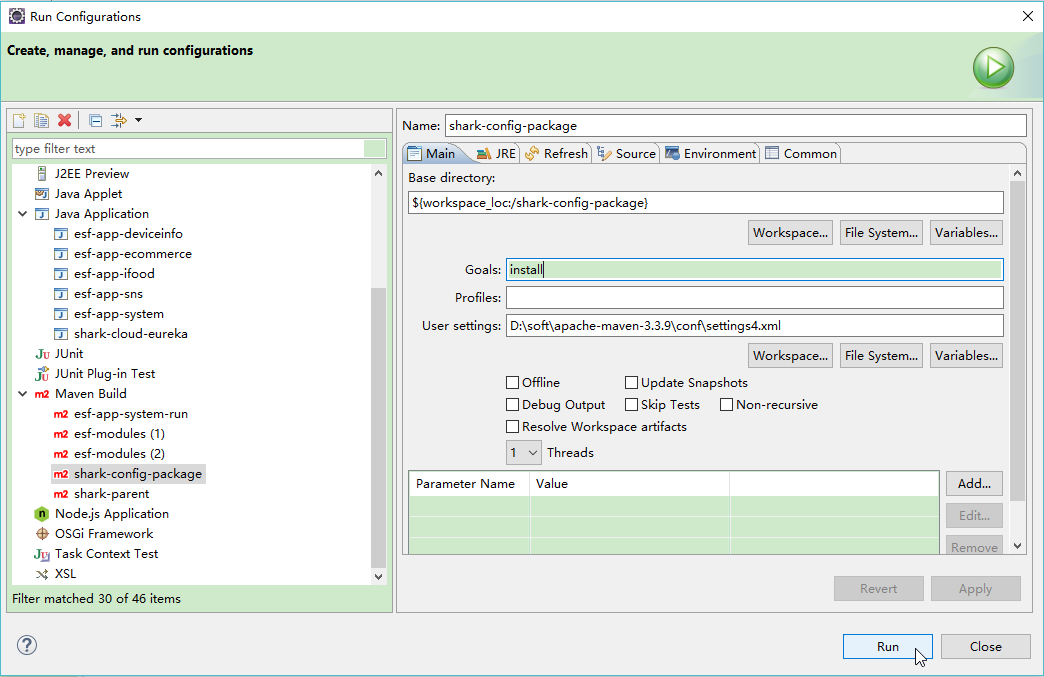
esf-document\settings\nginx-conf\esf-core.conf

esf-document\settings\nginx-conf\ nginx.conf

* + 1. Linux环境部署运行

以esf-app-system应用发布至Ubuntu为例

第1步，打包，选中shark-config-package、右键、【Run As】【Run Configurations】



第2步，从esf-app-system工程目录esf-biz\esf-app-system\target，拷贝应用jar包esf-app-system-0.0.2-SNAPSHOT.jar至Ubuntu服务机器的某个目录下，如/opt/tomcat/wtpwebapps/

第3步，在服务器上使用shell脚本启动停止jar，脚本参考文件为

esf-document\settings\springboot-run\esf-sbt

根据实际情况，可对该脚本内容进行调整

一般放置在服务器/etc/init.d/esf-sbt目录，需要授予执行权限sudo chmod a+x esf-sbt

运行jar：service esf-sbt start

停止运行jar：service esf-sbt stop

1. 项目目录说明

|  |  |  |
| --- | --- | --- |
| 组件包 | 组件 | 描述 |
| shark-parent | shark-parent | Maven父工程 |
| shark-biz | shark-biz | 业务目录，Maven聚合工程 |
| esf-biz-\* | 业务Controller、Service、DAO |
| esf-app-\* | Spring boot应用，Web应用 |
| esf-biz-rpc-service | 业务远程接口 |

1. 项目配置说明

|  |  |  |
| --- | --- | --- |
| 组件包 | 组件 | 描述 |
|  |  |  |
| shark-config-biz | esf-config-biz | 业务配置目录，Maven聚合工程 |
| esf-config-biz-\*-datasource | 数据库实体关系配置 |
| esf-config-biz-\*-index | ES索引配置 |
| esf-config-biz-\*-nosql | mongo实体关系配置 |
| esf-config-biz-share | 公共配置 |
| esf-config-biz-\*-springboot | spring boot配置 |
| shark-config-package | shark-config-package | 一键打包部署项目，Maven聚合工程 |