

若依系统开发环境搭建手册

1. 准备工作

1.1 前置环境准备——安装 Maven

若依基于 Maven 管理项目的构建，需要先安装好相应的版本。

1.2 开发工具

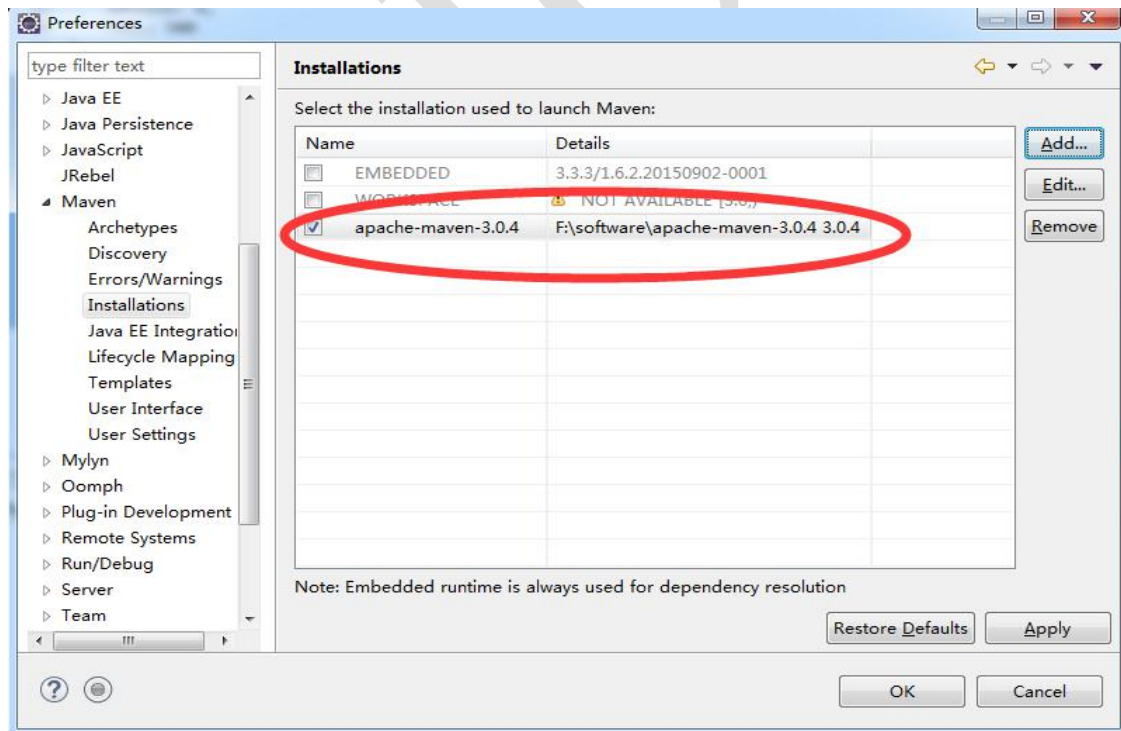
若依系统采用 Eclipse 作为开发工具。但不局限于 Eclipse。此处仅介绍在 Eclipse 搭建开发环境所需的操作。

2. 开发环境搭建

2.1 开发工具的配置

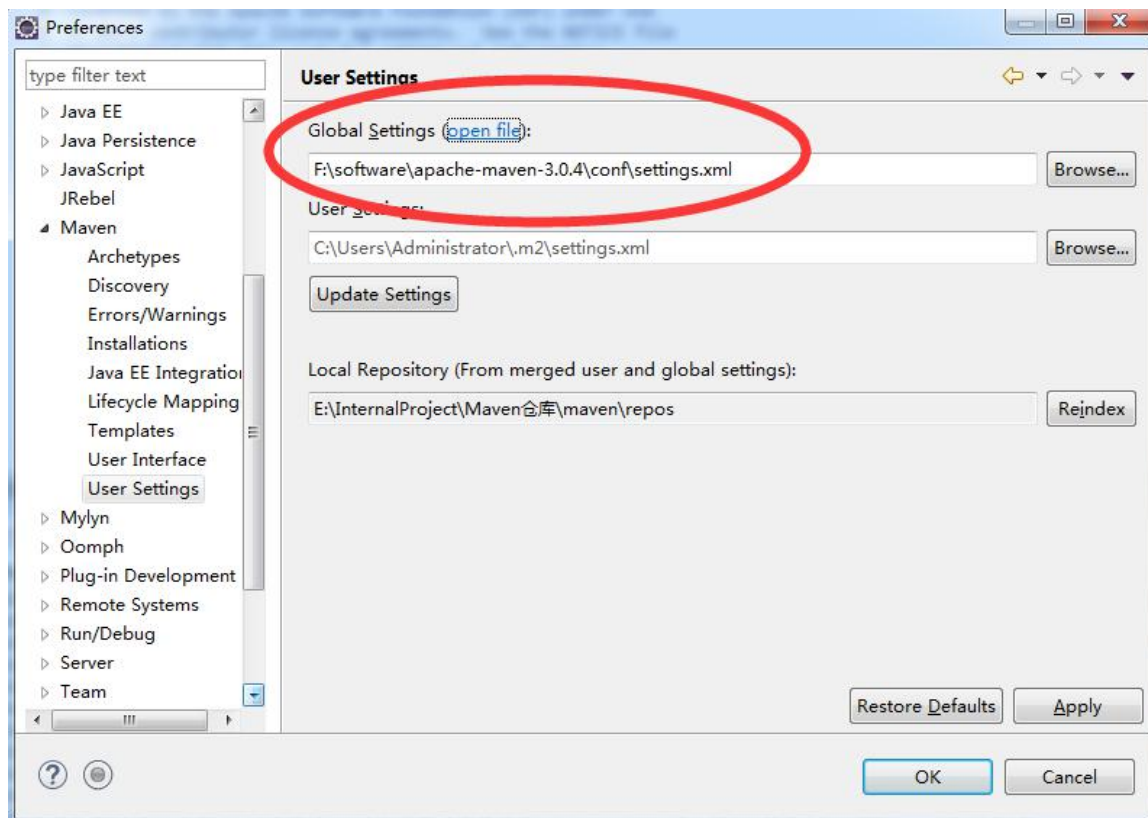
2.1.1 配置 Maven

进入 Window->Preferences->Maven->Installations 页面，设置已经安装好的 Maven



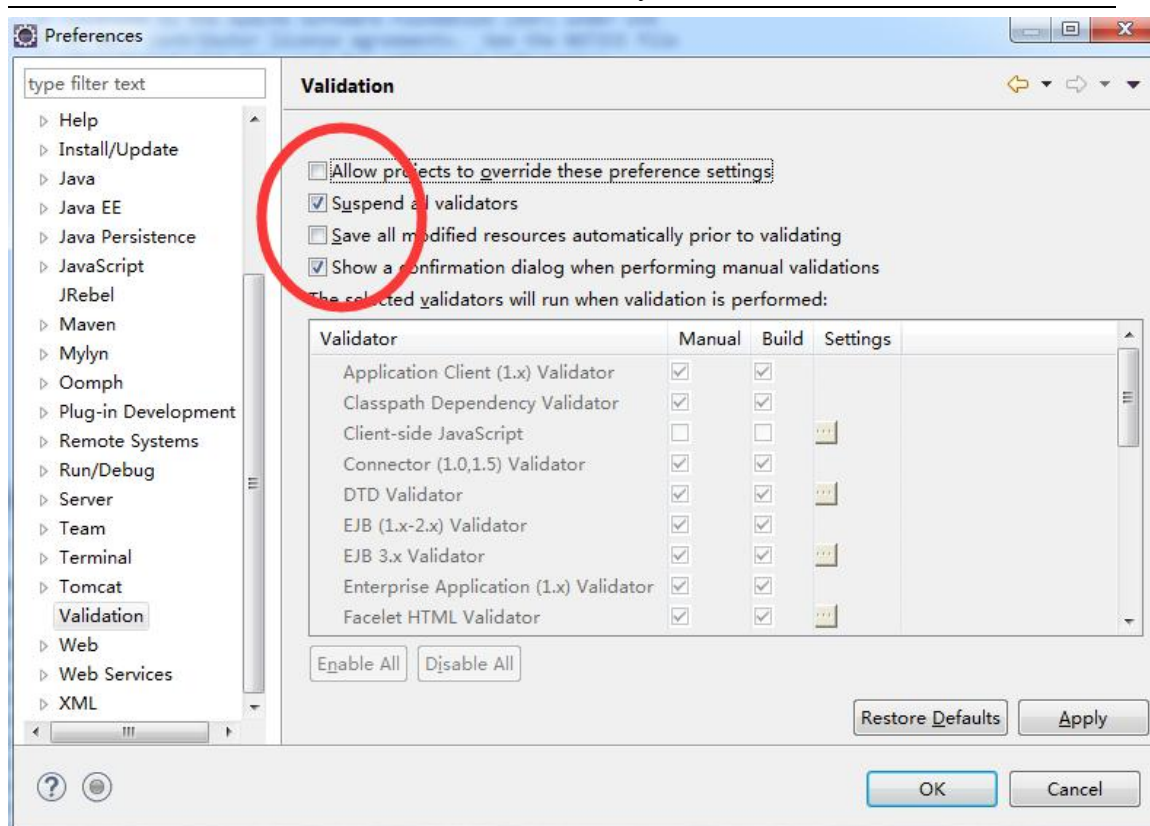
2.1.2 配置 Maven 仓库路径

进入 Window->Preferences->Maven->User Settings 页面，配置仓库路径



2.1.4 关闭校验

进入 Window->Preferences->Validation 页面，勾选 “Suspend all validators”，关闭校验

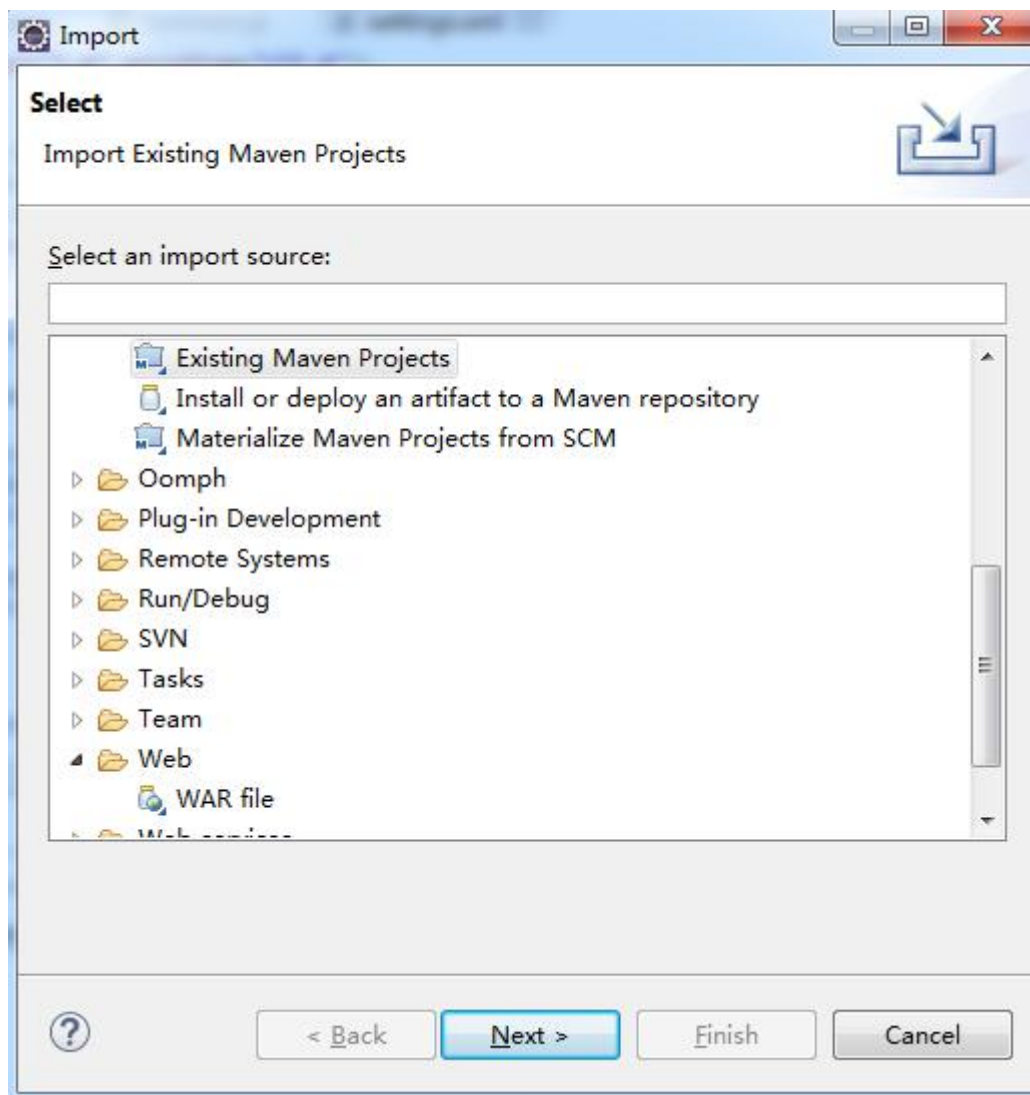


2.2 导入工程

通过 Eclipse 导入工程，步骤如下：

(1)点击左侧项目区域—>Import...

RuoYi Use Only▲



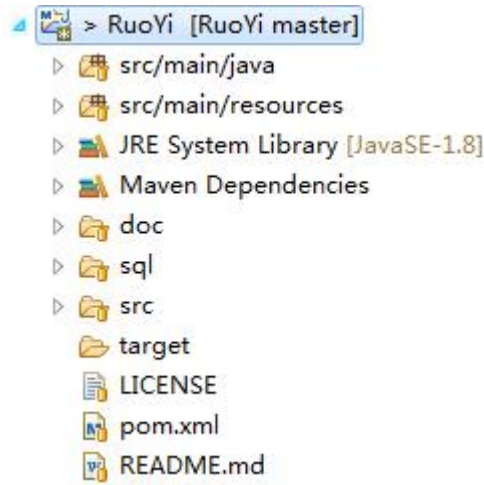
(2)选择 RuoYi



(3)点击 Finish

RuoYi Use Only▲

(4)RuoYi 的代码就被导出到 Eclipse 中了，此时可以在工程视图中看到。



3. 运行若依系统

3.1 必要的配置

3.1.1 修改数据库连接

编辑 src/main/ resources 目录下的 application-druid.yml 文件，修改数据库地址账号信息。
执行 sql/ ry_20180423.sql, quartz.sql 两个文件 日期随版本变化

3.1.2 开发环境配置

编辑 src/main/ resources 目录下的 application.yml 文件，
默认端口为 80

```
7 #开发环境配置
8 server:
9   #端口
10  port: 80
11  tomcat:
12    uri-encoding: utf-8
```

3.1.3 代码生成配置

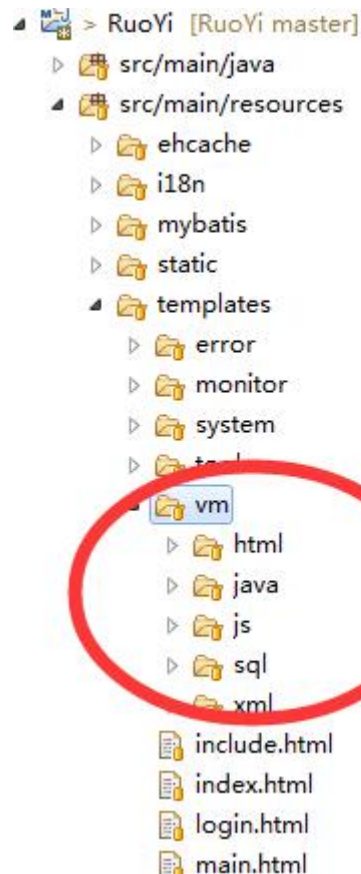
编辑 src/main/ resources 目录下的 application.yml 文件，

RuoYi Use Only ▲

```
'4 # 代码生成
'5 gen:
'6   # 作者
'7   author: ruoyi
'8   # 默认生成包路径 module 需改成自己的模块名称 如 system monitor tool
'9   packageName: com.ruoyi.project.module
'10  # 自动去除表前缀，默认是true
'11  autoRemovePre: true
'12  # 表前缀 (类名不会包含表前缀)
'13  tablePrefix: sys_
```

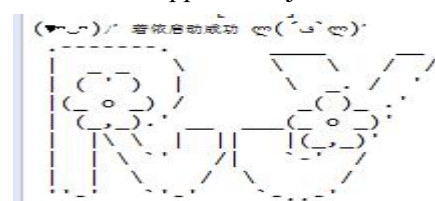
默认为 module，根据实际情况修改即可。生成的表要有注释

注：如对模板有特殊需求，可自行修改。编辑 src/main/resources/templates/vm 目录下



3.2 启动及验证

启动 RuoYiApplication.java 出现如下图表示启动成功



打开浏览器，输入：http://localhost:80/

<以上所有信息均为若依所有>

All Rights reserved, RuoYi 2018

若能正确展示登录页面，并能成功登录，登录后菜单及页面展示正常，则表明环境搭建成功。
默认密码为 admin/admin123

演示地址：<http://www.ruoyi.club>

4. 部署若依系统

4.1 War 部署方式

4.1.1 修改 pom.xml 文件。将 jar 修改为 war

```
→<groupId>com.ruoyi</groupId><br>→<artifactId>RuoYi</artifactId><br>→<version>1.1.1</version><br>→<packaging>war</packaging><br>
```

4.1.2 在 spring-boot-starter 依赖中移除 tomcat 模块

```
><dependencies><br>→<br>→<!-- SpringBoot 核心包 --><br>→<dependency><br>→→<groupId>org.springframework.boot</groupId><br>→→<artifactId>spring-boot-starter</artifactId><br>→→<exclusions><br>→→→<exclusion><br>→→→<artifactId>spring-boot-starter-tomcat</artifactId><br>→→→<groupId>org.springframework.boot</groupId><br>→→→</exclusion><br>→→</exclusions><br>→</dependency><br>
```

```
<exclusions><br>  <exclusion><br>    <artifactId>spring-boot-starter-tomcat</artifactId><br>    <groupId>org.springframework.boot</groupId><br>  </exclusion><br></exclusions>
```

4.1.3 部署到 tomcat 的 webapps 目录下面

<以上所有信息均为若依所有>
All Rights reserved, RuoYi 2018

默认为 RuoYi.war

名称	修改日期	类型
docs	2018/4/28 19:44	文件夹
examples	2018/4/28 19:44	文件夹
host-manager	2018/4/28 19:44	文件夹
manager	2018/4/28 19:44	文件夹
ROOT	2018/4/28 19:44	文件夹
RuoYi	2018/4/28 21:36	文件夹
RuoYi.war	2018/4/28 21:35	WAR 文件

4.1.4 启动及验证

运行 startup.bat 出现如下图即部署成功

```
localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment d
ager] has finished in [36] ms
localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying we

localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment c
has finished in [34] ms
main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler [http-nio-8080"]
main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
main] org.apache.catalina.startup.Catalina.start Server startup in 18721 ms
[RuoyiScheduler]] INFO o.q.c.QuartzScheduler - [start 547] - Scheduler RuoyiScheduler_$L
```

4.2 Jar 方式部署

执行命令：java -jar RuoYi.jar

脚本执行：ry.sh start