XXL-JOB 1.3 用户手册

[1. 简介 2](#_Toc27516)

[1.1 背景 2](#_Toc14618)

[1.1 发展 2](#_Toc25846)

[1.2 概述 2](#_Toc32235)

[1.2 特点 2](#_Toc3419)

[1.3 下载 3](#_Toc80)

[1.4 环境 3](#_Toc29457)

[2. 快速入门 4](#_Toc27036)

[2.1 初始化“调度数据库” 4](#_Toc17537)

[2.2 编译源码 5](#_Toc14192)

[2.3 部署“调度中心” 5](#_Toc22075)

[2.4 部署“执行器” 6](#_Toc19397)

[2.5 开发第一个任务“Hello World” 6](#_Toc2898)

[3. 系统详解 6](#_Toc27229)

[3.1 源码目录介绍 6](#_Toc14740)

[3.1 “调度数据库”详解 6](#_Toc30358)

[3.2 “调度中心”详解 6](#_Toc12426)

[3.3 “执行器”详解 6](#_Toc7410)

[4. 任务详解 6](#_Toc351)

[4.1 BEAN模式任务 6](#_Toc11736)

[4.2 GLUE任务任务 6](#_Toc22307)

[5. 任务管理 6](#_Toc23377)

[5.1 编辑任务信息 6](#_Toc27141)

[5.2 编辑GLUE代码 6](#_Toc23003)

[5.3 恢复/暂停 6](#_Toc11640)

[5.4 手动触发一次调度 7](#_Toc9336)

[5.5 查看日志 7](#_Toc6237)

[5.6 删除任务 7](#_Toc26287)

[6. 设计原理 7](#_Toc8106)

[6.1 总体架构 7](#_Toc25366)

[6.2 调度原理 7](#_Toc32554)

[6.3 通讯原理 7](#_Toc9110)

[6.4 执行器原理 7](#_Toc2371)

[7. 其他 7](#_Toc26780)

[7.1 接入登记 7](#_Toc15846)

[7.2 报告问题 7](#_Toc14010)

# 1. 简介

## 1.1 背景

作业调度系统我使用过相当一部分，比如Java Timer、linux corntab和quartz，但是都遇到一些共同的问题，开发成本高，维护复杂等等。在我进入大众点评之后，我试用过内部调度系统，功能完善，但是开发和部署流程略复杂。

我钟爱KISS原则，因此我萌发了打造XXL-JOB的想法，期望是能够创造一种全新的调度体验。

## 1.2 发展

我于2015-11-28在github上创建XXL-JOB项目仓库并提交第一个commit，随之进行系统结构设计，UI选型，交互设计……

于2016-12-05日XXL-JOB终于release了第一个大版本V1.0， 随后我将之发布到OSCHINA，XXL-JOB在OSCHINA上获得了@红薯的推荐，同期分别达到了OSCHINA的“热门动弹”排行第一和git.oschina的月热度排行第一，在此特别感谢红薯，感谢大家的关注和支持。

于2015-12月中旬我将XXL-JOB发表到我司内部知识库，得到内部同事认可。于2016-01-21开始我司展开XXL-JOB的内部接入和定制工作。

我司大众点评已接入XXL-JOB，内部别名《Ferrari》（Ferrari基于XXL-JOB的V1.1版本定制而成，新接入应用推荐升级最新版本V1.3）。自2016-01-21接入至2016-05-20为止，该系统已调度40000余次，表现优异。

至今，XXL-JOB已接入多家公司的线上产品线，场景如电商业务，O2O业务和大数据作业等等，欢迎大家使用，XXL-JOB也将拥抱变化，持续发展。

## 1.3 概述

XXL-JOB是一款简单高效的分布式任务调度平台，支持可视化且实时动态的管理Trigger和Job。

## 1.4 特点

简单：支持通过Web在线开发Job并配置，流程简洁；

高效：任务动态部署和配置，节省打包和部署消耗，提高开发效率；

分布式：得益于Quartz的集群支持，XXL-JOB调度中心天生支持HA，避免单点故障；

实时Trigger：支持可视化的管理Trigger，动态管理Triggrer；

实时Job：支持可视化的管理Job，动态管理Job，V1.3支持在线开发业务代码；

## 1.5 下载

Github地址：<https://github.com/xuxueli/xxl-job>

Git@OSC地址：<http://git.oschina.net/xuxueli0323/xxl-job>

（我将会在两个git仓库同步发布最新代码）



（图1.3：github下载链接位置）

源码下载请前往github自行下载，下载位置见上图1.3；

## 1.6 环境

Maven3

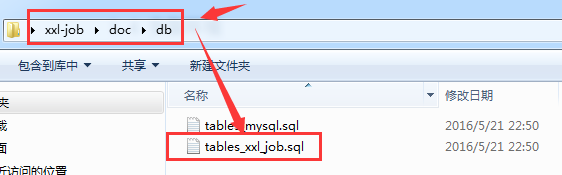
JDK1.7

Tomcat7

Mysql5.5

# 快速入门

## 2.1 初始化“调度数据库”



（图2.1A：数据库建表SQL文件位置截图，）

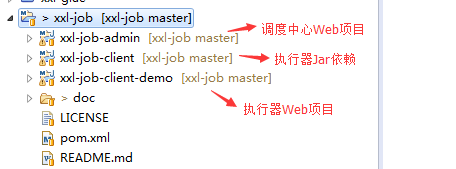
请下载项目源码并解压，然后前往“图2.1A”所示目录，获取SQL脚本并执行，脚本文件位置：**“源码解压根目录”\xxl-job\doc\db\tables\_xxl\_job.sql**



（图2.1B：调度数据库表一览）

正常情况下，应该生成（图2.1B）中所示14张表。

## 2.2 编译源码



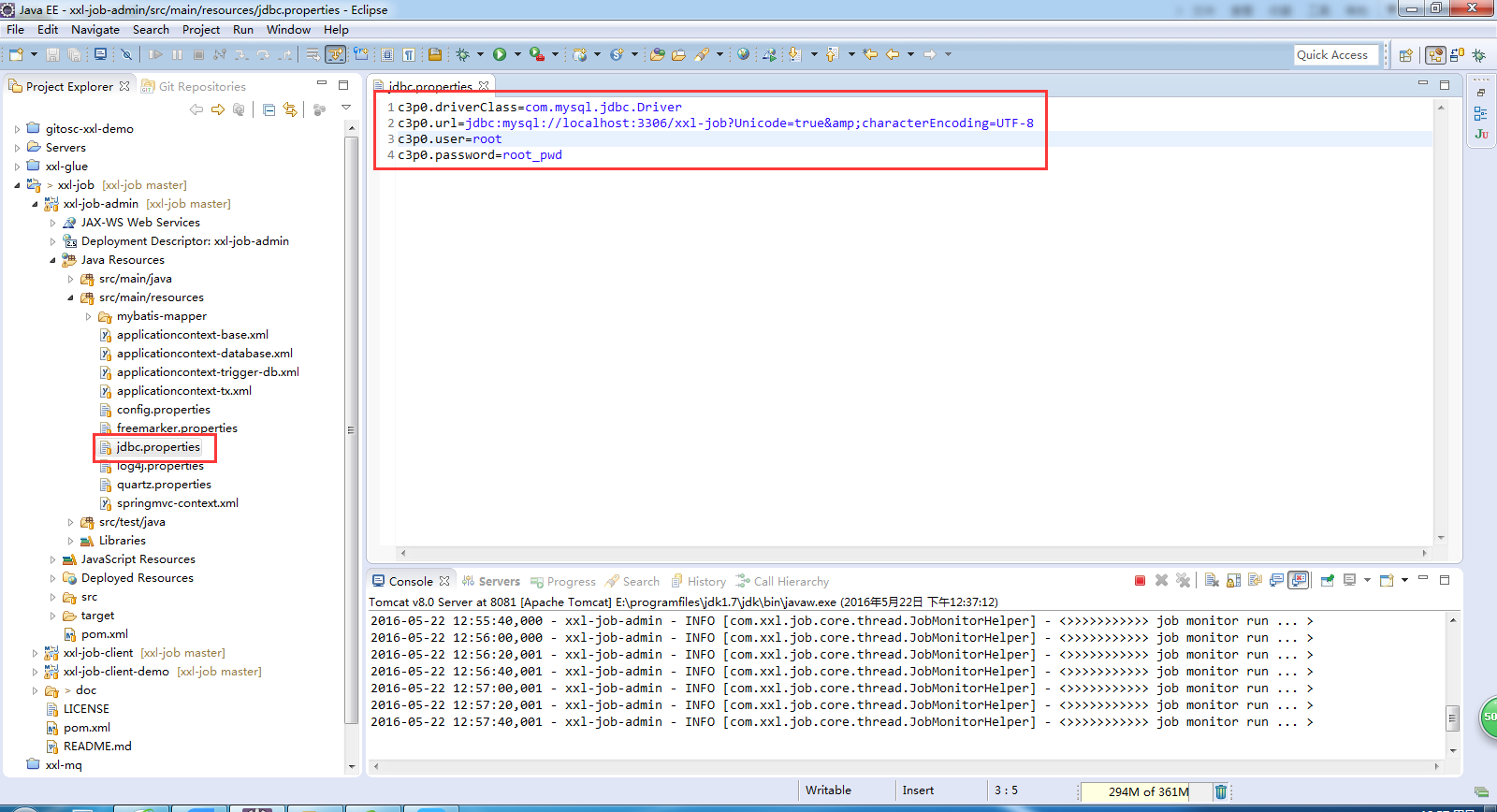
（图2.2：项目源码导入Eclipse截图）

解压源码，按照maven格式将源码导入IDE（文档以Eclipse为例），更新项目pom依赖，maven编译项目。

正常情况下，项目结构应该如图2.2所示，

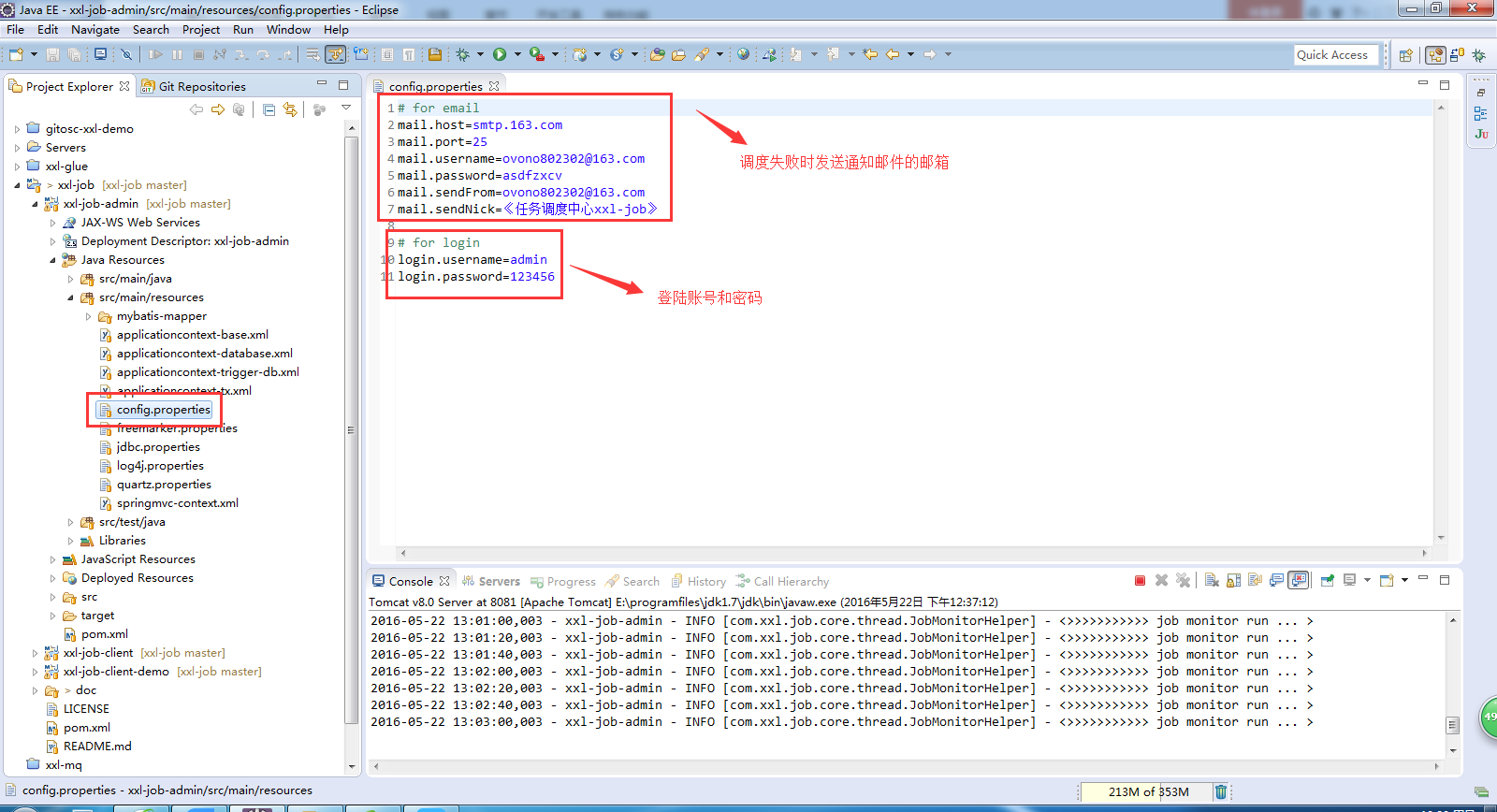
## 2.3 配置部署“调度中心”

**“调度中心”项目：xxl-job-admin**



（图2.2A：数据库JDBC配置截图）

**配置调度中心JDBC链接**：请在图2.2A所示位置配置jdbc链接地址，链接地址请保持和 2.1章节 所创建的调度数据库的地址一致。



（图2.2B：系统配置截图）

**配置报警邮箱和登陆账号**：请在图2.2B所示位置，设置自己的报警邮件发送邮箱和登陆账号密码。



（图2.2C：调度中心部署截图）

**部署项目**：如果已经正确进行上述配置，可将项目部署到eclipse下的tomcat服务器中。或者，将“调度中心”项目导出war包单独部署。

启动项目。



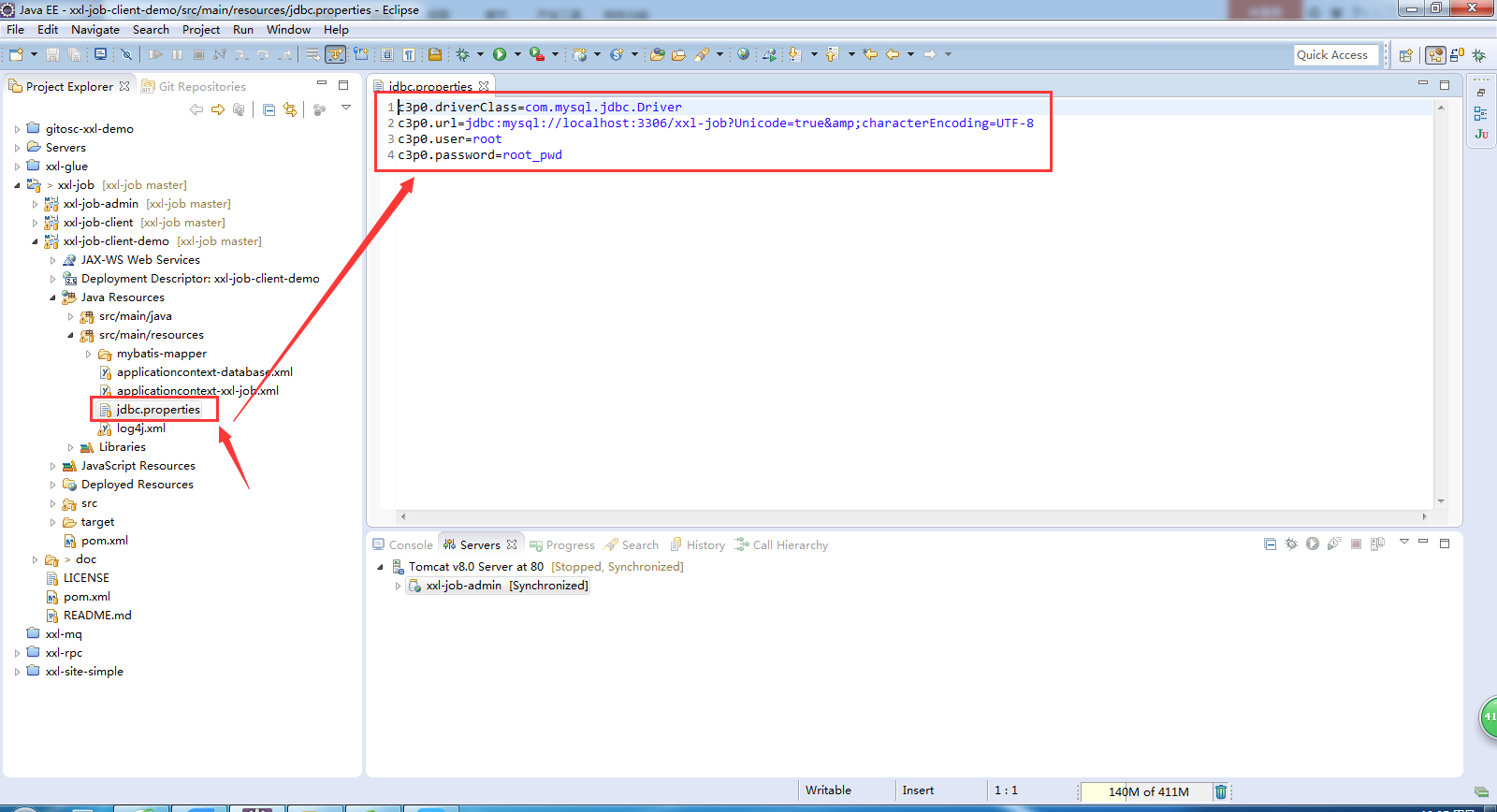
（图2.2D：“调度中心”登陆界面）

**访问链接**：http://localhost:8080/xxl-job-admin/

至此“调度中心”项目已经部署结束。

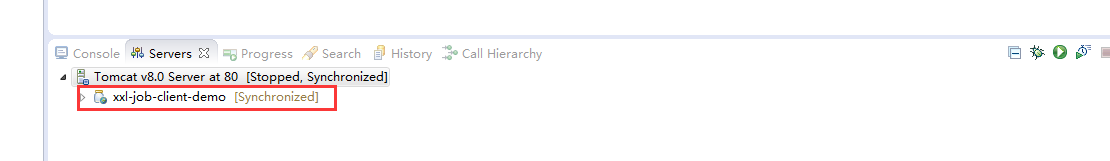
## 2.4 配置部署“执行器”

**“执行器”项目：xxl-job-client-demo**



（图2.4A：“执行器”项目Jdbc链接配置截图）

**配置“执行器”项目Jdbc链接**：请在图2.4A所示位置配置jdbc链接地址，链接地址请保持和 2.1章节 所创建的调度数据库的地址一致。



（图2.4B：“执行器”部署截图）

**部署项目**：如果已经正确进行上述配置，可将项目部署到eclipse下的tomcat服务器中。或者，将“执行器”项目导出war包单独部署。

启动项目。

至此“执行器”项目已经部署结束。

## 2.5 开发第一个任务“Hello World”

本示例为新建一个“GLUE模式任务”（GLUE模式任务的执行代码支持托管到调度中心在线维护，相比Bean模式任务需要在执行器项目开发部署上线，更加简便轻量）。更多有关任务的详细配置，请查看“章节4：任务详解”。

**前提：请确认“调度中心”和“执行器”项目已经成功部署并启动；**



（图2.5A：“调度中心”新建任务按钮）

（图2.5B：“调度中心”任务配置截图）

**步骤一（新建任务）：**登陆调度中心，点击图2.5A所示新建任务按钮，新建示例任务。然后，参考图2.5B进行示例任务参数配置，点击保存。



（图2.5C：“调度中心”管理管理界面，GLUE入口按钮）



（图2.5D：“调度中心”GLUE编辑界面）

**步骤二（GLUE开发）：**请点击图2.5C中所示GLUE入口按钮，进入GLUE开发界面，如图2.5D。GLUE任务默认已经初始化了示例任务代码，即打印Hello World，我自行修改。



（图2.5E：“调度中心”管理管理界面，任务手动执行按钮接口）

**步骤三（触发执行）：**点击图2.5E所示“执行”按钮，可手动触发一次任务执行。



（图2.5F：“调度中心”管理管理界面，任务日志入口）



（图2.5G：“调度中心”管理管理界面，任务日志入口）



（图2.5H：“调度中心”管理管理界面，任务日志入口）

**步骤三（查看日志）：**点击图2.5F所示“日志”按钮，可前往任务日志界面查看任务日志。在如图2.5G的任务日志界面中，可查看任务调度状态，执行器接收到调度请求后的执行状态，同时，点击如图2.5G中的“执行日志”按钮，可以查看本此调度在执行器端的完整执行日志，完整日志如图2.5H。

# 系统详解

## 3.1 源码目录介绍

## 3.1 “调度数据库”详解

## 3.2 “调度中心”详解

## 3.3 “执行器”详解

# 任务详解

## 4.1 BEAN模式任务

## 4.2 GLUE模式任务

# 任务管理

## 5.1 编辑任务信息

## 5.2 编辑GLUE代码

## 5.3 恢复/暂停

## 5.4 手动触发一次调度

## 5.5 查看日志

## 5.6 删除任务

# 设计原理

## 6.1 总体架构

## 6.2 调度原理

## 6.3 通讯原理

## 6.4 执行器原理

# 其他

## 7.1 接入登记

更多接入公司，欢迎在<https://github.com/xuxueli/xxl-job/issues/1> 登记。

## 7.2 报告问题

XXL-JOB托管在Github上，如有问题可在[ISSUES](https://github.com/xuxueli/xxl-job/issues)上提问，也可以加入技术交流群(仅作技术交流)：367260654 。