1. 下载xxl-job源码项目,<https://github.com/xuxueli/xxl-job>
2. 导入eclipse或idea
3. 将tables\_xxl\_job.sql文件导入数据库，修改配置文件
4. 启动项目
5. 对xxl-job-admin项目进行Maven clean，Maven bulid打成jar包
6. 打开命令窗口，执行java -jar命令
7. 创建springboot项目
8. 在pom.xml文件中添加依赖

<dependency>

<groupId>com.xuxueli</groupId>

<artifactId>xxl-job-core</artifactId>

<version>{版本}</version>

</dependency>

1. 修改yml配置文件

|  |
| --- |
| server.port=8081  # log config  #logging.config=classpath:logback.xml  #调度中心部署跟地址 [选填]：如调度中心集群部署存在多个地址则用逗号分隔。执行器将会使用该地址进行"执行器心跳注册"和"任务结果回调"；为空则关闭自动注册；  xxl.job.admin.addresses=http://127.0.0.1:1000/xxl-job-admin  ### access token  xxl.job.accessToken=  ### 执行器AppName [选填]：执行器心跳注册分组依据；为空则关闭自动注册  xxl.job.executor.appname=xxl-job-executor  ### 执行器通讯TOKEN [选填]：非空时启用；  xxl.job.executor.address=  ### 执行器IP [选填]：默认为空表示自动获取IP，多网卡时可手动设置指定IP，该IP不会绑定Host仅作为通讯实用；地址信息用于 "执行器注册" 和 "调度中心请求并触发任务"；  xxl.job.executor.ip=  ### 执行器端口号 [选填]：小于等于0则自动获取；默认端口为9999，单机部署多个执行器时，注意要配置不同执行器端口；  xxl.job.executor.port=9999  ### 执行器运行日志文件存储磁盘路径 [选填] ：需要对该路径拥有读写权限；为空则使用默认路径；  xxl.job.executor.logpath=/data/applogs/xxl-job/jobhandler  ### 执行器日志保存天数 [选填] ：值大于3时生效，启用执行器Log文件定期清理功能，否则不生效；  xxl.job.executor.logretentiondays=30 |

1. 编写配置类

@Configuration

public class XxlJobConfig {

private Logger logger = LoggerFactory.getLogger(XxlJobConfig.class);

@Value("${xxl.job.admin.addresses}")

private String adminAddresses;

@Value("${xxl.job.executor.appname}")

private String appName;

@Value("${xxl.job.executor.ip}")

private String ip;

@Value("${xxl.job.executor.port}")

private int port;

@Value("${xxl.job.accessToken}")

private String accessToken;

@Value("${xxl.job.executor.logpath}")

private String logPath;

@Value("${xxl.job.executor.logretentiondays}")

private int logRetentionDays;

@Bean(initMethod = "start", destroyMethod = "destroy")

public XxlJobSpringExecutor xxlJobExecutor() {

logger.info(">>>>>>>>>>> xxl-job config init.");

XxlJobSpringExecutor xxlJobSpringExecutor = new XxlJobSpringExecutor();

xxlJobSpringExecutor.setAdminAddresses(adminAddresses);

xxlJobSpringExecutor.setAppName(appName);

xxlJobSpringExecutor.setIp(ip);

xxlJobSpringExecutor.setPort(port);

xxlJobSpringExecutor.setAccessToken(accessToken);

xxlJobSpringExecutor.setLogPath(logPath);

xxlJobSpringExecutor.setLogRetentionDays(logRetentionDays);

return xxlJobSpringExecutor;

}

}

1. 测试代码

@Component

@JobHandler(value = "testHandler")

@Slf4j

public class UserJob extends IJobHandler implements Serializable {

@Override

public ReturnT<String> execute(String s){

log.info("统计user开始》》》》》》》》");

XxlJobLogger.log(this.getClass().getSimpleName() + "--start");

return ReturnT.SUCCESS;

}

}

1. 登录界面，测试功能

<http://127.0.0.1:8080/xxl-job-admin> admin/123456

