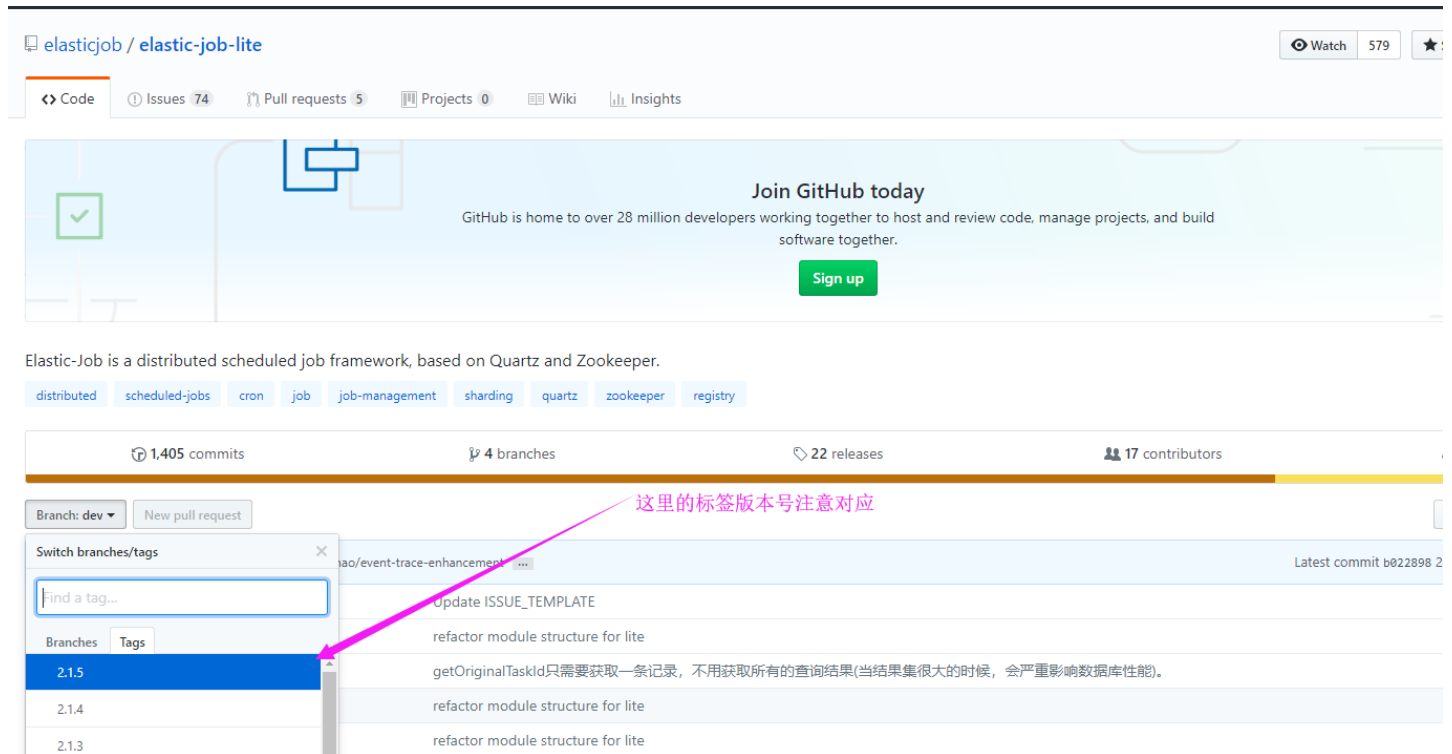


Elastic-Job作业监控平台搭建

获取源码

首先是上github下载elastic-job的源码，目前最新的地址如下：<https://github.com/elasticjob/elastic-job-lite>

下载前要注意源码需要和你使用的elastic-job版本一致。



1. 在你喜欢的终端应用里执行以下命令:

```
git clone https://github.com/elasticjob/elastic-job-lite.git
```

2. 下载完成以后cd到仓库

3. 以PigX所使用的2.1.5版本为例，执行以下命令切换到2.1.5标签的代码:

```
git checkout 2.1.5
```

编译源码

这个过程很简单，执行命令:

```
mvn package -DskipTests
```

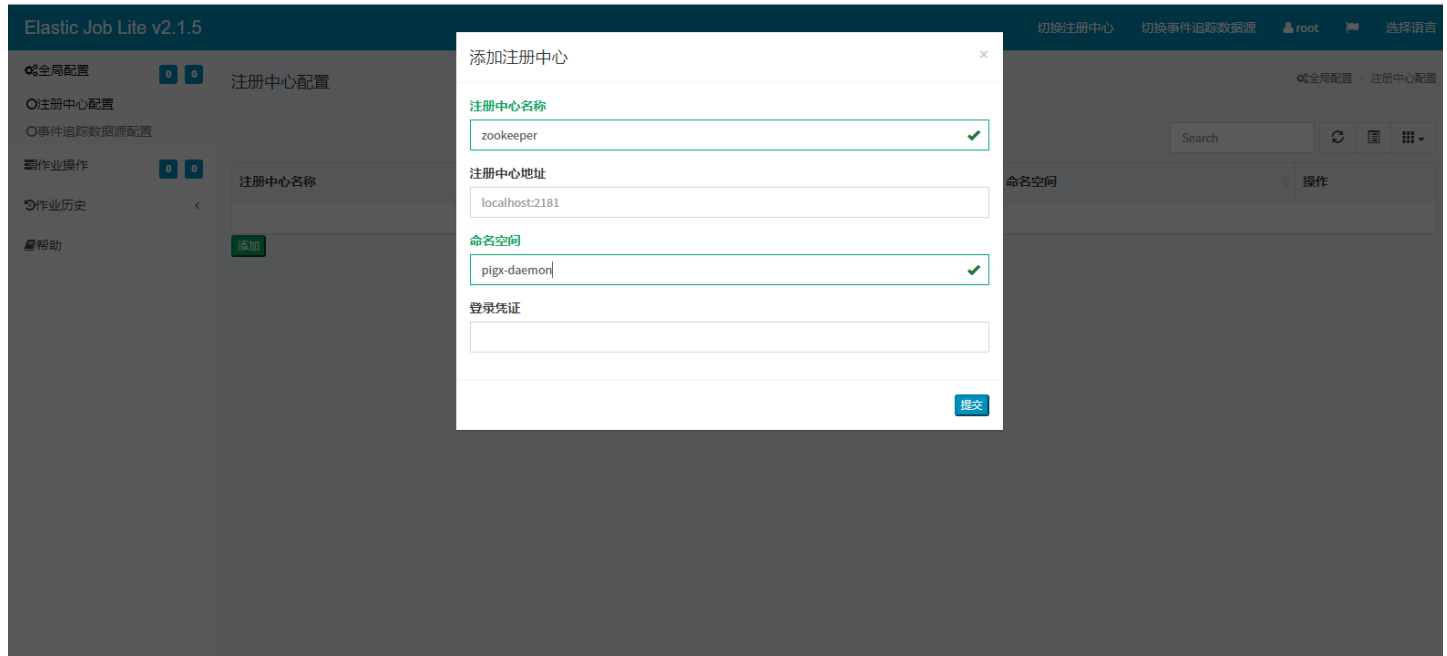
开始使用

1. 进入/elastic-job-lite/elastic-job-lite-console/target文件夹，有一个叫做 `elastic-job-lite-console-2.1.5.tar.gz` 的文件就是我们需要的了。

2. 解压之，执行bin目录下的start脚本，这货默认运行在8899端口，以用户名:root和密码:root进行登录
3. 知道大家可能看不懂，右上角切换一波语言吧！
4. 以pigx最新的1.6.0的稳定版代码为例，查看配置中心下的pigx-daemon-dev.yml文件,与我们目前相关的配置文件如下：

```
## 定时任务
spring:
  # 保存定时任务的数据源
  datasource:
    type: com.zaxxer.hikari.HikariDataSource
    driver-class-name: com.mysql.jdbc.Driver
    username: root
    password: root
    url: jdbc:mysql://127.0.0.1:3306/pigx?
characterEncoding=utf8&zeroDateBehavior=convertToNull&useSSL=false
elasticjob:
  # 分布式任务协调依赖zookeeper
  zookeeper:
    server-lists: 127.0.0.1:2181
    namespace: pigx-daemon
```

通过全局配置-注册中心配置,填入以下参数:



然后修改全局配置-事件追踪数据源配置,填入以下参数:

添加事件追踪数据源

事件追踪数据源名称

随便写个名字

事件追踪数据源驱动

com.mysql.jdbc.Driver

事件追踪数据源连接地址

jdbc:mysql://127.0.0.1:3306/pigx?characterEncoding=utf8&zeroDateTimeBehavior=conv

事件追踪数据源用户名

root

事件追踪数据源密码

测试连接

提交

在操作处点击连接，出现已连的字样表示连接成功。

事件追踪数据源名称	事件追踪数据源驱动	事件追踪数据源连接地址	事件追踪数据源用户名	操作
随便写个名字	com.mysql.jdbc.Driver	jdbc:mysql://127.0.0.1:3306/pigx?characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=false	root	<div>已连</div> <div>删除</div>

最终效果如下:

PigX 微服务

首页 文档 crud实例

admin

系统管理

系统监控

服务监控

接口文档

任务监控

任务轨迹

任务监控

Elastic Job Life v2.1.5

Switch registry center Switch event trace data source root Switch language

Global settings 1 1

Job operation 0 0

Job history

Help

Registry center

Global settings -> Registry center

Search

Name	Zookeeper address	Namespace	Operation
zookeeper	localhost:2181	pigx-daemon	<div>Connect</div> <div>Delete</div>

Showing 1 to 1 of 1 rows

Add