

客户端接入

1 部署前准备

1.1 软件准备

JDK >= 1.7

· 网络权限（服务器期间调用）

开发 <http://192.168.180.92:9088>

测试1 <http://saturnconsole.kjtpay.c1prod:9088>

测试2 <http://saturnconsole.kjtpay.c2prod:9088>

准生产 <http://saturnconsole.kjtpay.zprod:9088>

生产 <http://saturnconsole.kjtpay.tprod:9088>

1.2 其他要求

已迁移成单应用单服务器。

2 服务器上初始化saturn-executor 执行器

2.1 下载安装executor

2.1.1 从<https://github.com/vipshop/Saturn/releases> 中点击链接获取最新版本的'Executor Zip File'，将得到一个saturn-executor-{version}-zip.zip的文件，存放至系统的指定目录。

[saturn-executor-3.0.1-zip.zip](#)

2.1.2 设置环境变量

linux: export SATURN_ZIP_FILE= /opt/soft/saturn-executor-3.0.1-zip.zip 存放目录地址

window: 系统数据->高级->环境变量-> 用户变量 SATURN_ZIP_FILE，值 saturn-executor-3.0.1-zip.zip 存放地址，并且重启下 eclipse使环境变量生效

2.1.3 服务器saturn-executor 部署目录

程序默认写死部署目录： /opt/app/saturn

3.客户端接入

pom引入saturn的jar包

```
<dependency>
  <groupId>com.vip.saturn</groupId>
  <artifactId>saturn-job-api</artifactId>
  <version>3.0.1</version>
</dependency>
<dependency>
  <groupId>com.kjtpay.saturn</groupId>
  <artifactId>kjtpay-saturn-embed</artifactId>
  <version>1.0.2</version>
</dependency>
```

spring配置文件配置

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

  <bean id="demoService" class="com.vip.saturn.demo.embed.tomcat.service.DemoService"></bean>
  <!-- 注册作业 -->
  <bean id="demoJob" class="com.vip.saturn.demo.embed.tomcat.job.DemoJob"></bean>

  <!-- 加载saturn-executor -->
  <import resource="classpath:META-INF/spring/kjtpay-saturn-bootstrap.xml" />

</beans>
```

配置文件里配置

saturn.app.namespace=xxx.kjtpay.com //xxx 为cmdb 维护的系统名称

saturn.app.executorName=xxx-executor-embed

saturn.console.url=<http://192.168.180.92:9088>

saturn.app.executor.logUrl=/opt/logs/saturn

修改原有的Job类 继承AbstractSaturnJavaJob

```

package com.vip.saturn.demo.embed.tomcat.job;

import com.vip.saturn.demo.embed.tomcat.service.DemoService;
import com.vip.saturn.demo.embed.tomcat.utils.SpringContextUtils;
import com.vip.saturn.job.AbstractSaturnJavaJob;
import com.vip.saturn.job.SaturnJobExecutionContext;
import com.vip.saturn.job.SaturnJobReturn;
import org.springframework.beans.factory.annotation.Autowired;

public class DemoJob extends AbstractSaturnJavaJob{

    @Autowired
    private DemoService demoService;

    @Override
    public SaturnJobReturn handleJavaJob(String jobName, Integer shardItem, String shardParam,
SaturnJobExecutionContext shardingContext) {
        demoService.execute();
        System.out.println("running, " + jobName + "; " + shardItem + ";" + shardParam);
        return new SaturnJobReturn("我是分片" + shardItem + "的处理结果");
    }
    /**
     * 这是个静态方法，在executor初始化时会调用，并生成供saturn使用的实现类对象
     */
    public static Object getObject() {
        return SpringContextUtils.getBean("demoJob",DemoJob.class);
    }
}

```

服务类

```

package com.vip.saturn.demo.embed.tomcat.service;

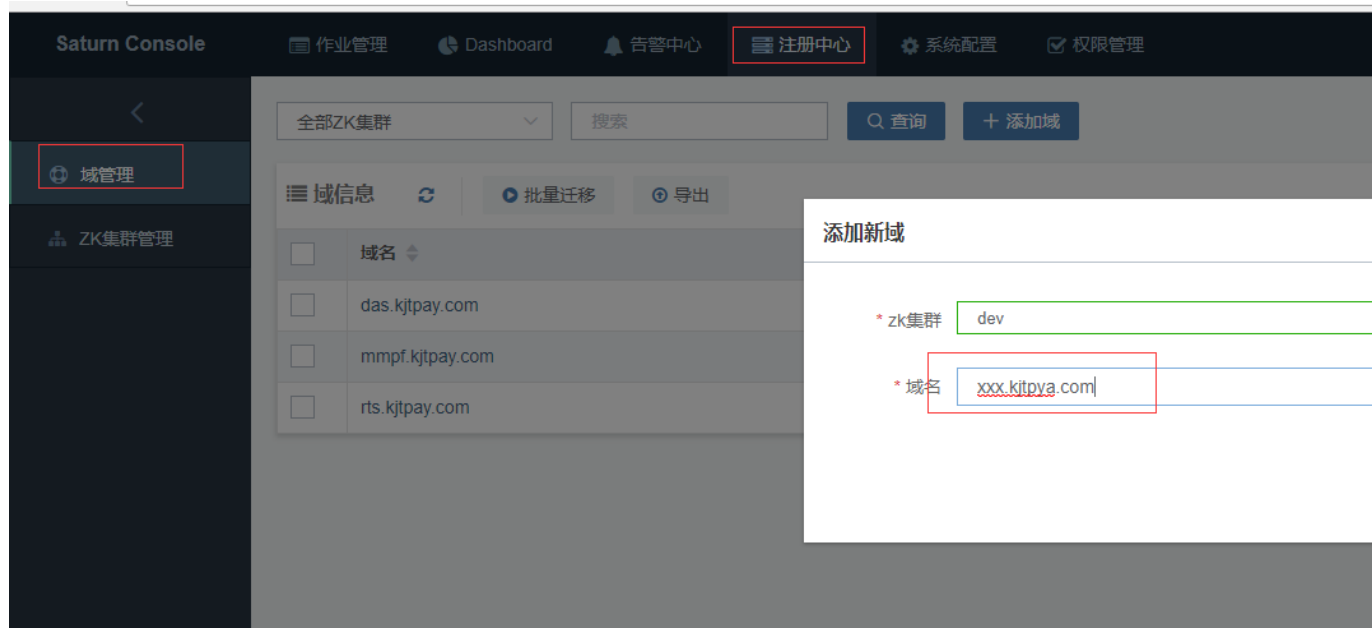
public class DemoService {
    public void execute() {
        // do business service
        System.out.println(".....");
    }
}

```

4.在Console添加一个Java作业，作业实现类必须是你所实现的Java作业的className。

开发环境地址<http://192.168.180.92:9088>，账户admin/admin

首先确保应用的 namespace 已存在，没有新建 xxx.kjtpay.com (xxx为cmdb中维护的系统名称)



创建完成后点击对应的域名进入



点击添加

添加作业



* 作业类型	Shell定时作业	
* 作业名	echoShellJob	
* cron表达式	0/5 * * * * ?	预测
	1. 每10秒运行一次的表达式：*/10 * * * * ? 2. 每小时运行一次的表达式：0 * * * * ?	
作业分片总数	1	^ v
* 分片参数	0=/apps/sh/demojob.sh	
自定义参数		
时区	Asia/Shanghai	
描述		

取消

确定

- 作业类型：分为Java定时作业和Shell定时作业，这里选择Java定时作业
- 作业名：作业ID标识，namespace下必须唯一
- 作业实现类：作业实现类的完整包名+类名
- cron表达式：作业定时表达式
- 作业分片总数：表示并发执行的数量，2代表该作业同时有两个进程在并发执行，每个进程都有自己专门的脚本和参数(这些进程可能同跑在不同机器上的)。
- 分片序号/参数对照表：定义每个分片执行的完整脚本路径及参数。这是saturn最重要的参数之一。分片号从0开始，最大为分片总数-1。
- 作业描述信息 (Optional)：作业描述

更多高级的配置，可以在作业编辑页面进行编辑。（如何作业是临时性的，配置成临时性作业），使注册中心里的，一键启用不对该作业生效

Saturn Console

作业管理Dashboard告警中心注册中心系统配置权限管理

当前域: aml.kjtpay.com 所属集群: dev

<

作业: AmlCheckJob 运行状态: 已停止

更新

作业设置

分片情况

运行状态

告警中心

基本配置

高级配置

超时告警(秒)0

超时强杀(秒)

所属分组

作业负荷1

统计处理间隔(秒)

时区Asia/Shanghai

控制台输出日志

上报运行状态

依赖作业请选择

执行结果发送的

然后启动这个作业

当前域: mydomain 所属集群: 集群1

<

作业: echoShellJob 运行状态: 已停止

更新

作业设置

分片情况

运行状态

告警中心

基本配置

* Cron0/5 * * * * ?

1. 每10秒运行一次的表达式: */10 * * * *

2. 每小时运行一次的表达式: 0 * * * * *

如果一切正常会在IDE的console看到作业运行的日志，也可以在“分片”标签看到执行的结果。（当然，前提是作业到点执行了）

下面是执行成功看到的日志：

```
[2017-12-13 15:09:45.206] [INFO] [executor_001_echoJavaJob-saturnQuartz-worker]
[com.vip.saturn.job.basic.AbstractSaturnJob] >>> [echoJavaJob] msg=Job echoJavaJob handle items: [0, 1]
[2017-12-13 15:09:45.206] [INFO] [Saturn-echoJavaJob-1-thread-1] [com.vip.saturn.job.java.SaturnJavaJob]
>>> [echoJavaJob] msg=Running SaturnJavaJob, jobClass is com.test.DemoJob
[2017-12-13 15:09:45.207] [INFO] [Saturn-echoJavaJob-1-thread-3] [com.vip.saturn.job.java.SaturnJavaJob]
>>> [echoJavaJob] msg=Running SaturnJavaJob, jobClass is com.test.DemoJob
[2017-12-13 15:09:45.207] [INFO] [executor_001_echoJavaJob-saturnQuartz-worker]
[com.vip.saturn.job.basic.AbstractSaturnJob] >>> [echoJavaJob] msg=echoJavaJob finished, totalCost=1,
return={0=SaturnJobReturn [returnCode=0, returnMsg=0, errorGroup=200,
prop={SATURN_TRACE_ID=4882587783541623202, SATURN_PARENT_SPAN_ID=1611817686233844130}], 1=SaturnJobReturn
[returnCode=0, returnMsg=1, errorGroup=200, prop={SATURN_TRACE_ID=4418419319829081425,
SATURN_PARENT_SPAN_ID=3024241466322103633}]}
```

下面是Console看到的结果：

🏠 当前域: mydomain

🏠 所属集群: 集群1

⚠️

⏪

🔧 作业设置

📋 分片情况

🔄 运行状态

🔧 作业 : echoShellJob

运行状态: 已就绪

📋 分片情况

🔄

Executor	IP	分片项
📱 🇺🇸 executor_001	10.199.172.93	0