看了activiti官网使用文档，看了前两章还挺舒服的，看到第三章，突然开始讲配置了，一脸懵逼呀有木有，虽然部署好了activiti-app.war，但是还是不清楚怎么进行部署，启动，给事件分配表单等。经过几天的探索和搜索资料，现在将官网demo应用activiti-app.war的使用做了总结，并做了一点点的扩展-加入自定义的spring-mvc配置，以访问自定义的controller，实现些自定义的业务

一、集成activiti-app到自己项目中

1.1 解压activiti-app.war，并将其中的前端文件和libs中的jar包copy到自己项目中

1.2、将web.xml 也copy到自己项目中

1.3 在resources目录下新建spring-mvc.xml和META-INF/activiti-app/activiti-app.properties

spring-mvc.xml配置如下

<?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"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd">

<!-- 自动扫描且只扫描@Controller -->

<context:component-scan base-package="com.lab.controller" use-default-filters="false">

<context:include-filter type="annotation" expression="org.springframework.stereotype.Controller" />

</context:component-scan>

<mvc:annotation-driven />

<mvc:default-servlet-handler />

</beans>

activiti-app.properties内容入下（需要修改为自己数据库相应的配置）

datasource.driver=com.mysql.jdbc.Driver

datasource.url=jdbc:mysql://192.168.34.36:3306/activiti6ui?useUnicode=true&characterEncoding=utf8&useSSL=false

datasource.username=root

datasource.password=root

datasource.max-idle-time=100

#hibernate.dialect=org.hibernate.dialect.H2Dialect

hibernate.dialect=org.hibernate.dialect.MySQLDialect

#hibernate.dialect=org.hibernate.dialect.Oracle10gDialect

#hibernate.dialect=org.hibernate.dialect.SQLServerDialect

#hibernate.dialect=org.hibernate.dialect.DB2Dialect

#hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect

1.4 在web.xml中加入如下配置，所有以/lab/开头的请求，都会走spring-mvc.xml中配置的controller

web.xml配置如下

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<listener>

<listener-class>org.activiti.app.servlet.WebConfigurer</listener-class>

</listener>

<servlet>

<servlet-name>springServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath\*:spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springServlet</servlet-name>

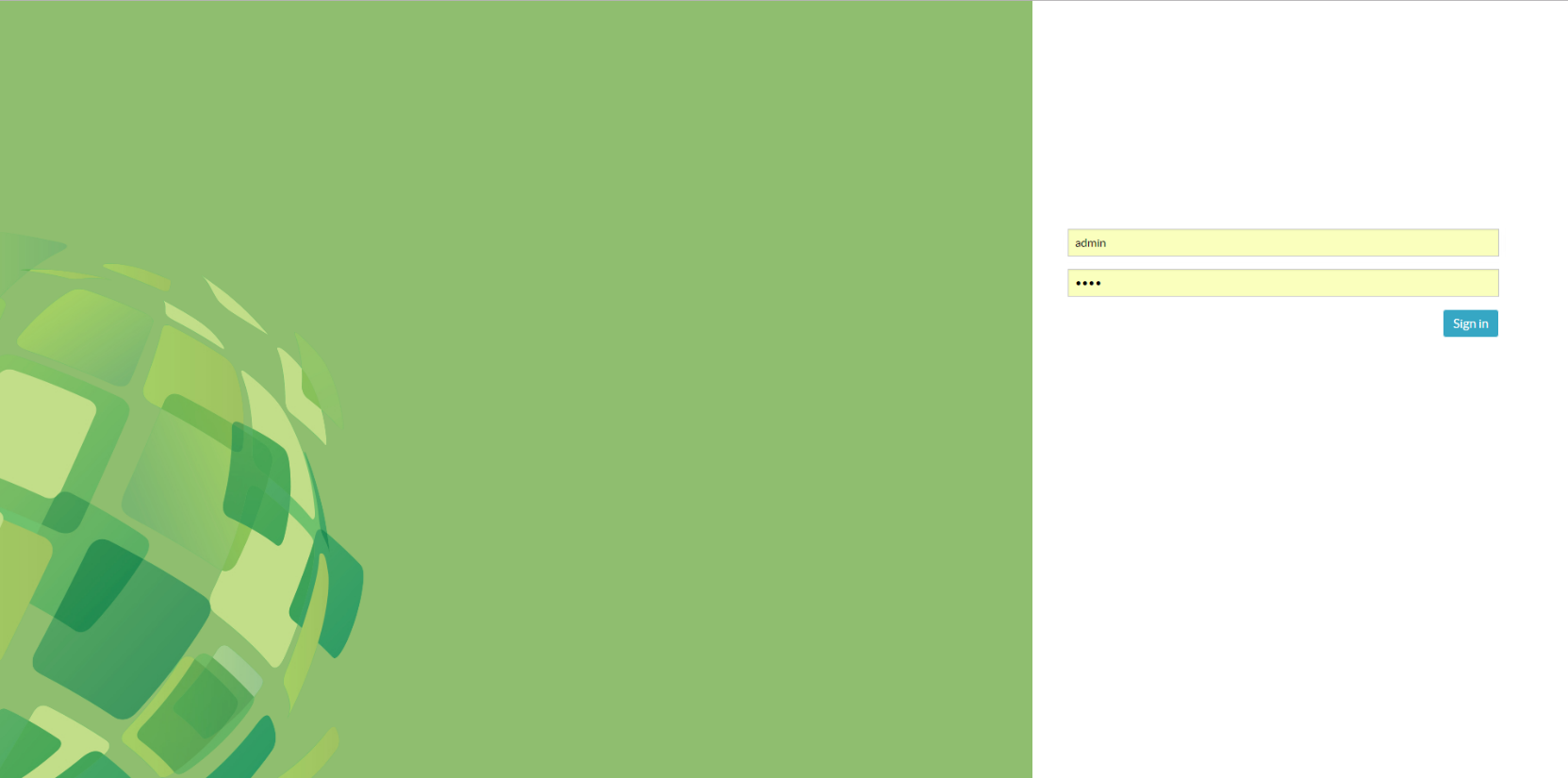
<url-pattern>/lab/\*</url-pattern>

</servlet-mapping>

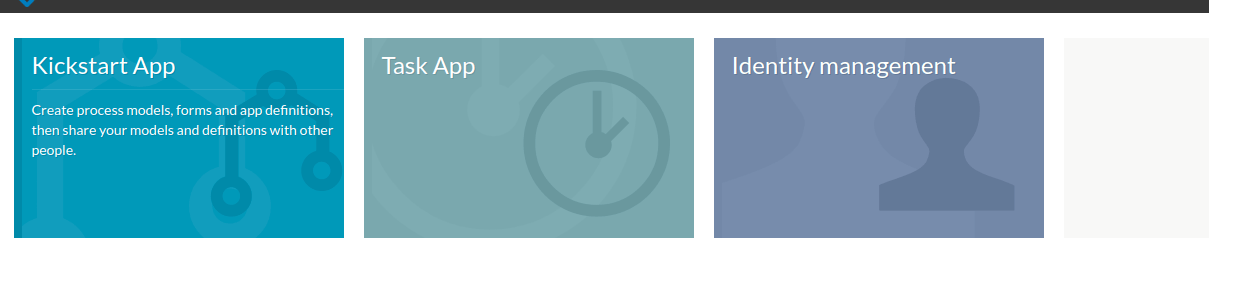
</web-app>

二、activiti-app使用

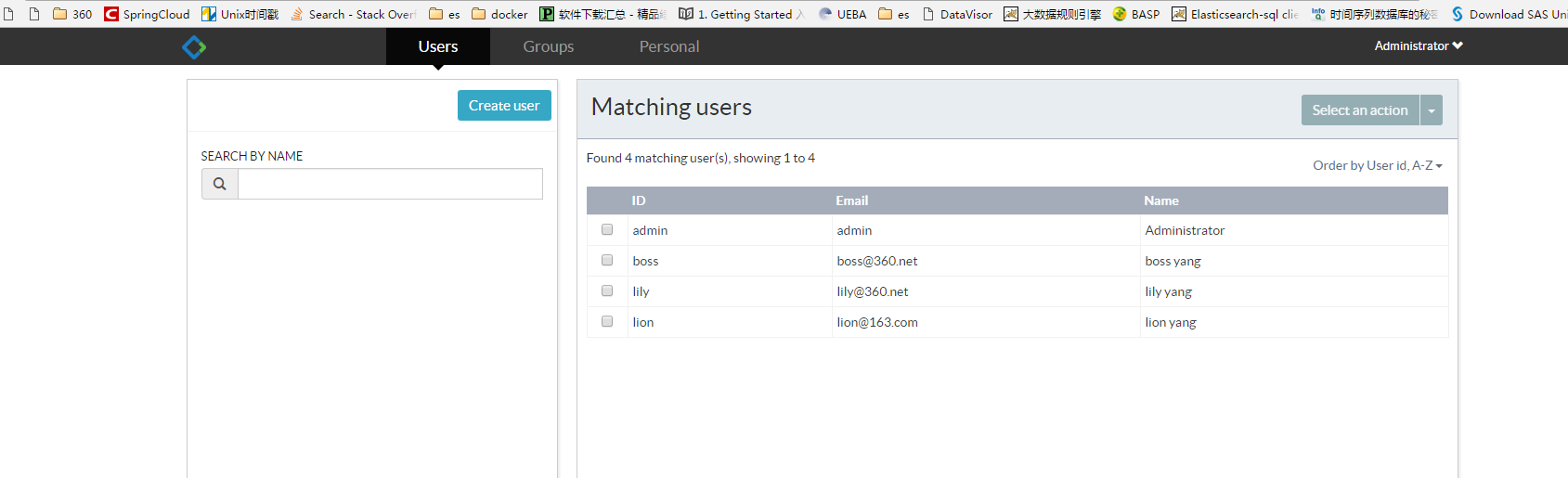
首先启动应用，访问 http://localhost:8080/activiti-app，进入到登录页面



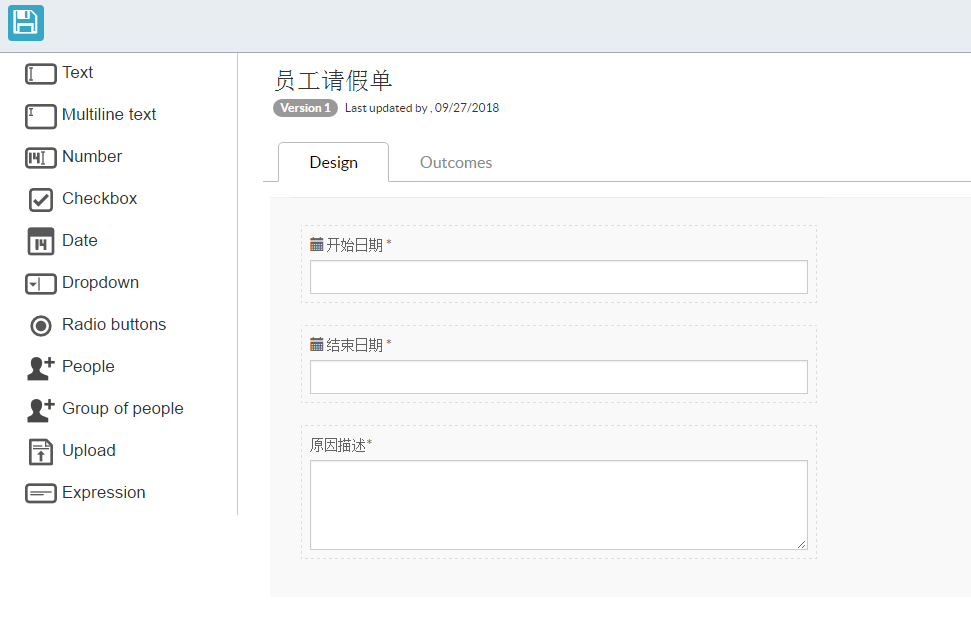
输入admin/test，就可以进入到系统首页

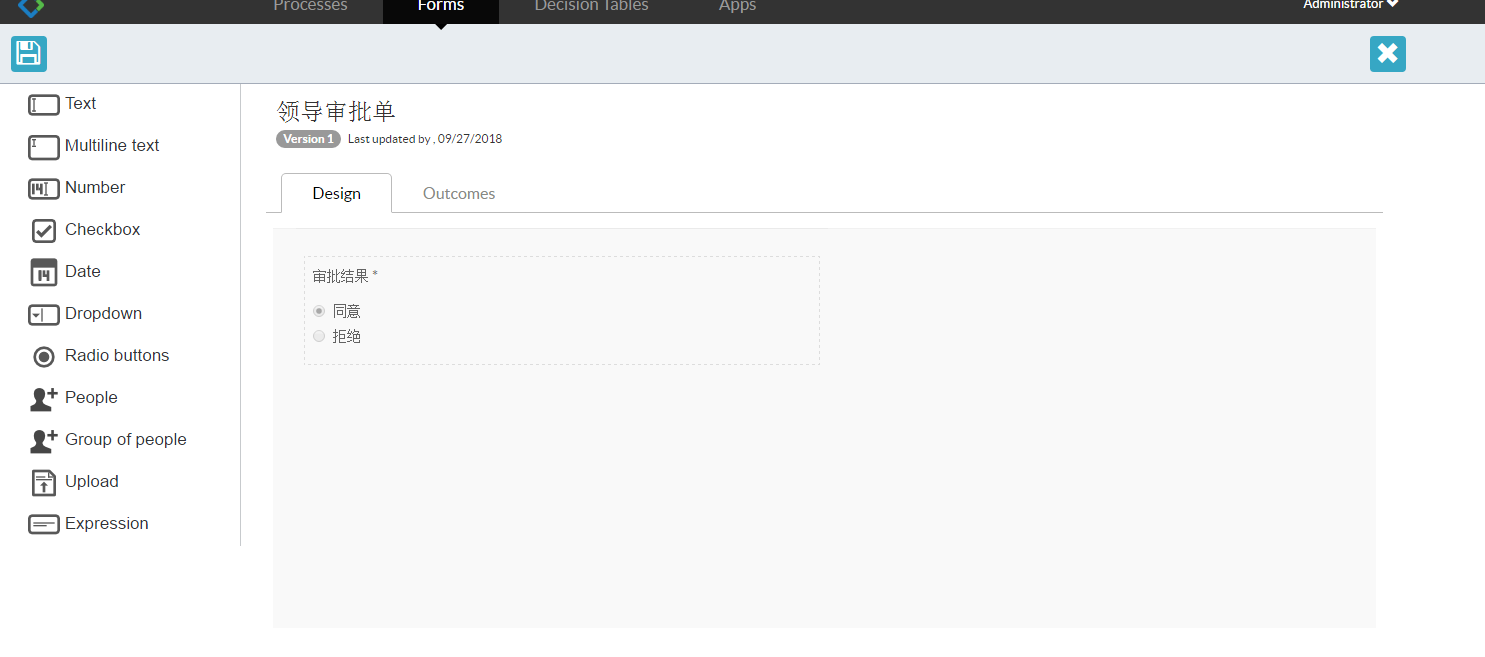


首先点击菜单identity management，进入如下页面，在该模块中，创建两个用户lily和boss，如图所示

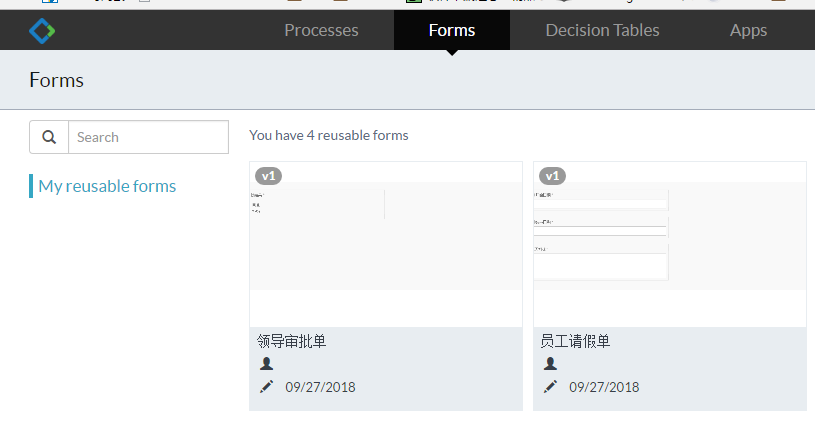


然后进入到KickStart app，点击进入kickstart App后，可以创建和导入流程定义和表单等，我们先创建两个表单：一个**请假表单**，一个老板**批假表单**，点击forms菜单，分别创建两个表单如下



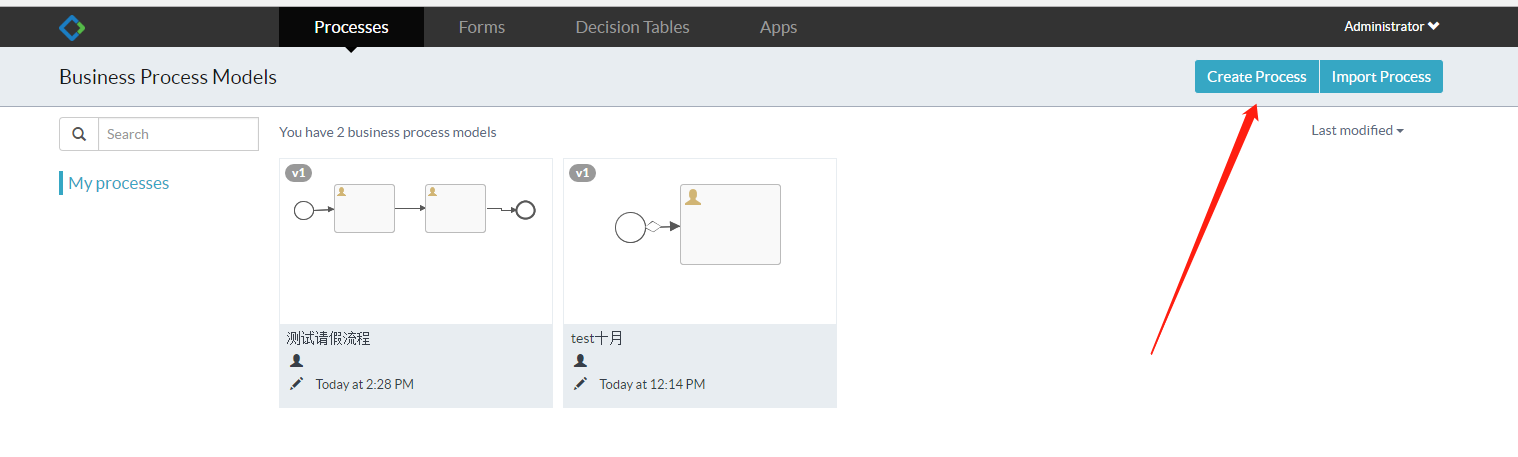


创建完成后，点击forms,可以看到如下列表

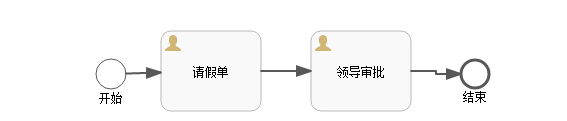


接着，我们来定义一个最简单的请假流程：lily填写假单，boss批假流程。

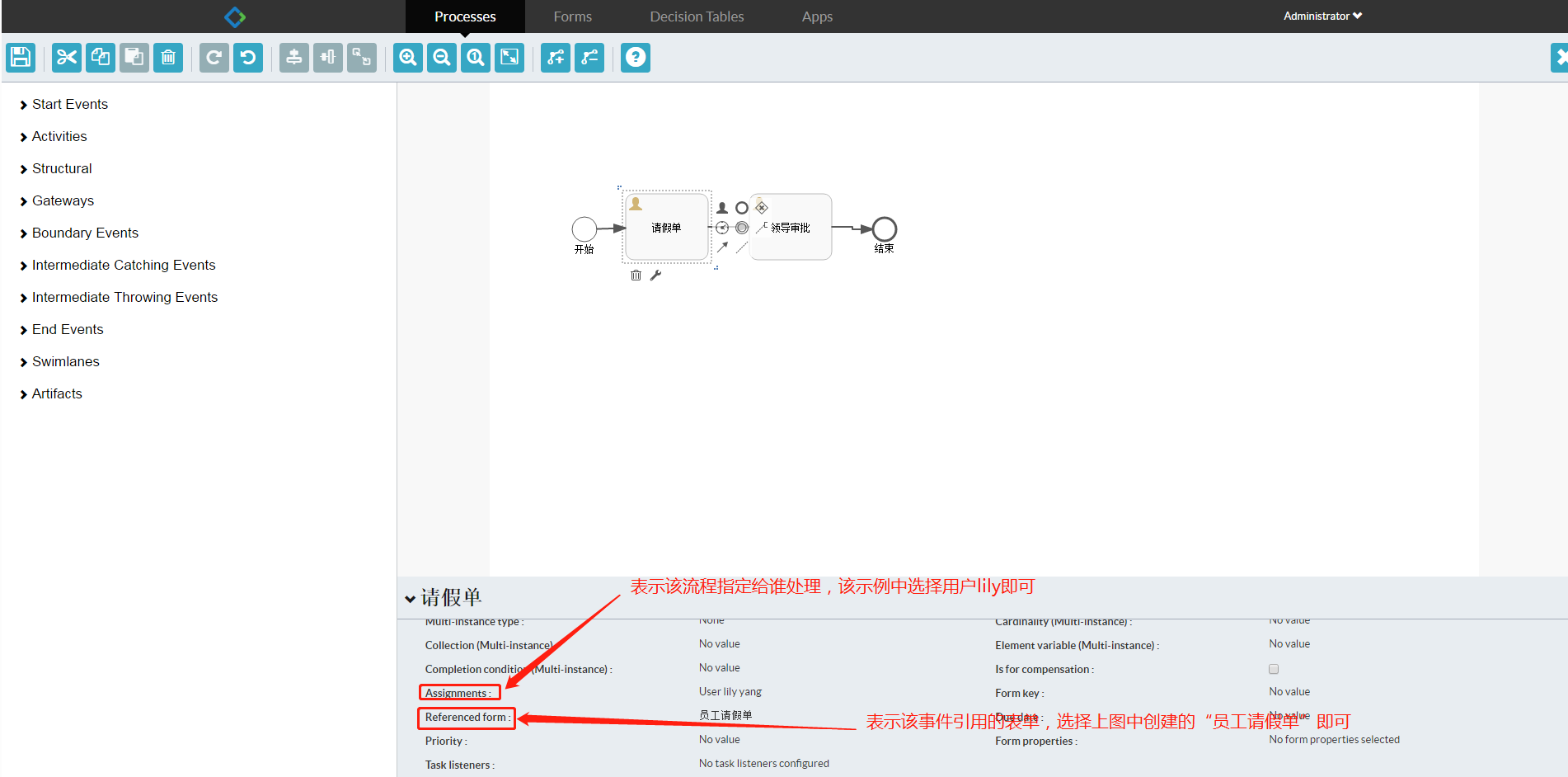
点击processes菜单，然后点击右上角的create process按钮，就可以进入到流程编辑器了



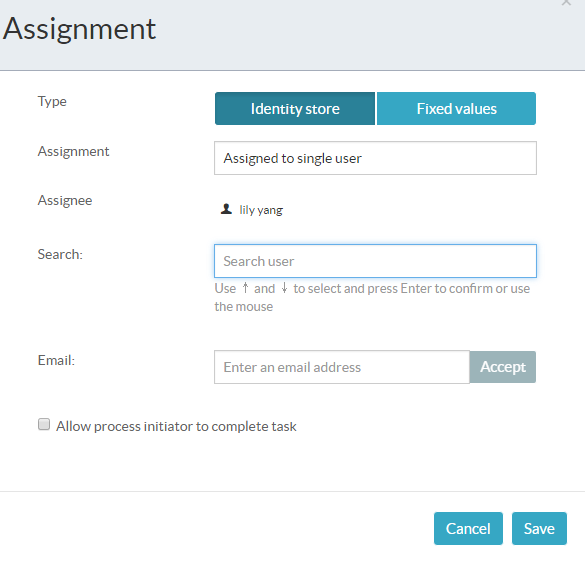
流程定义编辑器打开后，就可以设计自己的流程了

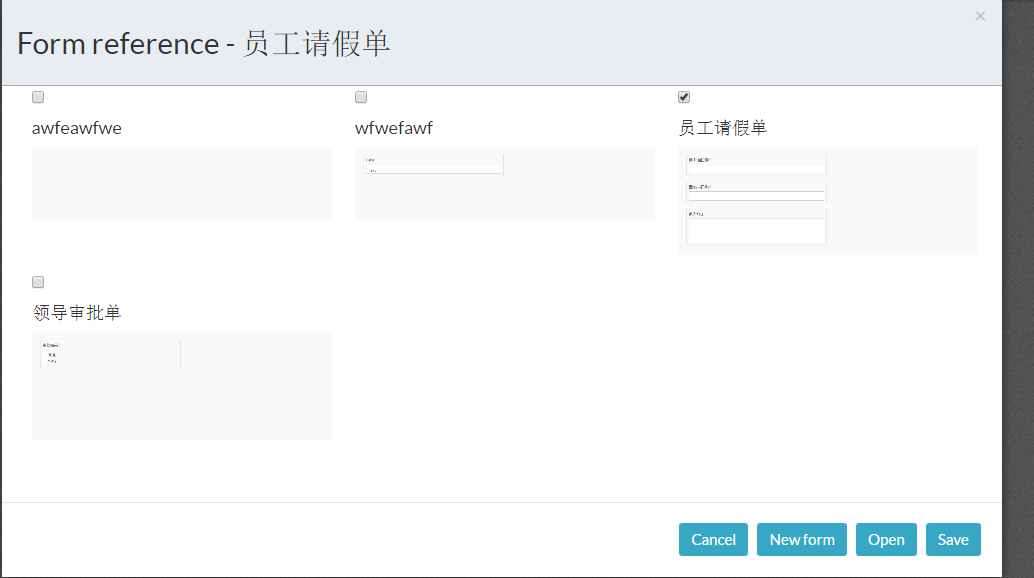


单击"**请假单**"事件时，可以看到如下选项



单击Assignments时，显示如下图所示：



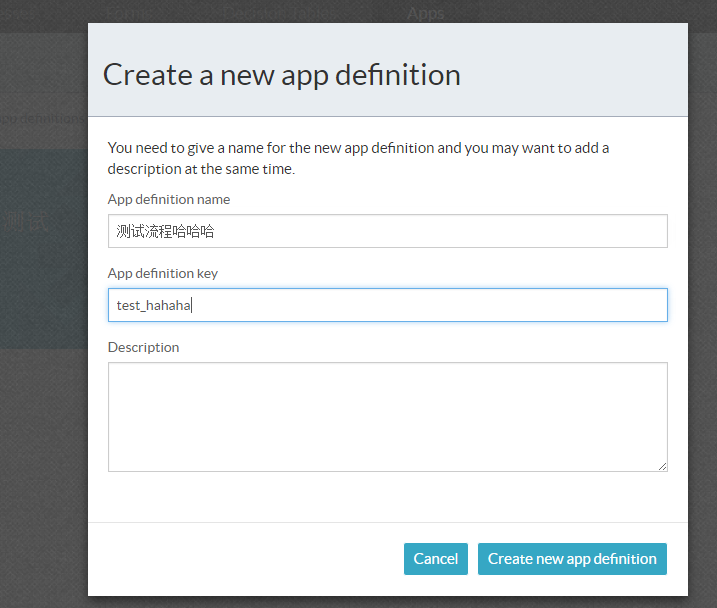
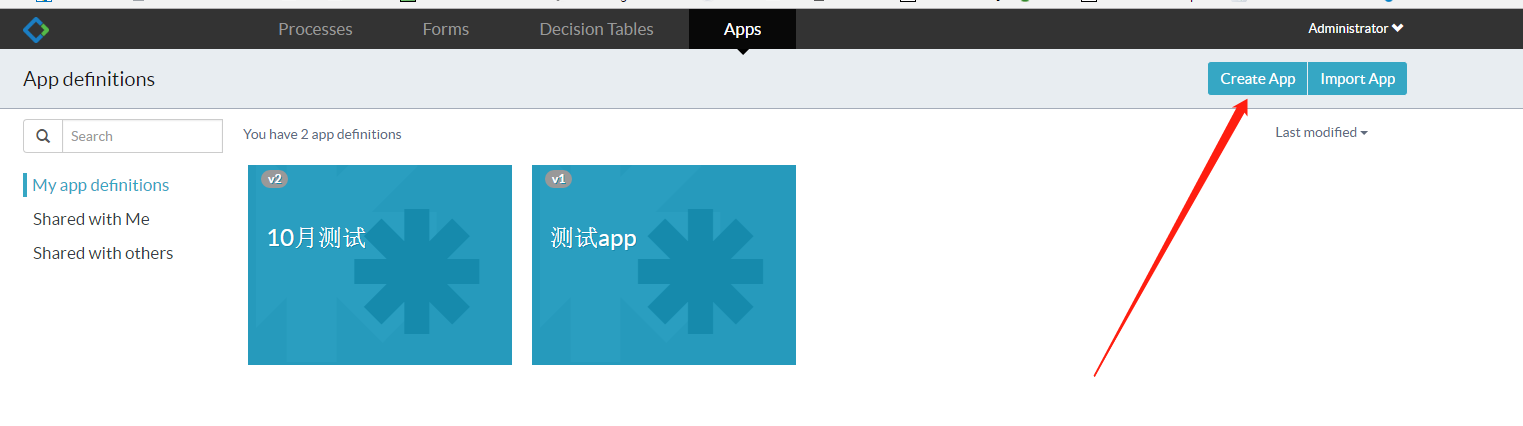
单击*Referenced Form，显示如下图所示*

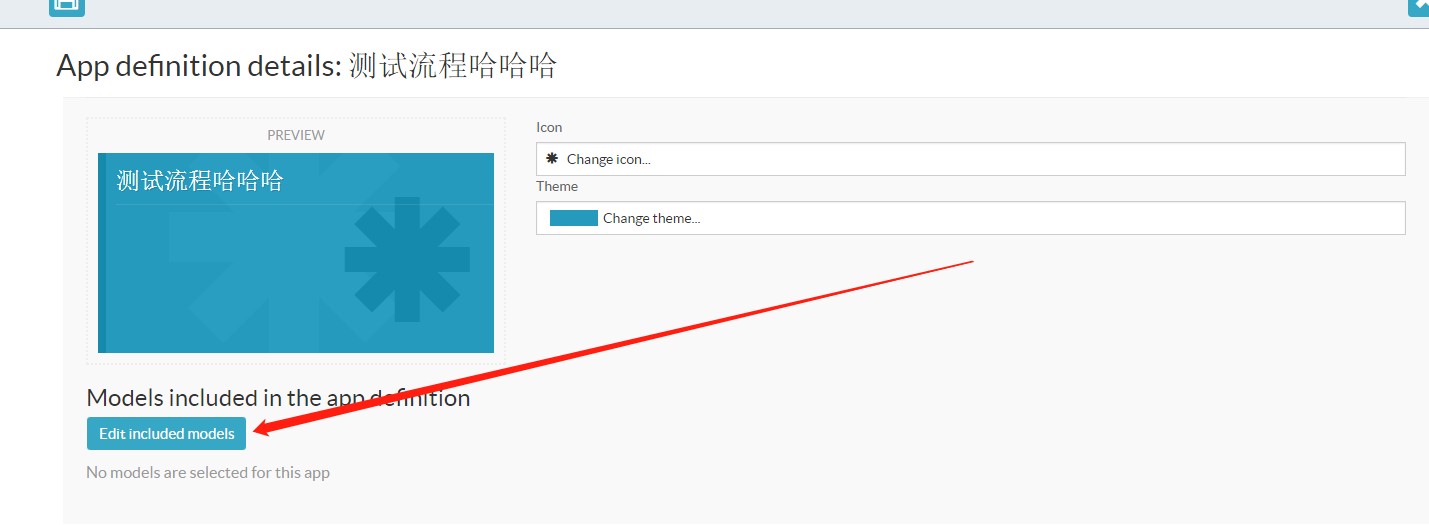
选择“员工请假单”即可

同理可以完成“领导审批”事件对应的assignments和*Referenced Form设置，设置完成后，保存即可*

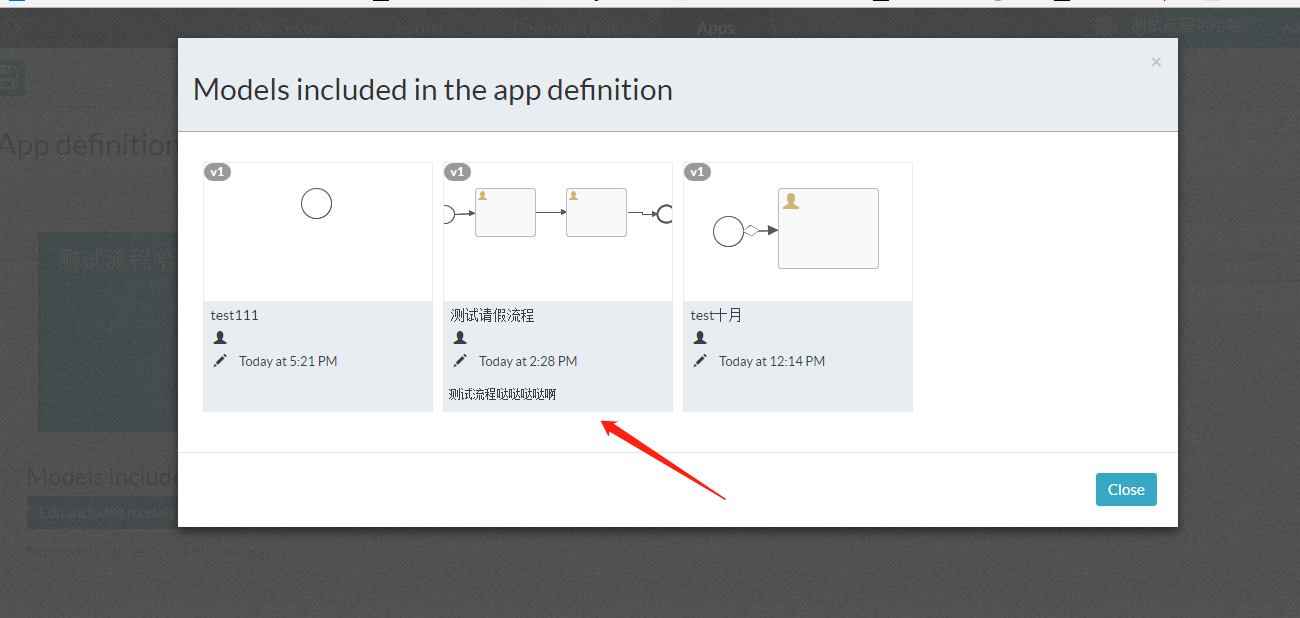
*至此，流程定义已完成，接下来我们要完成部署和启动该流程。*

接着上篇文字的创建自定义表单和流程，本文主要介绍怎么发布、启动和处理流程

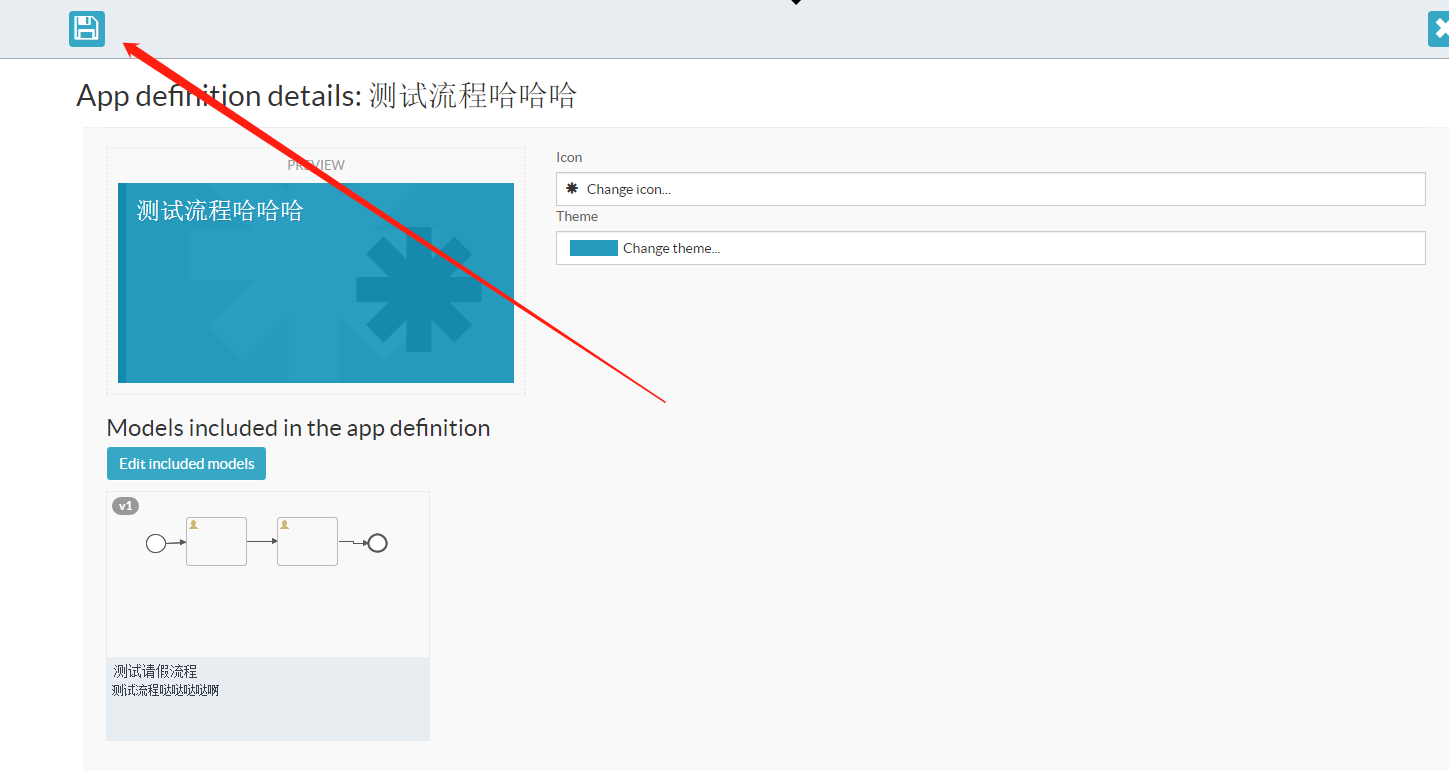
进入kickstart app，点击Apps菜单，然后点击右上角的create App按钮，创建个自己的app



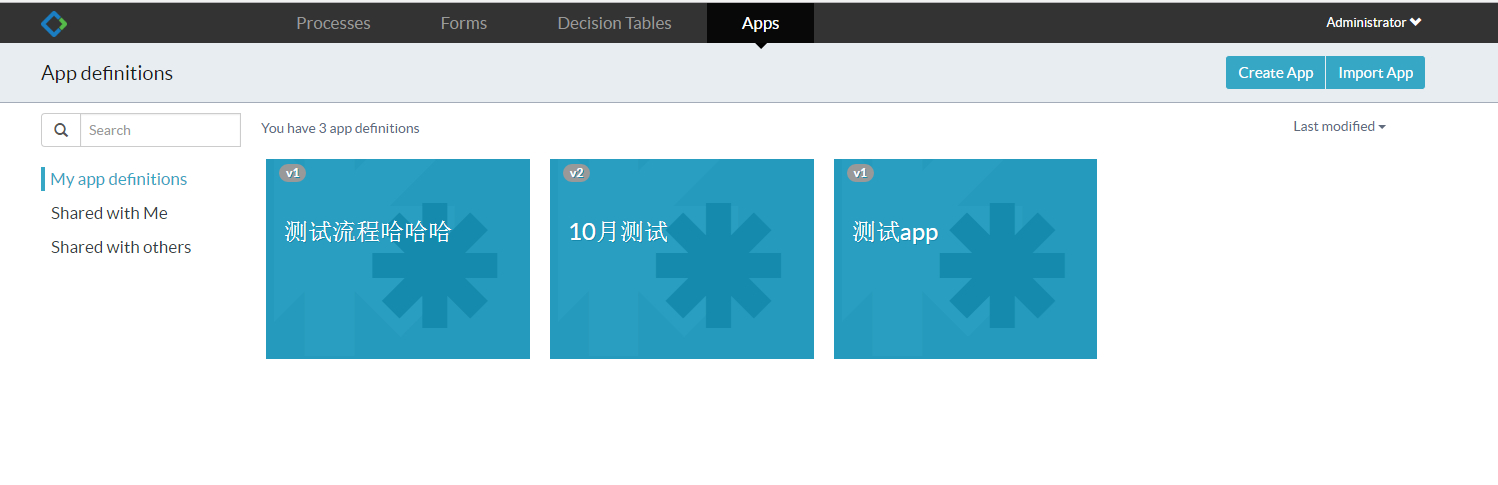
选择上篇文章中创建的流程定义，如下图所示：



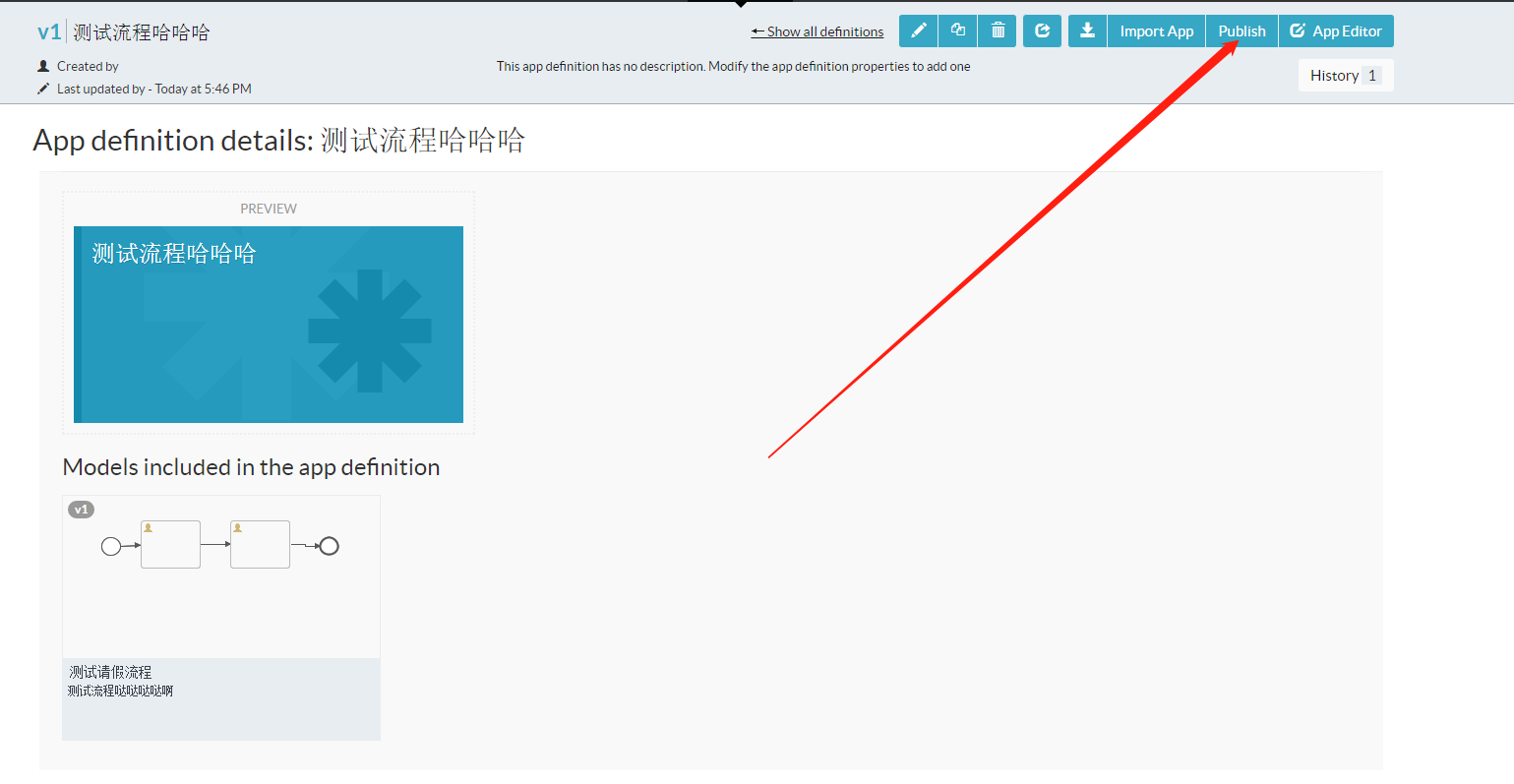
别忘了点击保存



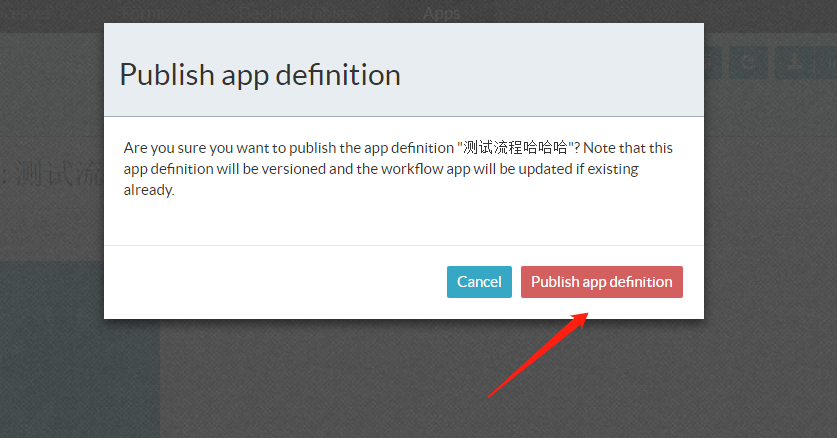
保存后，可以看到所有自己定义的app列表如下

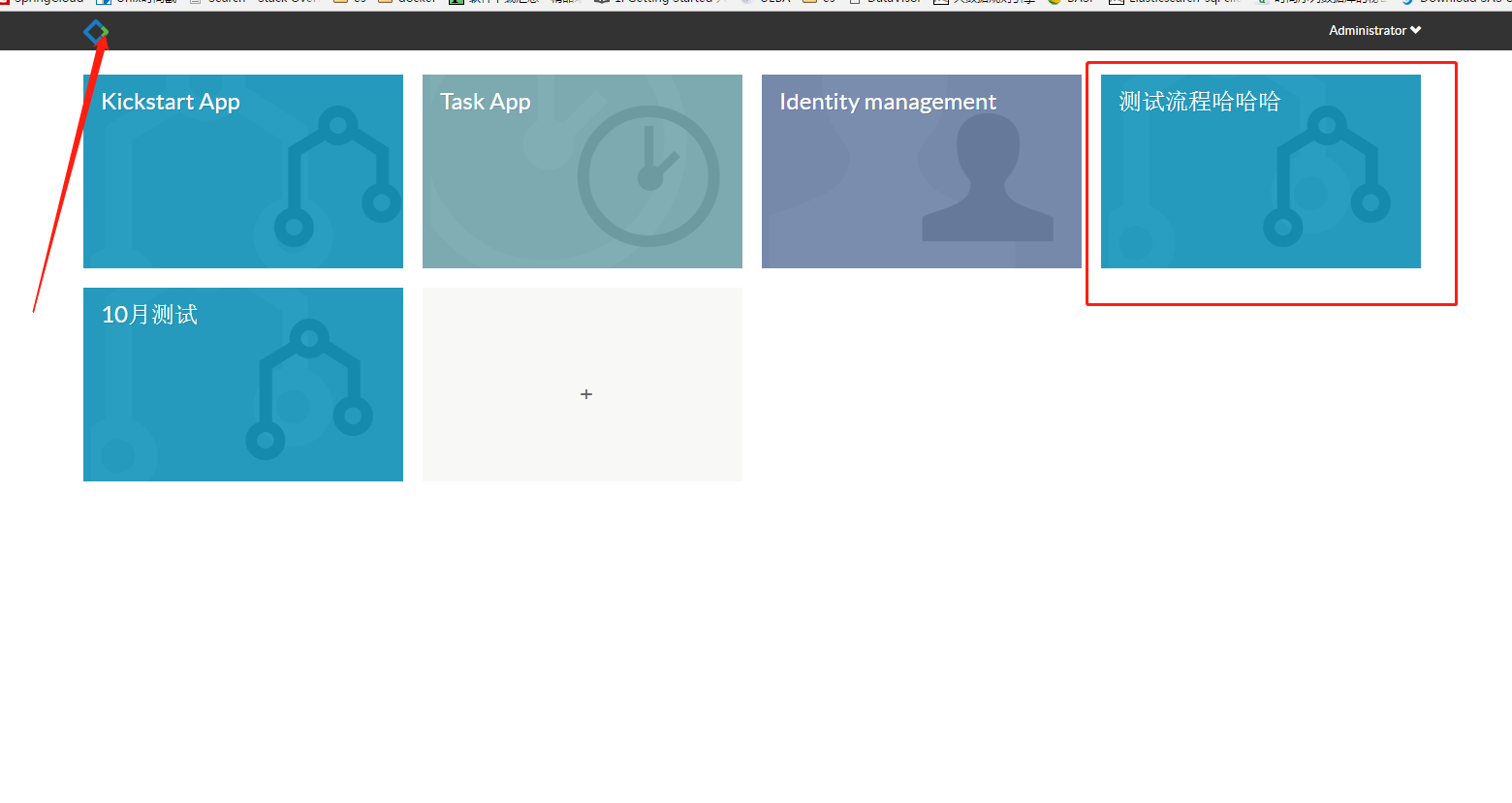


选择刚刚创建的这个app，点击进入，然后点击右上角的publish按钮，即可将该流程进行发布



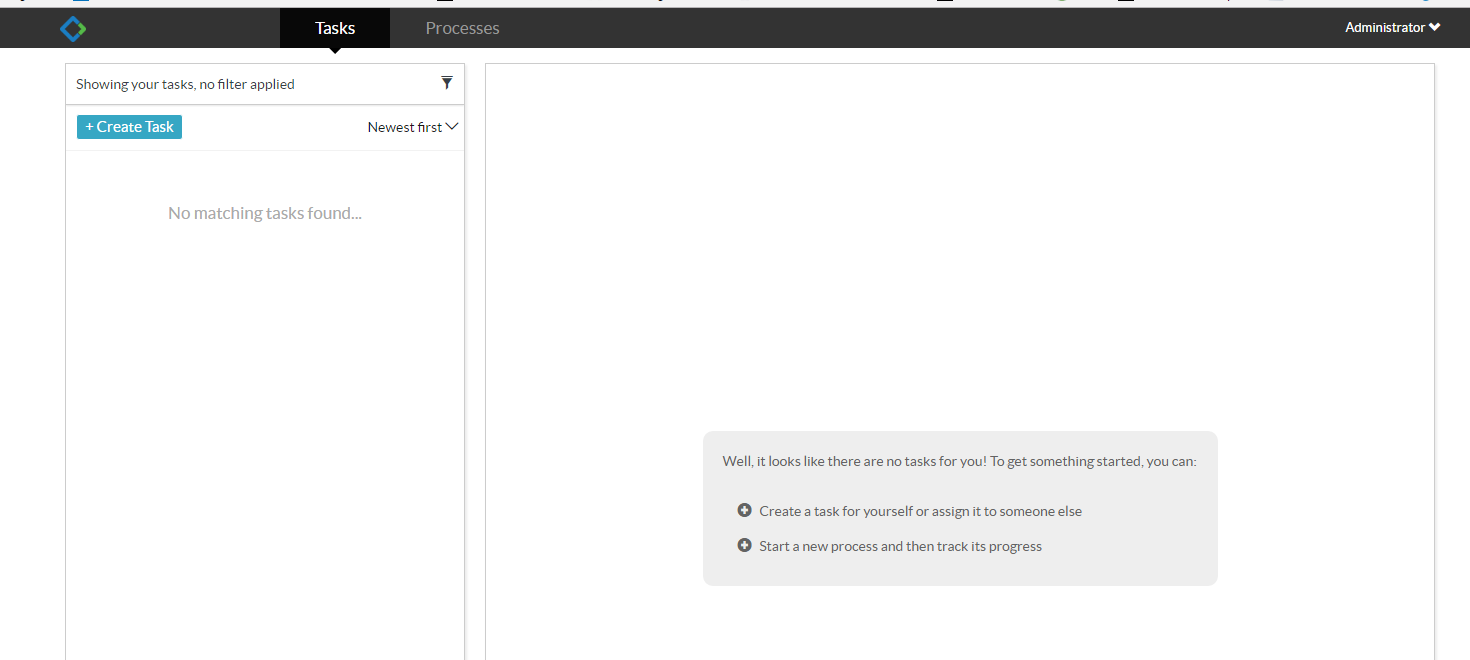
发布后，点击应用首页，可以看到刚刚自己发布的流程

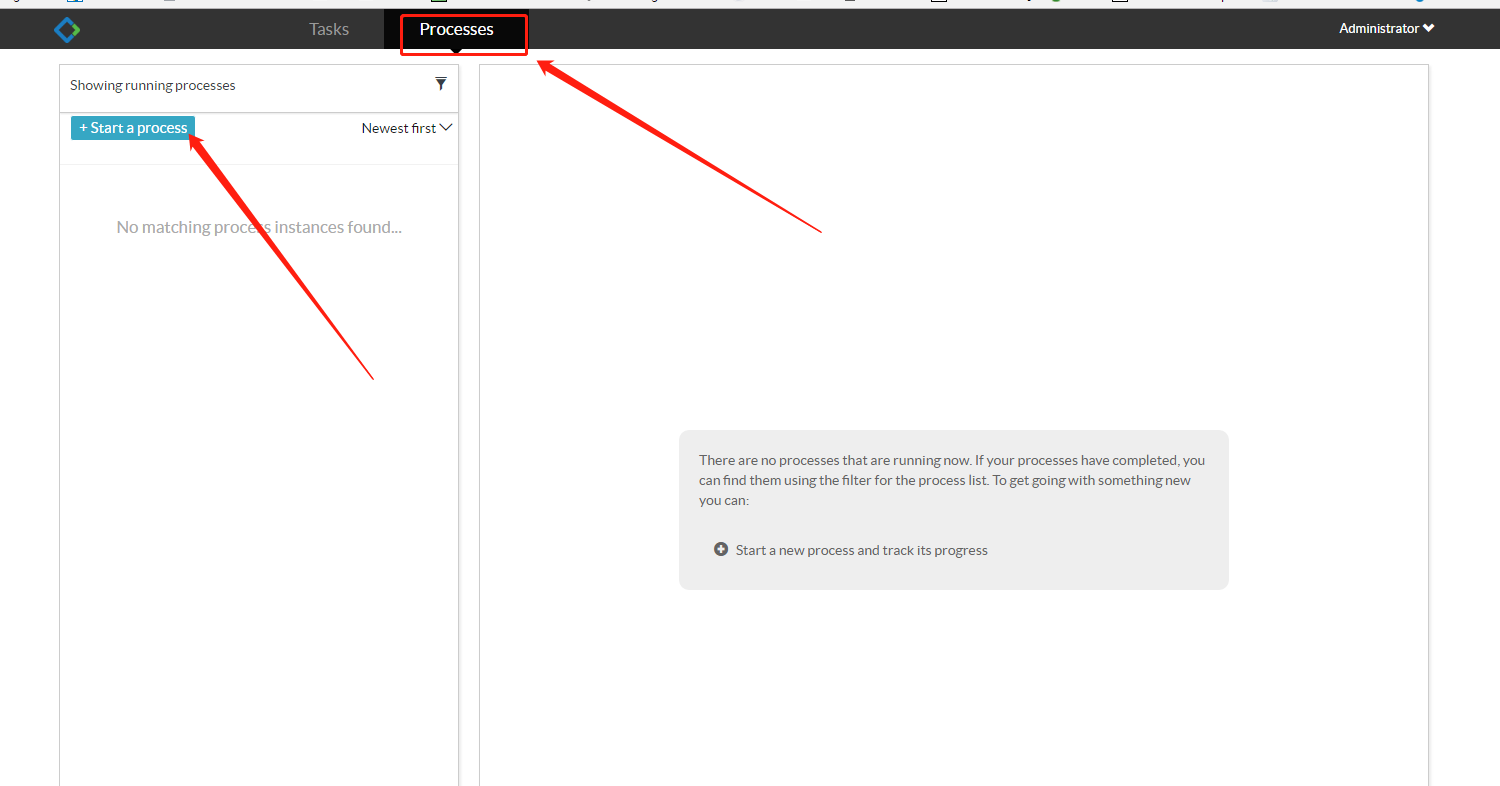




接着要启动该流程了

点击上图中自己刚发布的app，可以看到如下图所示页面

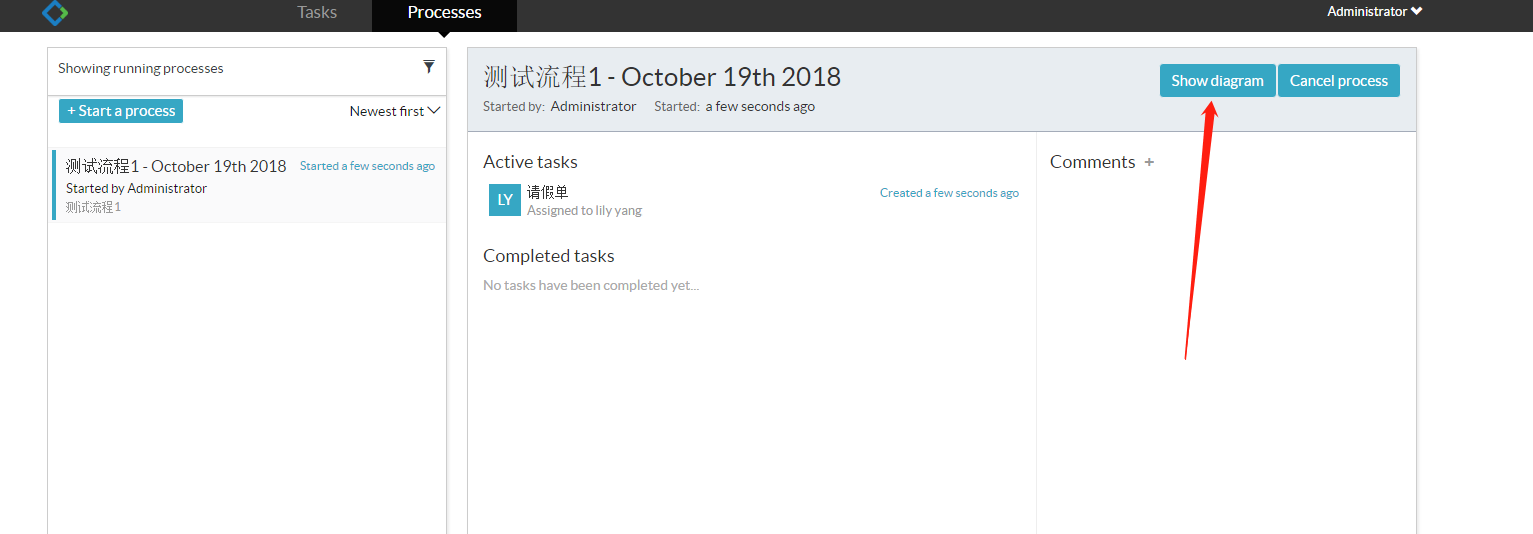




然后可以看到自己发布的流程了，点击start process即可，自定义的流程就发布了



可以看到现在正在进行的任务如下：

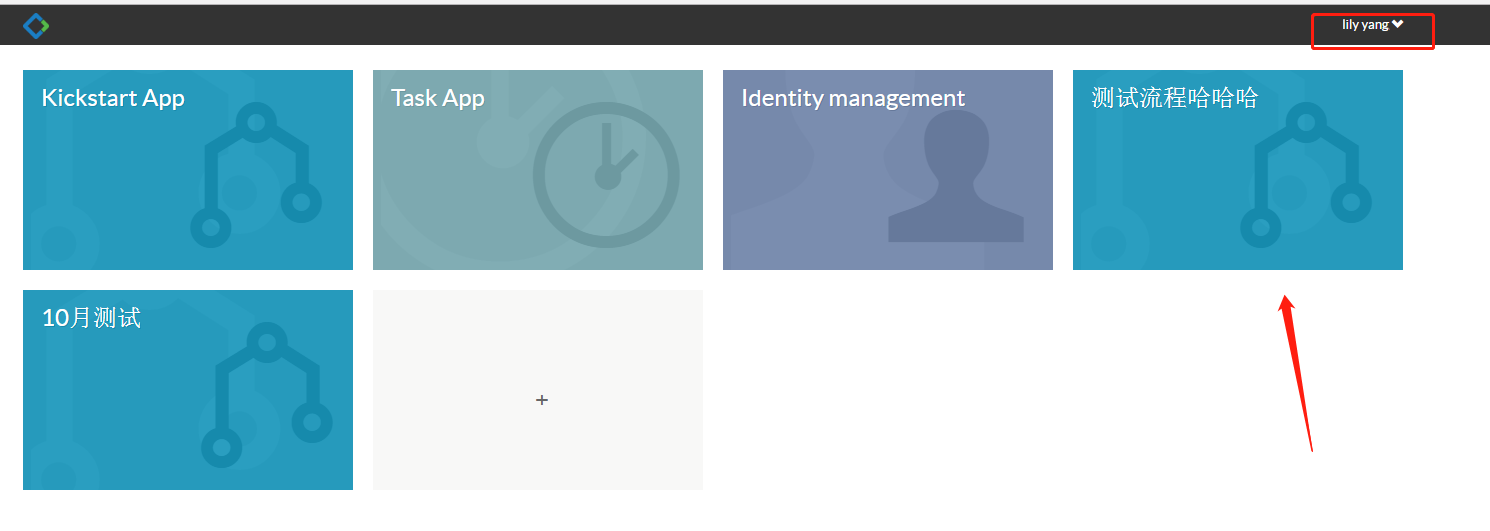


也可以查看流程图



此时自定义的流程已经启动完毕了，接着需要分别用lily和boss两个用户登录系统，进行相应的任务处理

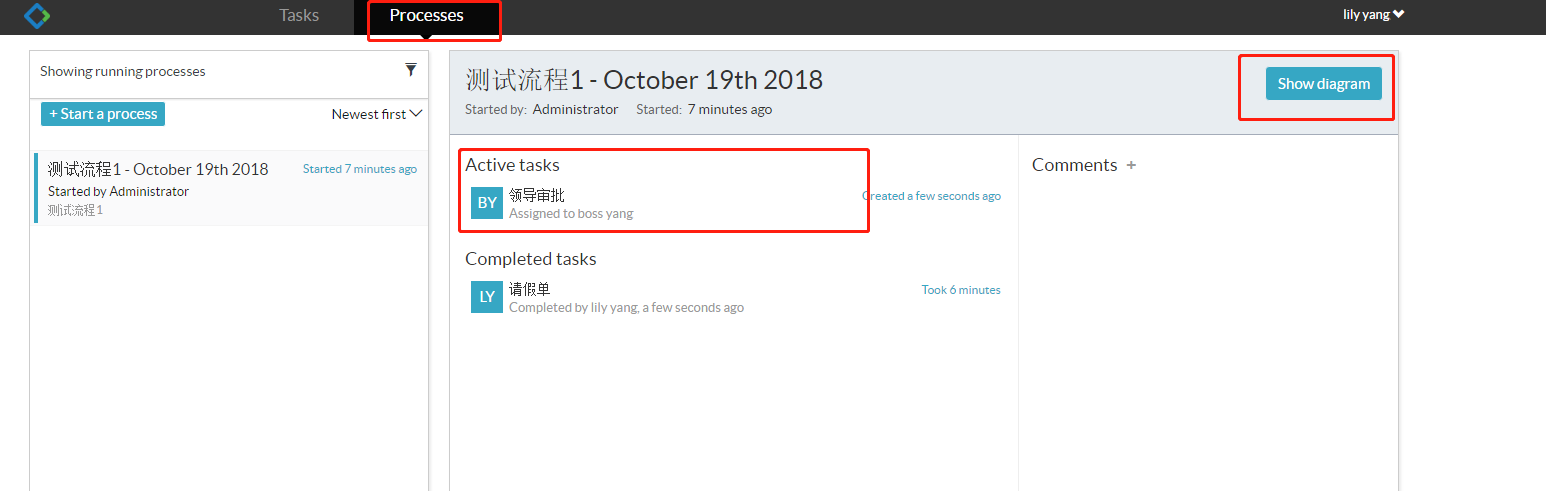
首先用lily用户登录，然后点击刚启动的流程



可以看到目前当前用户的任务

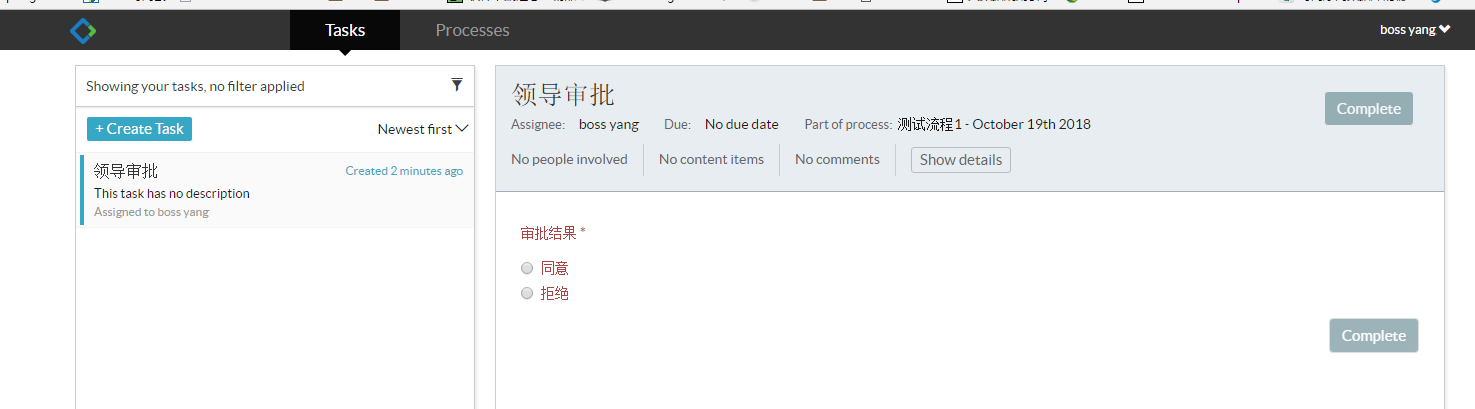


填写完成后，点击complete，可以看到流程走到boss用户了





用boss用户登录，可以看到如下界面



当boss用户点击审批结果和complete按钮后，此任务即结束了，一个完整的流程走完了