# Activiti6工作流集成平台操作手册

**开发人员：樊江锋**

目录

[Activiti6工作流集成平台操作手册 1](#_Toc41137174)

[一， 平台使用说明 4](#_Toc41137175)

[1，创建新模型 4](#_Toc41137176)

[2，已部署的流程定义 6](#_Toc41137177)

[3，流程实例管理 6](#_Toc41137178)

[4，流程实例状态图 7](#_Toc41137179)

[二， API调用接口说明 7](#_Toc41137180)

[1， 部署流程实例接口 7](#_Toc41137181)

[2， 查询组任务接口 9](#_Toc41137182)

[3， 拾取组任务接口 10](#_Toc41137183)

[4， 归还组任务接口 10](#_Toc41137184)

[5， 查询待办接口 11](#_Toc41137185)

[6， 处理待办->到下一节点（通过） 12](#_Toc41137186)

[7，处理待办->到上一节点（驳回） 13](#_Toc41137187)

[8，查询流程实例接口【包括业务编号】【重要】 14](#_Toc41137188)

[三， 平台使用注意事项 14](#_Toc41137189)

[1， 构建模型1 14](#_Toc41137190)

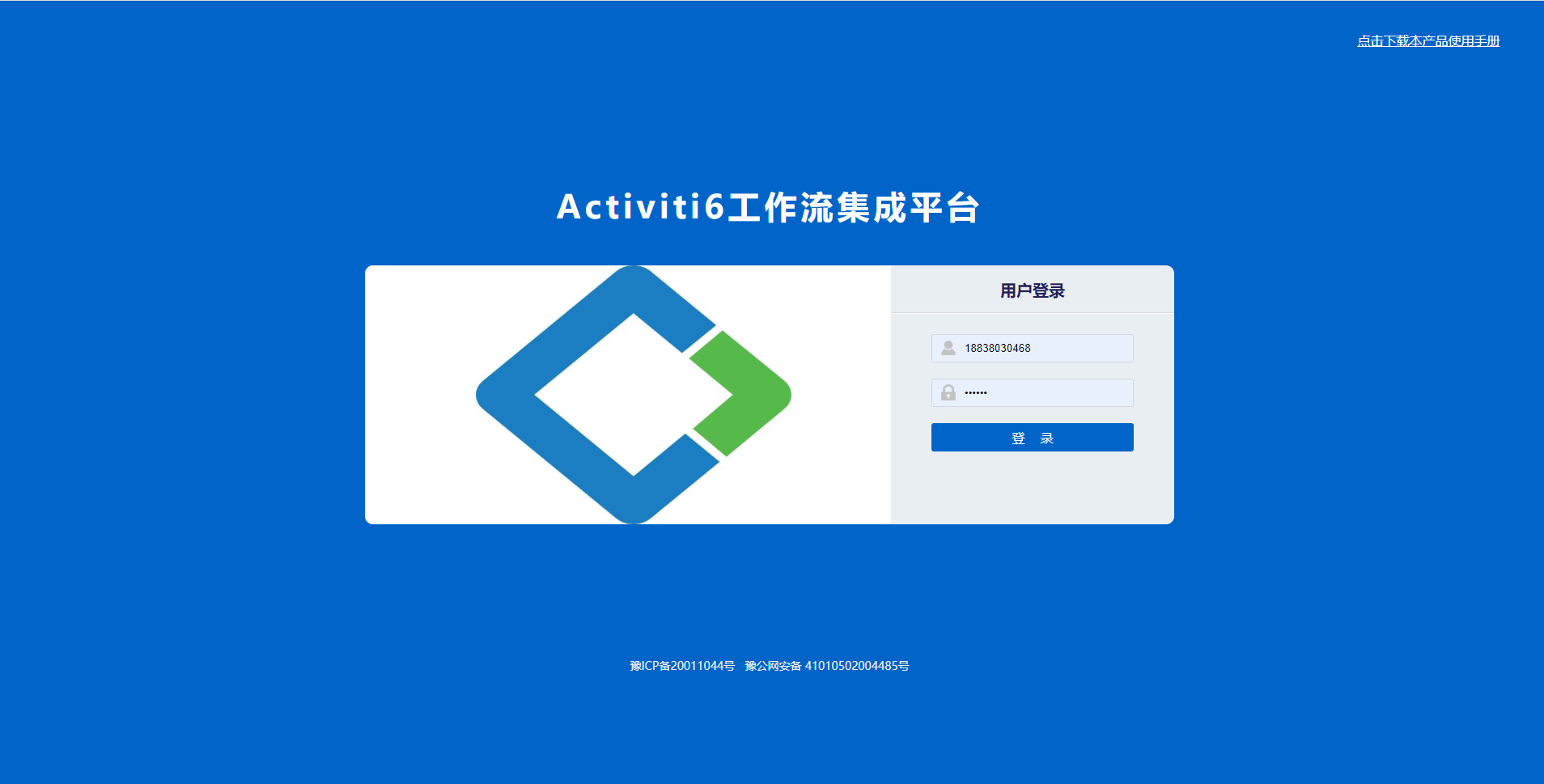
[2， 构建模型2 15](#_Toc41137191)

**文档修订记录表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **作者** | **操作** | **日期** | **说明** |
| V1.0 | 樊江锋 | 创建 | 2020-5-10 | 增加平台使用说明模块 |
| V1.0 | 樊江锋 | 创建 | 2020-5-23 | 增加API调用接口说明 |

# 平台使用说明

进入登录页面，输入账号密码进行登录，右上角可以下载操作手册。

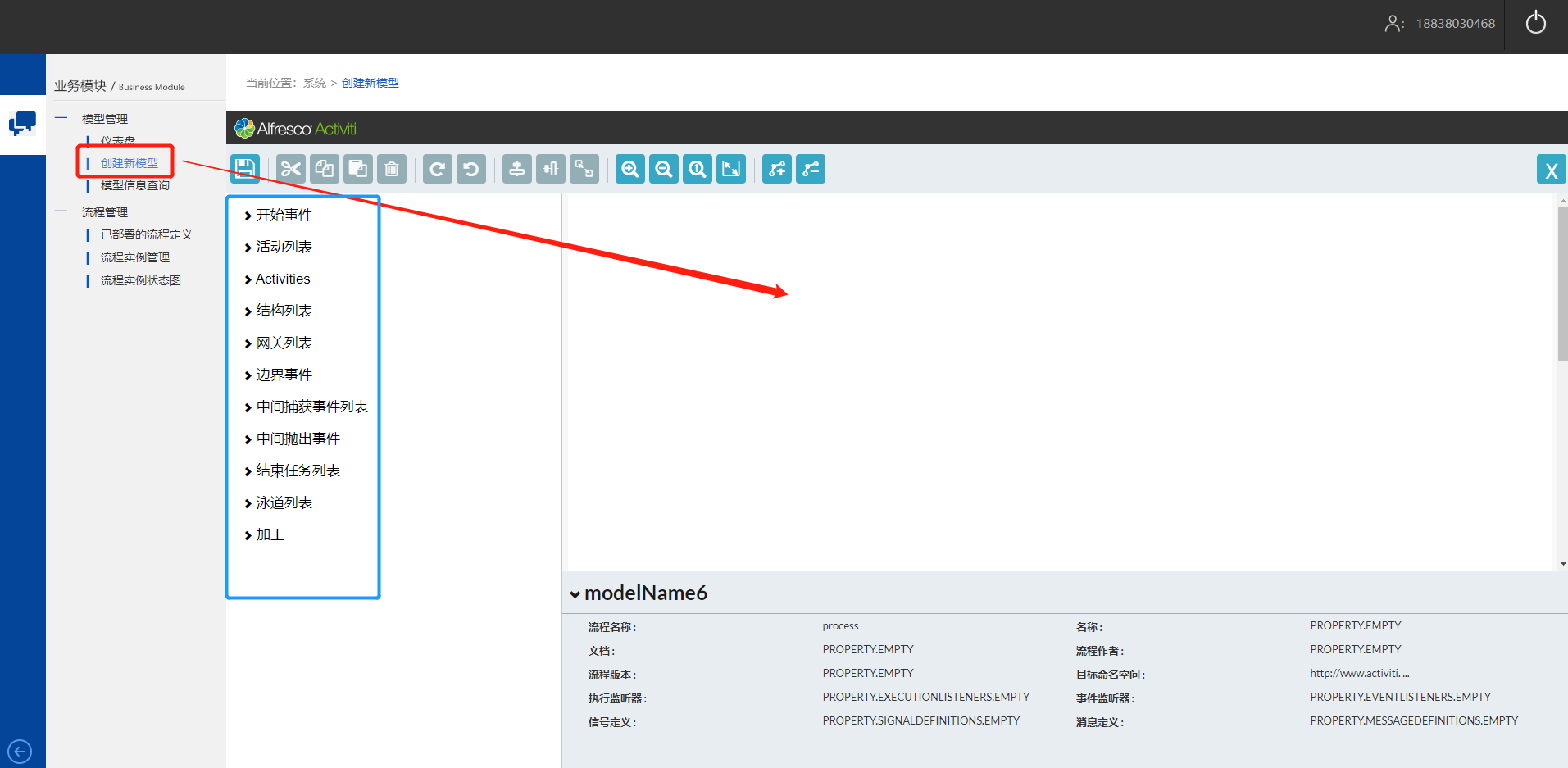


进入首页

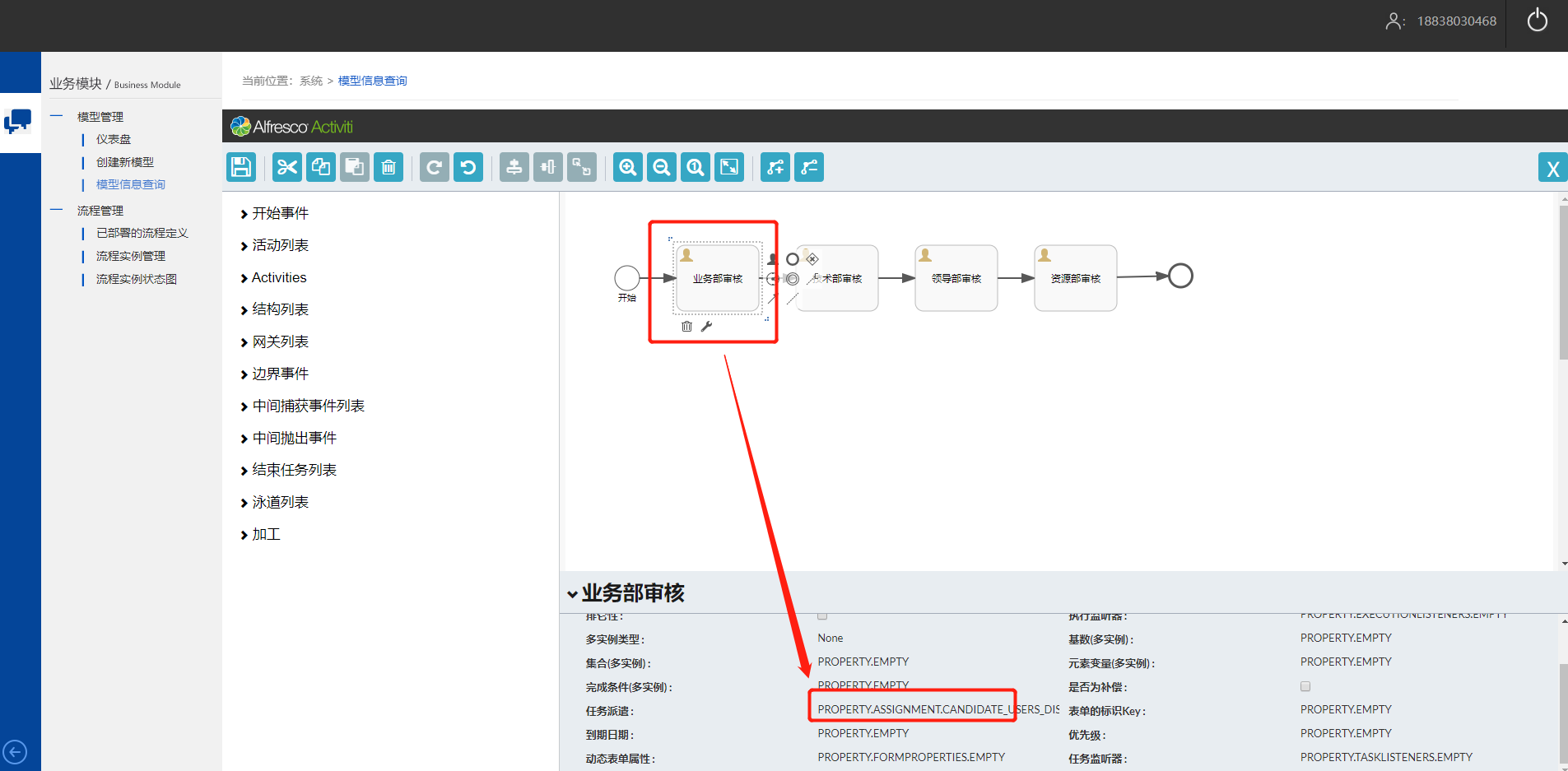


## 1，创建新模型

点击创建新模型模块，可以构建一个新的流程图模型。构建成功之后可以部署为流程定义。



画了流程图模型之后，点击用户任务节点，进行任务派遣。



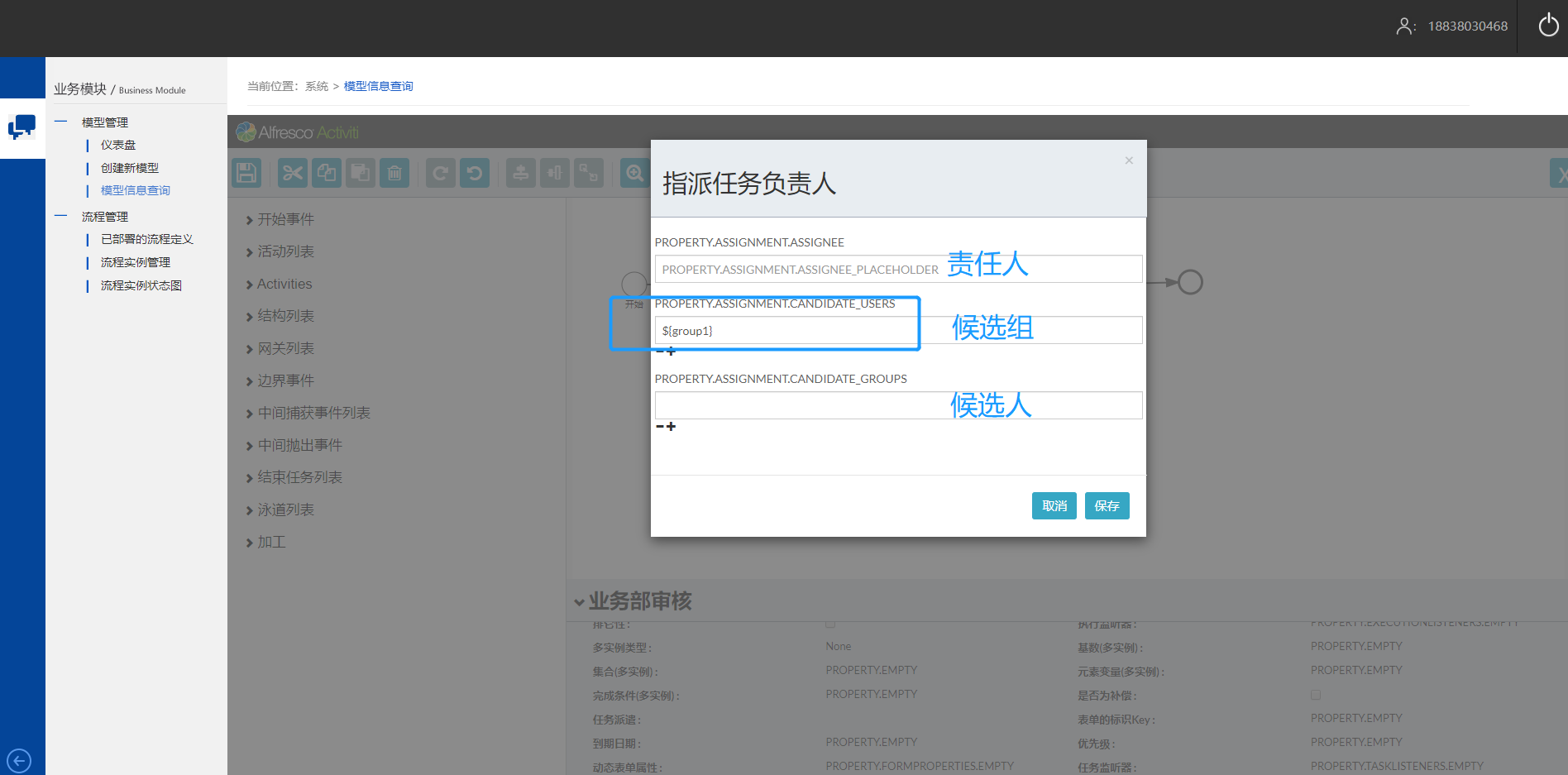
第一个是责任人，是给用户节点指定一个唯一的负责人，缺点是只能指定的某个人可以办理审核工作，写的太死！

第二个是候选组，第三个是候选人！用的最多的就是使用候选组！

三选一进行填写，只能填写一个！

写法举例：${ group }

用户只需要封装一个Map<String , Object> 类型的变量 group 即可传给平台的API即可，这个group 装载一个部门下的所有员工，String为group变量，Object为用户的登录用户名，这样，Activiti工作流系统就可以识别出你指定的部门下的所有员工！



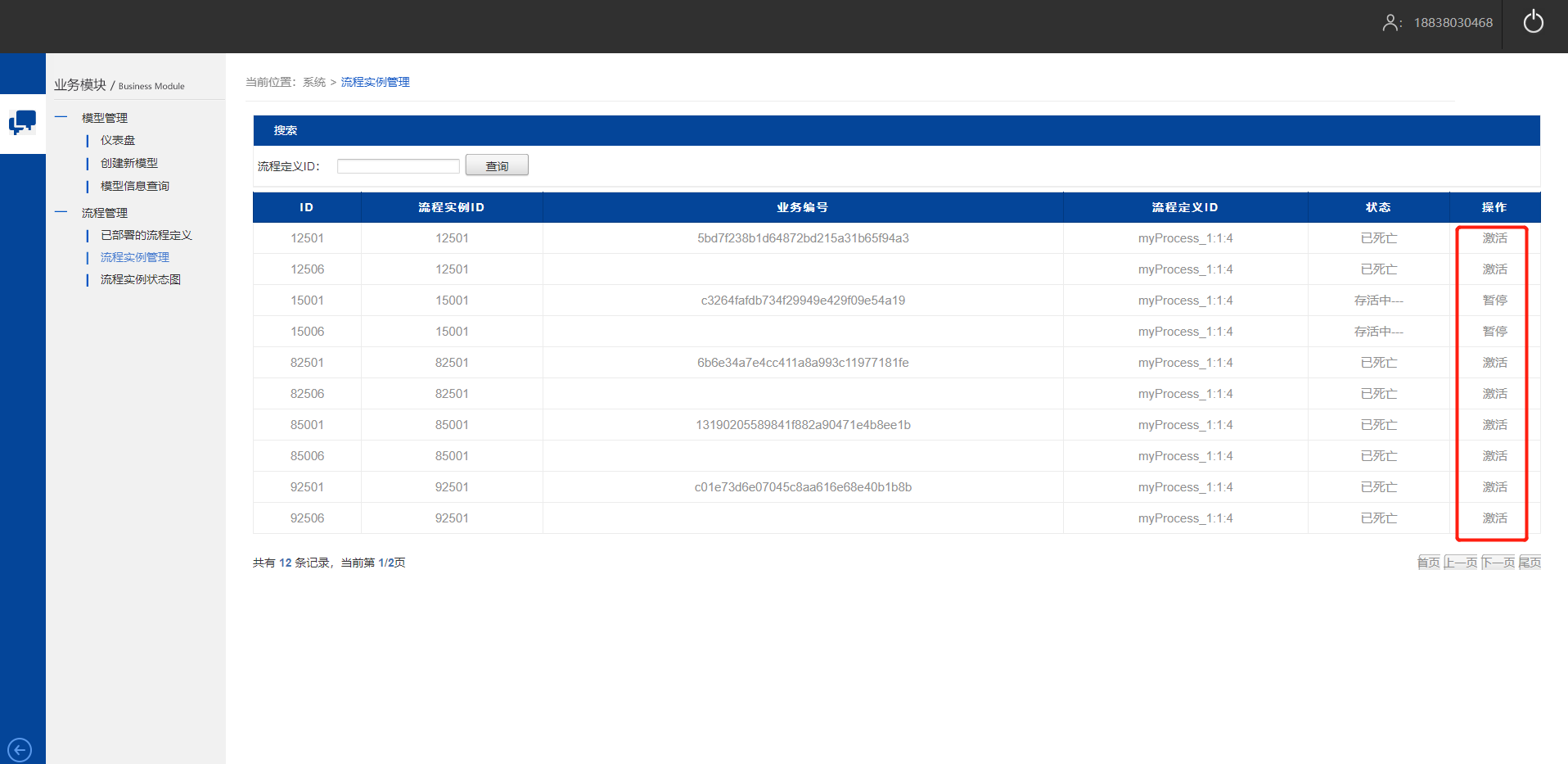
## 2，已部署的流程定义

点击已部署的流程定义模块，可以对流程定义进行激活，暂停，查看下面的示例和删除操作。



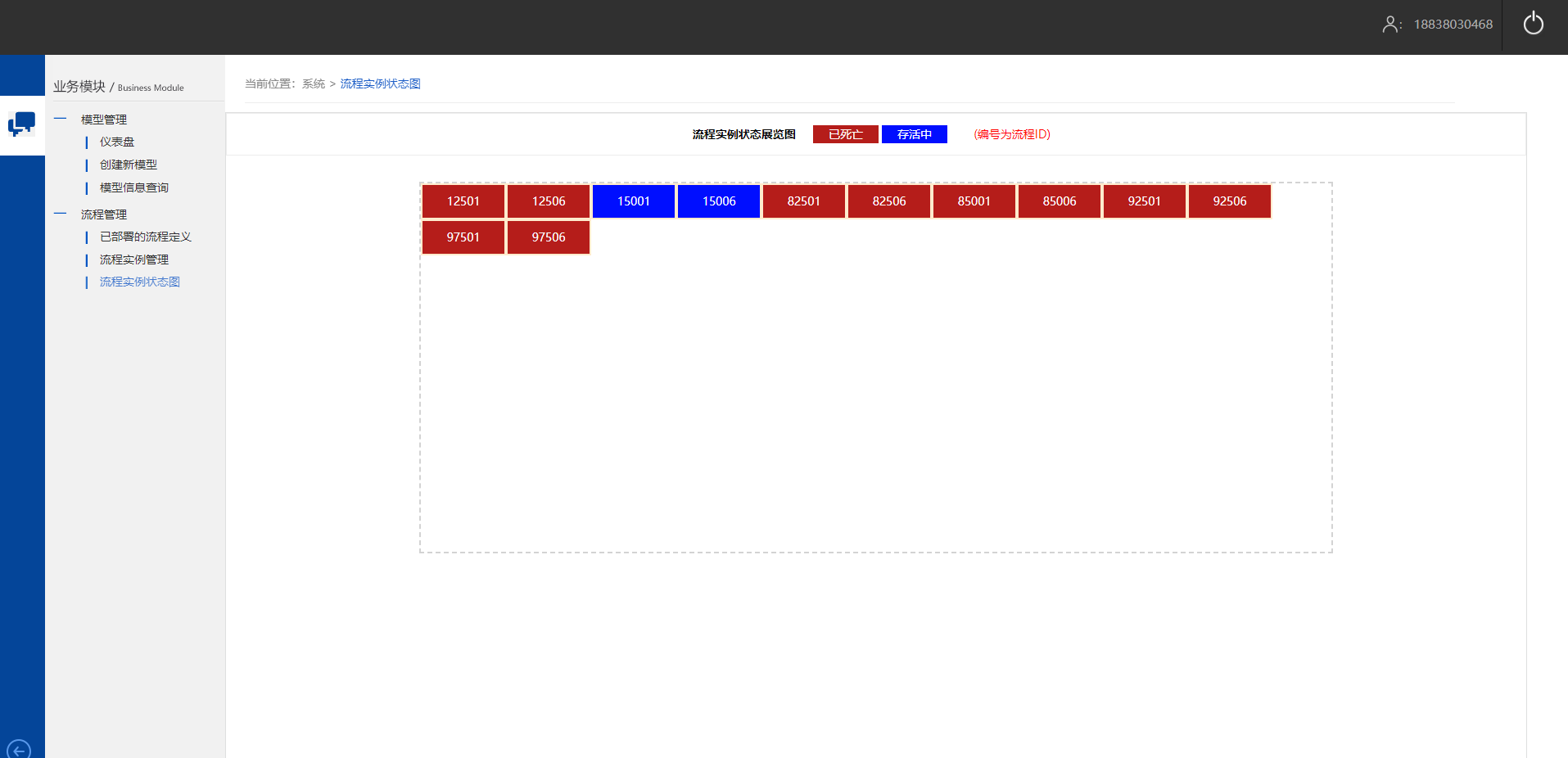
## 3，流程实例管理

点击流程实例管理，可以对流程实例进行暂停和激活的操作。



## 4，流程实例状态图

点击流程实例状态图，可以展览全部流程实例的允许状态（存活中或已死亡）



# API调用接口说明

## 部署流程实例接口

该接口通常在一个流程发起时调用，例如请假单审批流程，提交请假单的同时调用本接口部署流程实例。

请求方式：GET请求

请求参数说明：

|  |
| --- |
| processDefinitionKey ：流程定义KEY  businessKey ： 业务编号  Map<String,Object> variables ：任务负责人 |

响应参数；（json格式字符串）

|  |
| --- |
| {  "message": "success",  "status": "200"  } |

任务负责人参数如何封装？

讲一个部门下的所有员工的登录账号已逗号分割作为value存储到variables的map键值对中，键为画流程图指定任务候选人时的变量名。

代码举例

|  |
| --- |
| Map<String,Object> variables =**new** HashMap<String, Object>();  *// variables.put("group1", "张三，李四，王五");${group1}组的成员  //业务部成员账号列表* List<Map> ywbList = **mapper**.selectUserByDeptId(1);  String ywbTempStr=**""**;  **for**(**int** t=0;t<ywbList.size();t++){  **if**(t==ywbList.size()-1){  ywbTempStr+=ywbList.get(t).get(**"account"**);  }**else**{  ywbTempStr+=ywbList.get(t).get(**"account"**)+**","**;  }  } |

restTemplate请求方法demo

|  |
| --- |
| **public** JSONObject deployProcessInstance(String processDefinitionKey, String businessKey, Map<String,Object> variables){  String json = JSONObject.*toJSONString*(variables);  RestTemplate restTemplate = **new** RestTemplate();  String url = **"http://tiger2.cn:8085/node/deployProcessExecution?processDefinitionKey="**+processDefinitionKey+**"&businessKey="**+businessKey+**"&groupValue={json}"**;  HttpHeaders headers = **new** HttpHeaders();  MediaType type = MediaType.*parseMediaType*(**"application/json; charset=UTF-8"**);  headers.setContentType(type);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class**,json);  String responseData = response.getBody();  JSONObject jsonObject= JSON.*parseObject*(responseData);  **return** jsonObject; } |

## 查询组任务接口

请求方式：GET请求

请求参数说明：

|  |
| --- |
| processName ：流程定义KEY  assignment ： 任务负责人 |

restTemplate方法demo

|  |
| --- |
| **public** JSONObject getGroupList(String processName,String assignment){  RestTemplate restTemplate=**new** RestTemplate();  String url =***urls*** +**"/groupList?processDefinitionId="**+processName+**"&assignmentName="**+assignment;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject= JSON.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {  "data": "[{  "suspensionState":1, 是否挂起  "id":"50018", 任务ID  "processInstanceId":"50010", 流程实例ID  "processDefinitionId":"process:3:12511", 流程定义ID  "priority":50,  "revision":1,"  "executionId":"50015",  "taskDefinitionKey":"\_2",  "createTime":1590212347645,  "name":"业务部审核", 节点的任务状态  "tenantId":""  }]",  "message": "success", 请求结果状态  "status": "200" 状态值  } |
|  |

## 拾取组任务接口

请求方式：GET请求

请求参数说明：

|  |
| --- |
| taskId ：任务ID  assignmentName ：任务负责人 |

restTemplate请求方法demo

|  |
| --- |
| **public** JSONObject getGroupTask(String taskId,String assignmentName){  RestTemplate restTemplate=**new** RestTemplate();  String url =***urls*** +**"/getTask?taskId="**+taskId+**"&assignmentName="**+assignmentName;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject= JSON.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {  "message": "组任务拾取成功",  "status": "200"  } |

## 归还组任务接口

请求方式：GET请求

请求参数说明：

|  |
| --- |
| taskId ：任务ID  assignmentName ：任务负责人 |

restTemplate请求方法demo

|  |
| --- |
| **public** JSONObject returnGroupTask(String taskId,String assignmentName){  RestTemplate restTemplate=**new** RestTemplate();  String url =***urls*** +**"/returnTask?taskId="**+taskId+**"&assignmentName="**+assignmentName;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject= JSON.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {  "message": "组任务归还成功",  "status": "200"  } |

## 查询待办接口

请求方式：GET请求

请求参数说明：

|  |
| --- |
| processDefinitionId：流程定义ID  assignmentName：任务负责人 |

restTemplate请求方法demo

|  |
| --- |
| **public** JSONObject getDbList(String processDefinitionId,String assignmentName){  RestTemplate restTemplate=**new** RestTemplate();  String url = ***urls***+**"/dbList?processDefinitionId="**+processDefinitionId+**"&assignmentName="**+assignmentName;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject = JSONObject.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {"data":"[{  "suspensionState":1, 是否被挂起  "id":"50018", 任务ID  "processInstanceId":"50010", 流程实例ID  "processDefinitionId":"process:3:12511", 流程定义ID  "priority":50,  "claimTime":1590213306649,  "revision":4,  "executionId":"50015",  "taskDefinitionKey":"\_2",  "createTime":1590212347645,  "name":"业务部审核", 节点任务状态  "tenantId":"",  "assignee":"18838030468" 任务负责人的用户账号  }]",  "message":"success", 请求状态  "status":"200" 状态码  } |

## 处理待办->到下一节点（通过）

请求方式：GET请求

请求参数说明：

|  |
| --- |
| taskId ：任务ID  assignmentName ：任务负责人 |

restTemplate请求方法demo

|  |
| --- |
| **public** JSONObject soluteTask(String taskId,String assignmentName){  RestTemplate restTemplate=**new** RestTemplate();  String url = ***urls***+**"/soluteToNext?taskId="**+taskId+**"&assignmentName="**+assignmentName;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject = JSONObject.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {  "message": "待办已处理，已到下一节点",  "status": "200"  } |

## 7，处理待办->到上一节点（驳回）

请求方式：GET请求

请求参数说明：

|  |
| --- |
| taskId ：任务ID  assignmentName ：任务负责人 |

RestTemplate请求方法demo

|  |
| --- |
| **public** JSONObject soluteTaskToPrev(String taskId,String assignmentName){  RestTemplate restTemplate=**new** RestTemplate();  String url = ***urls***+**"/soluteToPrev?taskId="**+taskId+**"&assignmentName="**+assignmentName;  HttpHeaders headers = **new** HttpHeaders();  headers.setContentType(MediaType.***APPLICATION\_FORM\_URLENCODED***);  ResponseEntity<String> response = restTemplate.getForEntity( url, String.**class** );  String responseData = response.getBody();  JSONObject jsonObject = JSONObject.*parseObject*(responseData);  **return** jsonObject; } |

响应参数：（json格式字符串）

|  |
| --- |
| {  "message": "待办已处理，已回退到上一节点",  "status": "200"  } |

## 8，查询流程实例接口【包括业务编号】【重要】

请求方式：GET请求

请求参数说明：

|  |
| --- |
| processInstanceId：流程实例ID |

响应参数：（json格式字符串）

|  |
| --- |
| {  "data":  "{  "bUSINESS\_KEY\_":"5caca32cc32646a1a715e7ed8ec50019", 业务编号  "iD\_":"50010",  "iS\_ACTIVE\_":"1",  "pROC\_DEF\_ID\_":"process:3:12511",  "pROC\_INST\_ID\_":"50010",  "rEV\_":"1",  "sTART\_TIME\_":1590212347644"  }",  "message":"success",  "status":"200"  } |

# 平台使用注意事项

## 构建模型1

构建流程模型时，设置用户任务指定责任人时，如下图，点击用户任务，然后修改流程的ID值（节点）。

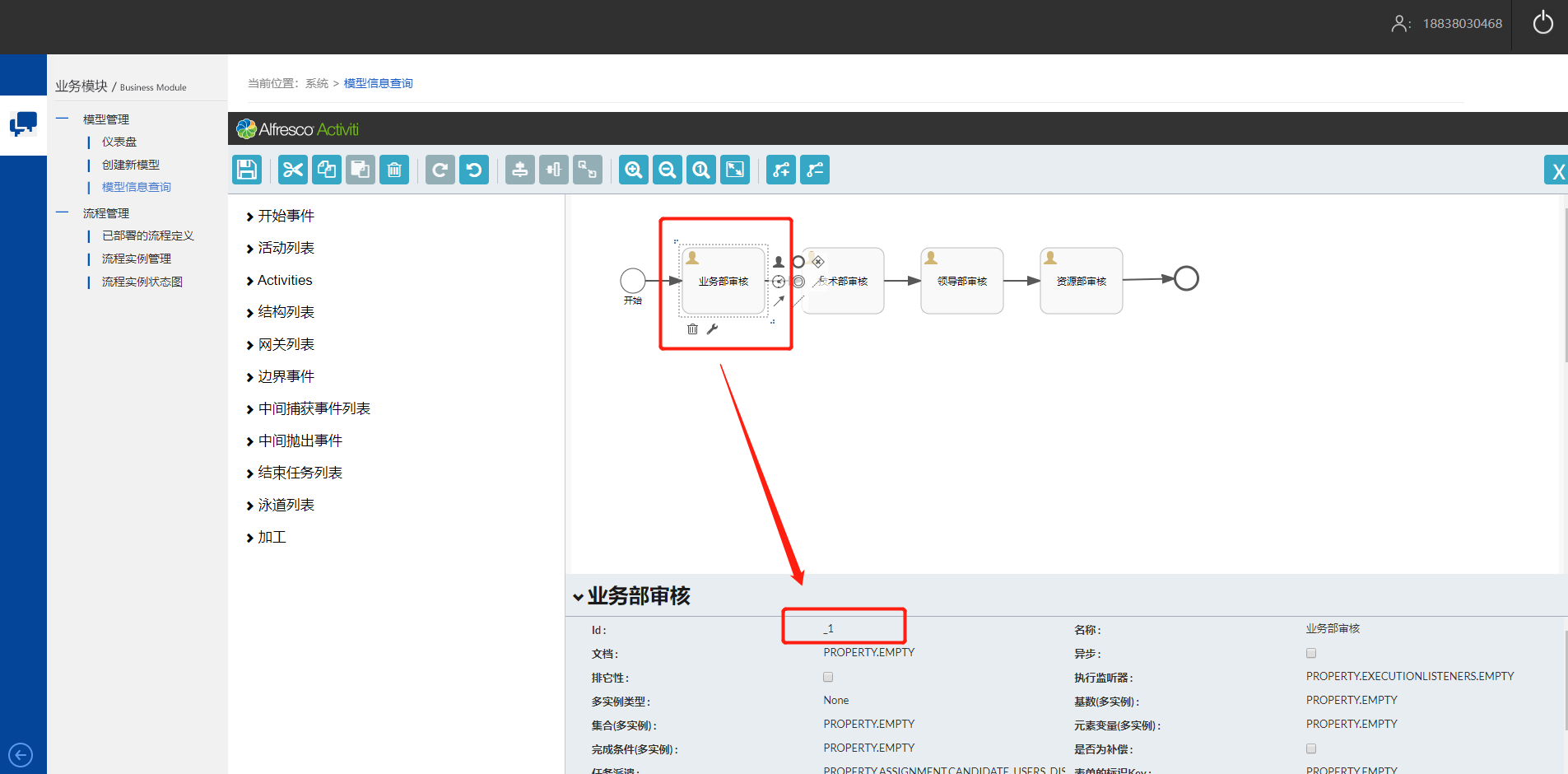
示例：【 下划线加数字，依次加1 】

第一个用户任务节点：\_4

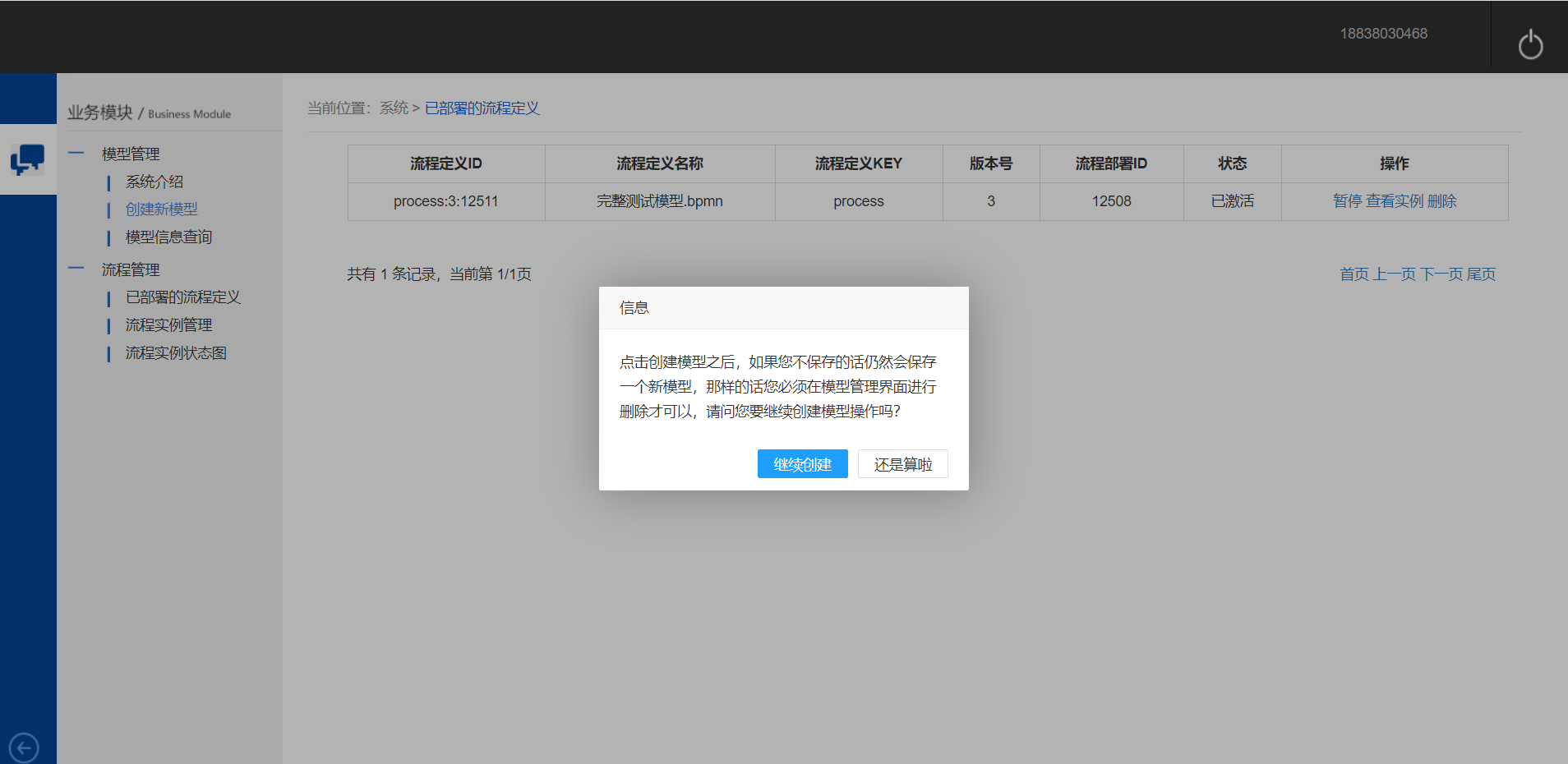
第一个用户任务节点：\_5

……… 以此类推

都要设置，否则流程会作废！



## 构建模型2



点击构建模型选项后，选择继续构建的话，那么平台会自动默认创建一个流程图（模型），尽管您没有保存，那样您还需要在模型管理界面进行删除！只有保存了之后才是您想要的流程图噢！

默认创建的模型，名称统一为：modelName6