



RCloud BPMS SDK

用户开发指南

中软国际新业务集团

2014 年

第一章 概述.....	3
1.1 概述.....	3
1.2 开发准备	4
1.3 SDK 统一调用方法	4
1.4 SDK 调用抛出异常说明	5
1.5 Rbpms-sdk API 接口说明.....	5
第二章 开发指导.....	5
2.1 开发流程定义	5
2.1.1 创建未发布流程定义.....	5
2.1.2 发布流程定义.....	6
2.1.3 修改流程定义（已发布、未发布）	6
2.1.4 删除未发布流程定义.....	7
2.1.5 其他辅助接口.....	7
2.2 管理流程实例	8
2.2.1 查询流程实例	8
2.2.2 启动流程实例	9
2.2.3 挂起和恢复流程实例	9
2.2.4 结束流程实例.....	10
2.2.5 删除流程实例	10
2.2.6 查询流程历史轨迹.....	11
2.3 任务管理.....	11
2.3.1 Task 查询	11
2.3.2 Task 提交	11
2.3.3 Task 跳转	12
2.3.4 Task 补发	12
2.3.5 Task 转发	12
2.3.6 Task 转办	13

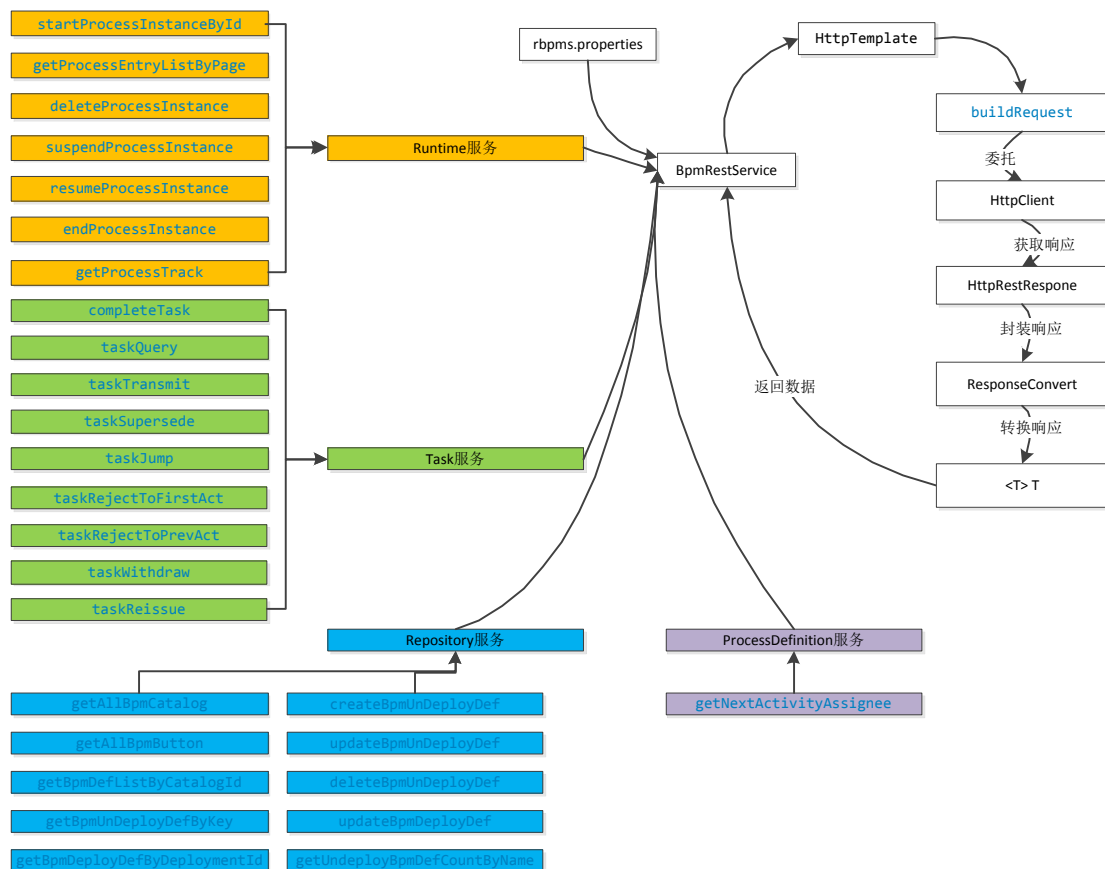
2.3.7	退回拟稿人.....	13
2.3.8	退回上一节点.....	14
2.3.9	收回 Task	14
2.4	获取处理人信息.....	14
2.4.1	获取处理人信息.....	14

第一章 概述

1.1 概述

本开发指南详细描述了 RBPMS REST 服务 JAVA 开发工具包(Rcloud BPMS REST JAVA SDK 以下简称 rbpms-sdk) 的使用方法，主要面向基于该产品的二次开发人员。

rbpms-sdk 是应用系统和 Rcloud BPMS 提供的 REST 服务交互的主要途径。通过 rbpms-sdk，用户应用程序可以访问和操作工作流的执行，进而控制流程运转并在适当时候完成对应的业务操作。rbpms-sdk 调用结构如下图所示：



RBPMs REST 服务 JAVA SDK 提供了以下四大类 API，分别是运行时（Runtime）服务、任务（Task）服务、部署（Repository）服务、流程定义（ProcessDefinition）服务。

1.2 开发准备

在利用 Rbpms-sdk API 开始进行 workflow 应用开发之前，需要进行必要的准备工作：

- 1、在工程中引入 rbpms-sdk 的 jar 包。Jar 包可以在 Rcloud 网站下载。
- 2、在应用的 classes 的跟目录建立 rbpms-sdk 的配置文件 rbpms.properties，文件内容如下：

#rbpms 服务路径

rbpm.url=http://www.rcloud.rbpms.com/:8080/rbpms

1.3 SDK 统一调用方法

RBPMs REST 服务 JAVA SDK 统一调用方法如下(以启动流程为例)：

```
//线程安全类
BpmRestService bpmRestService=BpmRestService.getInstance();
//添加启动参数
ProcessInstanceStart start=new ProcessInstanceStart();

start.setProcessDefinitionId("20140402100150-1");
start.setStartUserId("8afac7de454ffc5f01454fff82450001");
start.setStartDeptId("d_8afac7de454ffc5f01454ffe15e70003");
start.setFormCode("baidu");
Map<String, Object> variables=new HashMap<String, Object>();
variables.put("id", "formid");
start.setVariables(variables);
//获取对应服务，启动流程
String userId = "userId";
long id= bpmRestService.getRuntimeService(userId)
        .startProcessInstanceId(start, ak, sk);
```

要通过 Rbpms-sdk 操作 RBPMs REST 服务接口接口，都需要通过类 BpmRestService。类 BpmRestService 的实例通过 BpmRestService.getInstance()方法来获得。要操作哪个功能模块就调用 BpmRestService 的相应 get 方法来获得功能模块的具体接口实现。比如 RuntimeService = bpmRestService.getRuntimeService(userId)。userId 原本定义为 bpm 应用中的此操

作涉及到的用户 id，涉及不到用户时传空字符串即可。目前此 `userId` 已无实际意义，全部传空字符串即可。

注意：每个功能接口都包含两个共同的参数 `ak`（access key）和 `sk`（security key），此两个参数为从单点登录系统获得。

1.4 SDK 调用抛出异常说明

RBPMs REST 服务 JAVA SDK 调用过程会抛出以下异常，由调用方根据异常作相应处理。

第二章 BpmException

所有引起操作失败的异常：包括 `restful` 请求异常，`json` 转换等。

1.5 Rbpms-sdk API 接口说明

详见 `Rbpms-sdk` API 文档。

第二章 开发指导

2.1 开发流程定义

流程是 RBPMs 服务的核心，首先讲解流程定义的相关操作，如：创建、修改、发布、删除等。流程定义相关的接口都在 `RepositoryService` 类中。

2.1.1 创建未发布流程定义

开发 RBPMs 应用程序，首先要创建新的工作流程。工作流程由流程定义来描述。新创建的流程定义默认都是未发布的，需要调用接口是

```
createBpmUnDeployDef(
    ProcessDefVo pd,
```

```
java.lang.String ak,
java.lang.String sk)。
```

此接口包含参数 ProcessDefVo 和公共参数 ak、sk。其中 ProcessDefVo 是对要创建的流程定义的描述信息。ProcessDefVo 的具体内容参考 Rbpms-sdk API 文档。

2.1.2 发布流程定义

发布流程定义目前需要到 RBPMS 的管理控制点操作，sdk 中未提供操作接口。

2.1.3 修改流程定义（已发布、未发布）

修改未发布的流程定义需要调用的接口是

```
updateBpmUnDeployDef(
ProcessDefVo pd,
java.lang.String ak,
java.lang.String sk)。
```

此接口包含参数 ProcessDefVo 和公共参数 ak、sk。其中 ProcessDefVo 是对要创建的流程定义的描述信息。ProcessDefVo 的具体内容参考 Rbpms-sdk API 文档。

修改已发布的流程定义需要调用的接口是

```
updateBpmDeployDef (
ProcessDefVo pd,
java.lang.String ak,
java.lang.String sk)。
```

此接口包含参数 ProcessDefVo 和公共参数 ak、sk。其中 ProcessDefVo 是对要创建的流程定义的描述信息。ProcessDefVo 的具体内容参考 Rbpms-sdk API 文

档。

2.1.4 删除未发布流程定义

删除未发布的流程定义需要调用的接口是

deleteBpmUnDeployDef (

java.lang.String dbid,

java.lang.String ak,

java.lang.String sk)。

此接口包含参数 dbid 和公共参数 ak、sk。其中 dbid 是要删除的流程定义 id。具体内容参考 Rbpms-sdk API 文档。

2.1.5 其他辅助接口

除了以上几个接口完成流程定义的注意功能，还有几个辅助作用的接口对帮助开发者开发 RBPMS 应用程序。

获取所有按钮接口

getAllBpmButton(java.lang.String ak, java.lang.String sk),

接口返回所有流程按钮集合。

获取所有目录接口

getAllBpmCatalog(java.lang.String ak, java.lang.String sk),

接口返回当前租户已创建的所有目录

根据流程目录 id，获取流程目录下的流程定义信息

getBpmDefListByCatalogId(

java.lang.String catalogId,

java.lang.String ak,


```
java.lang.String sk),
```

接口返回参数 catalogId 目录下的流程定义集合。

通过部署 ID 获取流程定义信息，包括流程 xml 和 png

getBpmDeployDefByDeploymentId(

```
java.lang.String deploymentId,
```

```
java.lang.String ak,
```

```
java.lang.String sk),
```

接口返回参数 deploymentId 的流程定义

通过流程 key 获取未发布流程定义信息，包括流程 xml 和 png

getBpmUnDeployDefByKey(

```
java.lang.String processDefKey,
```

```
java.lang.String ak,
```

```
java.lang.String sk),
```

接口返回参数 processDefKey 的流程定义

通过流程名字查询流程个数

getUndeployBpmDefCountByName(

```
java.lang.String flowName,
```

```
java.lang.String ak,
```

```
java.lang.String sk),
```

接口返回流程个数。

2.2 管理流程实例

流程实例的相关操作都在 `RuntimeService` 类中。如查询、启动、挂起、恢复、删除等。

2.2.1 查询流程实例

查询流程实例需要调用的接口是

getProcessEntryListByPage(

ProcessInstanceQuery query,

java.lang.String ak,

java.lang.String sk).

ProcessInstanceQuery 中封装了查询可以用到的参数（具体内容参考 Rbpms-sdk API 文档）。接口返回结果对象 ProcessInstanceQueryResult，结果中包括查询到的符合条件的实例总数和当期页面的实例集合。

2.2.2 启动流程实例

启动流程实例需要调用的接口是

startProcessInstanceById(

ProcessInstanceStart processInstanceStart,

java.lang.String ak,

java.lang.String sk),

参数类 ProcessInstanceStart 封装了启动实例必要的参数（具体内容参考 Rbpms-sdk API 文档）。返回值为该流程实例的 id。

2.2.3 挂起和恢复流程实例

处于流转中的流程可以暂时挂起和恢复。

挂起流程实例需要调用的接口为：

suspendProcessInstance(

java.lang.String processInstanceId,

java.lang.String ak,

java.lang.String sk)

参数 processInstanceId 为流程实例 id。

恢复已挂起了流程实例需要调用接口

```
resumeProcessInstance(  
    java.lang.String processInstanceId,  
    java.lang.String ak,  
    java.lang.String sk),
```

参数 processInstanceId 为流程实例 id。

2.2.4 结束流程实例

结束流程实例需要调用接口

```
endProcessInstance(  
    java.lang.String executionId,  
    java.lang.String userId,  
    java.lang.String ak,  
    java.lang.String sk),
```

参数 executionId 为流程执行 id，参数 userId 为业务用的 id。

注意：已结束的流程不能挂起和恢复。

2.2.5 删除流程实例

删除流程实例需要调用的接口是

```
deleteProcessInstance(  
    java.lang.String executionId,  
    java.lang.String ak,  
    java.lang.String sk),
```

参数 executionId 为流程执行 id。

2.2.6 查询流程历史轨迹

查询流程历史轨迹需要调用的接口是

getProcessTrack(

java.lang.String processInstanceId,

java.lang.String ak,

java.lang.String sk),

参数 processInstanceId 为流程实例 id。接口返回的结果对象

HistoryActivitysQueryResult 包括总数和当前页历史信息对象集合。

2.3 任务管理

任务管理相关操作都封装在类 TaskService 中。如查询、转发、转办、补发、提交、退回等。

2.3.1 Task 查询

Task 查询接口查询用户有多少待办任务。需要调用的接口是

taskQuery(

TaskQuery query,

java.lang.String ak,

java.lang.String sk),

参数 TaskQuery 对象封装了必要的查询条件（具体内容参考 Rbpms-sdk API 文档）。接口返回对象 TaskQueryResult 包含了总数和任务对象的当前页的集合（具体内容参考 Rbpms-sdk API 文档）。

2.3.2 Task 提交

此接口用于业务用户提交自己的任务，需要调用的接口是

completeTask(

java.lang.String taskId,

TaskComplete var,

java.lang.String ak,

```
java.lang.String sk),
```

参数 `taskId` 为任务 id, 对象 `TaskComplate` 封装了提交任务所需的必要内容
(具体内容参考 `Rbpms-sdk API` 文档)

2.3.3 Task 跳转

此接口用于任务的跳转, 需要调用的接口是

```
taskJump(
```

```
TaskJump jump,
```

```
java.lang.String ak,
```

```
java.lang.String sk),
```

参数 `TaskJump` 封装了跳转所需的必要内容 (具体内容参考 `Rbpms-sdk API` 文档)。

2.3.4 Task 补发

此接口用于任务的补发, 需要调用的接口是

```
taskReissue(
```

```
TaskReissue reissue,
```

```
java.lang.String ak,
```

```
java.lang.String sk),
```

参数 `TaskReissue` 封装了补发所需的必要内容 (具体内容参考 `Rbpms-sdk API` 文档)。

注意: 流程的补发只会补发给下一节点。

2.3.5 Task 转发

此接口用于任务的转发, 需要调用的接口是

```
taskTransmit(
```

TaskTransmit transmit,

java.lang.String ak,

java.lang.String sk) ,

参数 TaskTransmit 封装了转发所需的必要内容（具体内容参考 Rbpms-sdk API 文档）。

2.3.6 Task 转办

此接口用于任务的转办，需要调用的接口是

taskSupersede (

TaskSupersede supersede,

java.lang.String ak,

java.lang.String sk) ,

参数 TaskSupersede 封装了转办所需的必要内容（具体内容参考 Rbpms-sdk API 文档）。

2.3.7 退回拟稿人

此接口用于任务的退回拟稿人，需要调用的接口是

taskRejectToFirstAct (

TaskReject reject,

java.lang.String ak,

java.lang.String sk) ,

参数 TaskReject 封装了退回所需的必要内容（具体内容参考 Rbpms-sdk API 文档）。

2.3.8 退回上一节点

此接口用于任务的退回上个节点，需要调用的接口是

`taskRejectToPrevAct`(

`TaskReject` reject,

java.lang.String ak,

java.lang.String sk) ,

参数 `TaskReject` 封装了退回所需的必要内容（具体内容参考 Rbpms-sdk API 文档）。

2.3.9 收回 Task

此接口用于任务的收回，需要调用的接口是

`taskWithdraw`(

`TaskWithdraw` withdraw,

java.lang.String ak,

java.lang.String sk) ,

参数 `TaskWithdraw` 封装了收回所需的必要内容（具体内容参考 Rbpms-sdk API 文档）。

2.4 获取处理人信息

2.4.1 获取处理人信息

获取处理人信息接口用来查询当前处理人可以讲任务流转到下一节点的人员列表。此接口操作封装在类 `ProcessDefinitionService` 中。需要调用的接口是：

`getNextActivityAssignee`(

NextActivityAssignee query,

java.lang.String ak,

java.lang.String sk)

接口返回为用户集合。