Rcloud BPM
REST 服务 API 开发指南

北京中软国际信息技术有限公司

2014 年 5 月

目 录

第一章 概述

1.1 面向读者

本文档主要面向基于 Rcloud BPM 提供的 REST 服务接口来开发工作流应用的开发人员，列举了若干典型工作流应用场景以及 REST 服务接口（以下简称 BPM REST 服务）核心功能的用法和开发技巧，方便开发人员查阅 BPM 提供的 REST 服务接口的详细用法。

1.2 手册导读

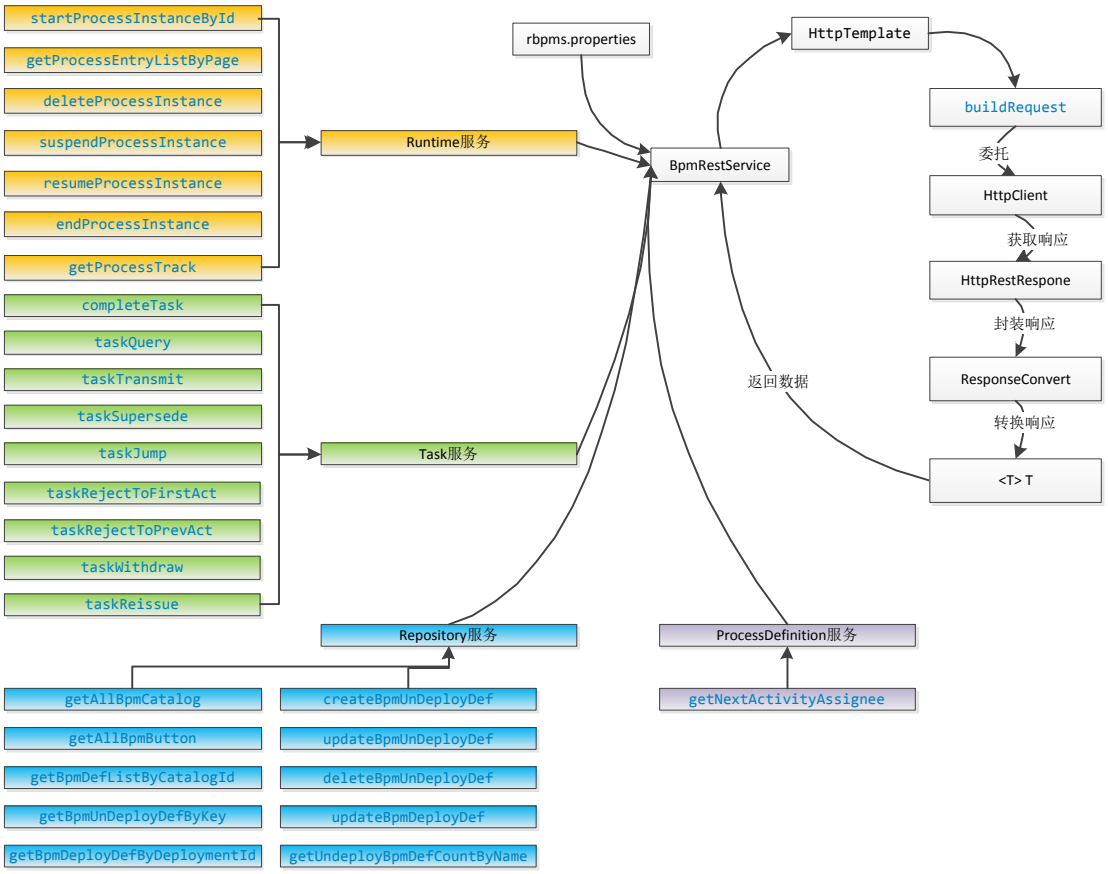
第一章为手册概述。

第三章详细介绍基于 BPM REST 服务 JAVA SDK API 开发的典型场景和各功能实现方法。

第二章 REST 服务 JAVA SDK 开发参考

2.1 REST 服务 JAVA SDK API 概述

BPM REST 服务 JAVA 开发工具包（Rcloud BPM REST JAVA SDK）是应用系统和 Rcloud BPM 提供的 REST 服务交互的主要途径。通过 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是时候完成对应的业务操作。SDK 调用结构如下图所示：



2.1.1 REST 服务 JAVA SDK API 列表

BPM 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。

2.1.2 SDK 统一调用方法

REST 服务 JAVA SDK 统一调用方法如下：

```
//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
long id= bpmRestService.getRuntimeService().startProcessInstanceById(start);
```

2.1.3 SDK 调用抛出异常说明

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。

- **BpmRestException**

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

2.2 集成 JAVA SDK

在 Web 应用中集成 BPM REST 服务 JAVA SDK 开发库，步骤如下：

1. 导入工作流开发组件库：

BPM REST 服务提供的 java 客户端 api 是以 jar 包的形式提供的，把此 jar 包和对应的依赖 jar 放入 lib 中即可。

2. 配置文件

把配置文件放在 classes 下。

2.3 Runtime 服务

2.3.1 启动流程

对于启动流程，有下面 2 种情况：

1. 直接启动——填写表单后直接提交启动流程
2. 选择从哪个活动启动——填写表单后，选择提交给哪个活动和参与者

创建流程实例并直接启动流程：

```
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);

long id= bpmRestService.getRuntimeService().startProcessInstanceById(start);
```

2.3.2 流程实例操作

2.3.2.1 结束流程

流程终止操作一般用于在特别情况下，人工干预流程，使其不按原定的流程规则流转，终止当前流程的执行。实现 API 是

```
bpmRestService.getRuntimeService().endProcessInstance("20140402100150.570001","8afac7de454ffc5f01454fff82450001");
```

2.3.2.2 恢复流程

流程恢复操作一般针对于已挂起的流程，使其恢复并按原定的流程规则流转，使当前流程继续执行。实现 API 是

```
bpmRestService.getRuntimeService().resumeProcessInstance("570001");
```

2.3.2.3 挂起流程

流程挂起操作一般用于在特别的业务场景下，人工干预流程，使其暂时停止流转。但可以通过恢复操作来恢复流程，一般恢复操作由具有相应权限的操作员执行。实现 API 是

```
bpmRestService.getRuntimeService().suspendProcessInstance("570001");
```

2.3.3 获取活动实例信息

通过活动实例 ID 获取实例信息，实现 API 是

```
//获取流程实例请求对象
//参数：流程实例 ID
ActivityInstGetRequest request = new ActivityInstGetRequest(activityInstId);
SFClient.newClient(sfServerHost).execute(request);
ActivityInstGetResponse resp = request.getResponse();
//流程实例对象
ActivityInstance activityInstance = resp.getActivityInstance();
```

请求类 ActivityInstGetRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	activityInstId	字符串	活动实例 ID	是

2.3.4 获取流程实例办理轨迹

通过流程实例 ID 获取流程办理轨迹信息，实现 API 是

```
//获取流程实例请求对象
//参数：流程实例 ID, includeSubs 是否包括子流程定义和实例信息。“1”-包括，“0”-不包括；若不设置某认为 0
ProcessInstTrackRequest request = new    ProcessInstTrackRequest(processInstId,includeSubs);
SFClient.newClient(sfServerHost).execute(request);
ProcessInstTrackResponse resp = request.getResponse();
//流程实例办理轨迹对象
ProcessInstanceTrack processInstanceTrack= resp.getProcessInstanceTrack();
```

请求类 ProcessInstTrackRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	processInstId	字符串	流程实例 ID	是
2	includeSubs	字符串	是否包括子流程定义和实例信息。“1”-包括，“0”-不包括。若不设置，默认为“0”	否

2.3.5 删除流程实例服务

删除指定的流程实例 id 的服务，实习 API 是

```
//删除流程实例请求对象
//参数：流程实例 ID,
ProcessInstDeleteRequest request=new ProcessInstDeleteRequest(processInstId);
SFClient.newClient(host).execute(request);
ProcessInstDeleteResponse resp=request.getResponse();
//获取状态：200 表示删除成功
int statusCode=resp.getStatusCode();
```

2.4 Task 服务

2.4.1 获取工作项信息

进入工作项办理页面，一般会在页面中缓存工作项信息数据，以便于工作项办理页面的渲染等操作，实现 API 是

```
//获取工作项请求对象
//参数：工作项 ID
WorkItemGetRequest request = new WorkItemGetRequest( workEffortId,
Context.getInstance().getCurrentPersonUuid());
SFClient.newClient(sfServerHost).execute(request);
WorkItemGetResponse resp = request.getResponse();
//工作项信息对象
WorkItem workItem = resp.getWorkItem();
```

请求类 WorkItemGetRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	workEffortId	字符串	工作项 ID	是
2	personUuid	字符串	人员 ID	是
3	commentScope	字符串	审批意见获取范围	否

2.4.2 特送工作项

特送也叫自由流，是能够直接指定流程跳到任意一个办理环节进行办理的操作，一般用于在特别情况下，人工干预流程，使其不按原定的流程规则流转，直接指定其办理环节。此功能多用于国内的场景，但我们建议通过建立支持这种特殊情况的流程规则定义来达到目的。

特送操作会将当前活动节点的所有工作项将被设置为“过期”状态。特送功能需要指定特送目标活动和参与者，对应的 API 是

```
String personUuid = Context.getInstance().getCurrentPersonUuid();
//创建特送请求对象，构造参数（工作项 ID，人员 ID）
WorkItemSendRequest request = new WorkItemSendRequest(workEffortId,personUuid);
request.setTimeLimits(buildTimeLimit(timeLimitsString));
request.setComments(buildComment(commentsString));
request.setContext(buildContext(contextString));
request.setSendActParticipant(buildSendAct(sendActivityString));
// 发送请求
SFClient.newClient(sfServerHost).execute(request);
WorkItemSendResponse resp = request.getResponse();
```

请求类 WorkItemSendRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	workEffortId	字符串	工作项 ID	是
2	personUuid	字符串	登录人的 uuid	否（不传的话后台自动获取）
3	SendActParticipant	SendActParticipant 对象	特送活动信息对象	是
4	TimeLimits	TimeLimit 对象数组	动态设定的下一活动时限集合	否
5	Comments	Comment 对象数组	办理意见对象数组	否

编号	参数名	参数类型	描述信息	是否必需
6	Context	Map 映射	上下文变量	否

使用的时候，可能需要给用户一个待选择的目标活动列表和该活动节点上的参与者列表，让用户自己选择特送的目标，即结合获取所有活动及参与者服务使用。

2.4.3 工作项办理

2.4.3.1 提交工作项

从任务列表进入表单完成表单处理后，用户需要提交完成工作项（也叫任务或工作项分配），使流程继续流转。提交工作项前要做的工作一般包括：

1、保存审批意见

```
//提交审批意见请求对象
//参数：活动实例 ID
CommentListCommitRequest request = new CommentListCommitRequest(activityInstId);
request.setPersonUuid(Context.getInstance().getCurrentPersonUuid());
// 设置审批意见
request.addComment(new Comment("yangguang", "yangguang 同意 OK"));
request.addComment(new Comment("fuming", "fuming 同意 OK"));
SFClient.newClient(sfServerHost).execute(request);
CommentListCommitResponse resp = request.getResponse();
```

2、提交完成工作项：

```
//提交工作项请求对象
//参数：工作项 ID、人员 ID
WorkItemCompleteRequest request = new WorkItemCompleteRequest(workEffortId,
personUuid);
request.setAppDataUuid(appDataUuid);
//设置启动审批意见（传递审批意见字符串并构造审批意见 Comment 对象）
request.setComments(buildComment(commentsString));
//设置下一办理人信息（传递下一办理人字符串并构造下一办理人集合
List<NextTransition>对象）。
request.setNextTransitions(buildNextAct(nextTransitionsString));
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
```

```
//设置办理时限（传递办理时限字符串并构造办理时限 List<TimeLimit>集合对象）
request.setTimeLimits(buildTimeLimit(timeLimitsString));
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//获取提交工作项响应
WorkItemCompleteResponse resp = request.getResponse();
```

一般而言，保存审批意见与提交完成工作项根据业务情况分开使用，提交工作项包含了审批意见数据的提交。

2.4.3.2 接收工作项

引擎在默认情况下，创建工作项时，这些工作项的状态为 CAL_ACCEPTED（已接收）状态。工作项接收适用于某些拒绝过的工作项，实现 API 是

```
//接收工作项请求对象
//参数：活动实例（工作项）ID、人员 ID
WorkItemAcceptRequest request = new WorkItemCompleteRequest(activityInstId,
personUuid);
request.setAppDataUuid(appDataUuid);
//设置启动审批意见（传递审批意见字符串并构造审批意见 Comment 对象）
request.setComments(buildComment(commentsString));
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
//设置办理时限（传递办理时限字符串并构造办理时限 List<TimeLimit>集合对象）
request.setTimeLimits(buildTimeLimit(timeLimitsString));
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//接收工作项响应
WorkItemAcceptResponse resp = request.getResponse();
```

2.4.3.3 退回工作项

办理者在办理当前工作时，由于某些原因，需要将流程退回给上一活动，由上一活动的办理人再次办理。实现 API 是：

```
//退回工作项请求对象
//参数：活动实例（工作项）ID、人员 ID
WorkItemWithdrawRequest request = new    WorkItemWithdrawRequest(workEffortId,
```

```
personUuid);
request.setAppDataUuid(appDataUuid);
//设置启动审批意见（传递审批意见字符串并构造审批意见 Comment 对象）
request.setComments(buildComment(commentsString));
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
//设置办理时限（传递办理时限字符串并构造办理时限 List<TimeLimit>集合对象）
request.setTimeLimits(buildTimeLimit(timeLimitsString));
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//退回工作项响应
WorkItemWithdrawResponse resp = request.getResponse();
```

2.4.3.4 转发工作项（委托/代办）

参与者可能由于工作需要，在一定的时间和应用范围内，将其负责的工作委托给其他人办理，转由此活动的其他办理者办理，实现 API 是

```
//转发工作项请求对象
//参数：工作项 ID、人员 ID
WorkItemDelegateRequest request = new WorkItemDelegateRequest(workEffortId,
personUuid);
request.setAppDataUuid(appDataUuid);
//设置启动审批意见（传递审批意见字符串并构造审批意见 Comment 对象）
request.setComments(buildComment(commentsString));
//设置转发人信息(传递转发人字符串并构造转发集合 List<NextTransition>对象,仅一个)。
request.setNextTransitions(buildNextAct(nextTransitionsString));
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
//设置办理时限（传递办理时限字符串并构造办理时限 List<TimeLimit>集合对象）
request.setTimeLimits(buildTimeLimit(timeLimitsString));
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//转发工作项响应
WorkItemDelegateResponse resp = request.getResponse();
```

实际上是对工作项的重新分配，注意，只能分配给当前活动的参与者！

2.4.3.5 收回委托工作项

同时也可以针对委托过的工作项做收回操作，实现 API 是：

```
//解除转发工作项请求对象
//参数：工作项 ID、人员 ID
//特殊说明：personUuid 指被解除委托的人员 ID
WorkItemUnDelegateRequest request = new WorkItemUnDelegateRequest(workEffortId,
personUuid);
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//解除转发工作项响应
WorkItemUnDelegateResponse resp = request.getResponse();
```

2.4.3.6 移除工作项

工作项的移除是指当前办理人认为工作项已经不适合再办理，如某些过期的情况，需要将此工作项移除。实现 API 是

```
//移除工作项请求对象
//参数：工作项 ID、人员 ID
WorkItemRemoveRequest request = new WorkItemRemoveRequest(workEffortId,
personUuid);
request.setAppDataUuid(appDataUuid);
//设置审批意见（传递审批意见字符串并构造审批意见 Comment 对象）
request.setComments(buildComment(commentsString));
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
//设置办理时限（传递办理时限字符串并构造办理时限 List<TimeLimit>集合对象）
request.setTimeLimits(buildTimeLimit(timeLimitsString));
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//移除工作项响应
WorkItemRemoveResponse resp = request.getResponse();
```

2.4.4 重激活工作项

重激活工作项服务，是指流程办理完成后，想重新激活流程上某个节点下参与者的已完成工作项，使之变成待办状态，同时流程变成运行状态，实现 API

是

```
//重新激活工作项请求对象
//参数：活动实例 ID,参与者 uuid
WorkItemReactivateRequest request=new WorkItemReactivateRequest(workeffortid,
personUuid);
SFClient.newClient(host).execute(request);
WorkItemReactivateResponse resp=request.getResponse();
//获取状态：200 表示重激活成功
int statusCode=resp.getStatusCode();
```

2.5 Repository 服务

2.5.1 获取全部流程目录

在启动流程或办理工作项时，可以动态指定下一办理活动及参与者，此时，需要动态获取下一办理活动及参与者数据。实现 API 是

```
//下一办理人请求对象
//流程未启动时，参数：启动流程包 ID、包版本、流程 ID、流程版本
//ParticipantNextActRequest request = new
ParticipantNextActRequest(packageId,packageVersion, processId, processVersion);
//流程启动后，参数：活动实例 ID
ParticipantNextActRequest request = new ParticipantNextActRequest(activityInstId);
//设置上下文（传递上下文字符串并构造上下文 Context 对象）
request.setContext(buildContext(contextString));
//设置人员 ID
request.setPersonUuid(Context.getInstance().getCurrentPersonUuid());
//发送请求
SFClient.newClient(sfServerHost).execute(request);
//获取下一办理人请求响应对象
ParticipantNextActResponse resp = request.getResponse();
NextActsParticipant nextActsParticipant = resp.getNextActsParticipant();
```

请求类 ParticipantNextActRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	activityInstId	字符串	活动实例 ID	非拟稿节点必需
2	packageId	字符串	包 ID	拟稿节点必需

编号	参数名	参数类型	描述信息	是否必需
3	packageVersion	字符串	包版本	拟稿节点必需
4	processId	字符串	流程 ID	拟稿节点必需
5	processVersion	字符串	流程版本	拟稿节点必需
6	context	Map 映射	上下文变量	否

2.5.2 获取所有流程按钮信息

在处理工作项，若需要特送操作时，需要获取当前流程所包含的所有活动及下属参与者数据。实现 API 是

```
//获取所有活动及参与者请求对象
//参数：流程实例 ID
ParticipantAllActRequest request = new ParticipantAllActRequest(processInstId);
SFClient.newClient(serverHost).execute(request);
ParticipantAllActResponse resp = request.getResponse();
AllActParticipant[] allActParticipant = resp.getAllActParticipants();
```

请求类 ParticipantAllActRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	processInstId	字符串	流程实例 ID。	是

2.5.3 根据流程目录 id，获取流程目录下的流程定义信息

在处理工作项，若需要转发（委托）操作时，需要获取当前系统中所有组织及人员数据。实现 API 是

```
//同步获取所有组织及人员请求对象
ParticipantGlobalOrgRequest request = new ParticipantGlobalOrgRequest();
SFClient.newClient(serverHost).execute(request);
ParticipantGlobalOrgResponse resp = request.getResponse();
Participant orgParticipantInfo = resp.getParticipant();
```

2.5.4 创建未发布的流程定义

在处理工作项，若需要转发（委托）操作时，以异步的方式获取当前系统中

的组织数据。实现 API 是

```
//异步获取组织请求对象
//参数：指定组织 ID，获取根组织时，orgId 为-1
ParticipantGlobalOrgAsynRequest request = new ParticipantGlobalOrgAsynRequest(orgId);
SFClient.newClient(serverHost).execute(request);
ParticipantGlobalOrgAsynResponse resp = request.getResponse();
Participant orgInfo = resp.getParticipant();
```

请求类 ParticipantGlobalOrgAsynRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	orgId	字符串	指定组织 ID，获取根组织时为-1。	是

2.5.5 修改未发布的流程定义文件

在处理工作项，若需要转发（委托）操作时，以异步的方式获取当前系统中组织人员数据。实现 API 是

```
//异步获取组织下所有人员请求对象
//参数：指定组织 ID
ParticipantGlobalOrgHumanRequest request = new
ParticipantGlobalOrgHumanRequest(orgId);
SFClient.newClient(serverHost).execute(request);
ParticipantGlobalOrgHumanResponse resp = request.getResponse();
Participant orgHuman = resp.getParticipant();
```

请求类 ParticipantGlobalOrgHumanRequest 结构说明:

编号	参数名	参数类型	描述信息	是否必需
1	orgId	字符串	指定组织 ID，获取根组织时为-1。	是

2.6 ProcessDefinition 服务

2.6.1 获取后续节点的参与者

```
bpmRestService.getRuntimeService().suspendProcessInstance("570001");
```