# 流程接口API

想对本文档的实现做了解，实现的可以联系作者联系方式如下

**戴凯** **个人邮箱13621151569@yeah.net**

目录

[流程接口API 1](#_Toc485210709)

[一 web端统一流程接口及参数规范 3](#_Toc485210710)

[1.1流程开启及表单提交任务的执行接口 3](#_Toc485210711)

[1.2表单数据列表项查询 4](#_Toc485210712)

[1.3待办列表项 5](#_Toc485210713)

[1.4列表项详情查询 7](#_Toc485210714)

[1.5审批接口 8](#_Toc485210715)

[1.6已办列表项 9](#_Toc485210716)

[1.7流程意见 11](#_Toc485210717)

[1.8任务意见 12](#_Toc485210718)

[二 服务器端调用流程服务器后台的接口规范 14](#_Toc485210719)

[2.1流程开启及表单提交任务的执行接口 14](#_Toc485210720)

[2.2待办列表项 14](#_Toc485210721)

[2.3审批接口 15](#_Toc485210722)

[2.4已办列表项 15](#_Toc485210723)

[2.5流程意见 15](#_Toc485210724)

[2.6任务意见 16](#_Toc485210725)

[三 流程服务器端的开发规范。 17](#_Toc485210726)

[3.1跟需求画流程图 17](#_Toc485210727)

[3.2设计实现类实现客户端流程各个映射接口的实现 17](#_Toc485210728)

## 一 web端统一流程接口及参数规范

注：接口规范仅仅是范例，具体应用场景可根据需求定制接口

### 1.1流程开启及表单提交任务的执行接口

接口名称及接口说明

**xxxx**/**uniformProc**/**startApply**

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**startApply

支持格式

**JSON**

HTTP请求方式

**POST**

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **leave** | true | string | 请假的表单项数据，具体业务视具体情况而定 |
| procType | false | string | 表单类型 |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/startApply

请求数据格式

**JSON示例**

{

"leave": {

"leaveDays": "3.5",

"leaveReason": "回家看看儿子...."

},

"procType": "01"

}

相应的返回结果

**JSON示例**

{

"message": "Ok",

"code": 200,

"data": {

"result": {"execute": "true"}

}

}

### 1.2表单数据列表项查询

接口名称及接口说明

xxxx/uniformProc/querySelfAppliesList

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**querySelfAppliesList

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **pageNo** | true | string | 页码数 |
| pageSize | true | string | 每页条目数 |
| procType | false | string | 表单类型 |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/querySelfAppliesList?pageNo=1&pageSize=10&procType=02

相应的返回结果

**JSON示例**

{

"message": "Ok",

"code": 200,

"data": {

"result": {"execute": "true"}

}

}

### 1.3待办列表项

接口名称及接口说明

xxxx/uniformProc/toDoList

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**toDoList

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **pageNo** | true | string | 页码数 |
| pageSize | true | string | 每页条目数 |
| procType | false | string | 表单类型 |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/toDoList?pageNo=1&pageSize=10&procType=02

相应的返回结果

**JSON示例**

{

"code": 200,

"data": {

"taskInfos": {

"tasks": [

{

"bizId": 16,

"bizType": "asdfasdf",

"createTime": 1497347420978,

"procInstId": "107674",

"procType": "03",

"status": "01",

"taskId": "107682",

"taskName": "直属主管审批"

},

{

"bizId": 15,

"bizType": "thf",

"createTime": 1497347409382,

"procInstId": "107664",

"procType": "03",

"status": "01",

"taskId": "107672",

"taskName": "直属主管审批"

}

],

"total": 2

}

},

"message": "Ok"

}

### 1.4列表项详情查询

接口名称及接口说明

xxxx/uniformProc/seeDetail

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**seeDetail

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| bizId | true | int | 业务主键id |
| procType | false | string | 表单类型 |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/seeDetail?bizId=3&procType=02

相应的返回结果

**JSON示例**

{

"code": 200,

"data": {

"result": {

"applyDate": 1497337494236,

"applyTm": "2017-06-13 15:04:54",

"bgnTm": "1497335160000",

"duration": "1",

"endTm": "1497417960000",

"goOutId": 34,

"offReason": "qqqq",

"operateTime": "1497337494236",

"operator": "11",

"procInstId": "107541",

"state": "01",

"user": "11",

"userName": "申请人"

},

"comments": [

{

"message": "Agree",

"taskId": "107549",

"id": "107551",

"fullMessage": "Agree",

"time": 1497337533710,

"processInstanceId": "107541",

"userId": "张三[项目主管]",

"action": "AddComment",

"type": "comment"

}

]

},

"message": "Ok"

}

### 1.5审批接口

接口名称及接口说明

xxxx/uniformProc/audit

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**audit

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| bizIdVal | true | string | 业务主键值 |
| state | true | int | 1：同意2：拒绝 |
| procType | false | string | 表单类型 |
| comment | true | string | 意见 |
| taskId | true | string | 任务id |

注意事项

输入参数示例

[http://localhost:8080/xxxx/uniformProc/audit](http://localhost:8080/pcms/uniformProc/audit)

{

"procType": "02",

"bizIdVal": "13",

"state": 1,

"comment": "同意",

"taskId":"87549"

}

相应的返回结果

**JSON示例**

{

"code": 200,

"data": {

"oppose": false,

"agree": true,

"auditors": "张三,李四"

},

"message": "Ok"

}

### 1.6已办列表项

接口名称及接口说明

xxxx/uniformProc/hasDoneList

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**hasDoneList

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **pageNo** | true | string | 页码数 |
| pageSize | true | string | 每页条目数 |
| procType | false | string | 表单类型 |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/hasDoneList?pageNo=1&pageSize=10&procType=02

相应的返回结果

**JSON示例**

{

"code": 200,

"data": {

"taskInfos": {

"tasks": [

{

"bizId": 16,

"bizType": "asdfasdf",

"createTime": 1497347420978,

"procInstId": "107674",

"procType": "03",

"status": "01",

"taskId": "107682",

"taskName": "直属主管审批"

},

{

"bizId": 15,

"bizType": "thf",

"createTime": 1497347409382,

"procInstId": "107664",

"procType": "03",

"status": "01",

"taskId": "107672",

"taskName": "直属主管审批"

}

],

"total": 2

}

},

"message": "Ok"

}

### 1.7流程意见

接口名称及接口说明

xxxx/uniformProc/procComments

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**procComments

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **procInstId** | true | string | 流程实例id |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/procComments?procInstId=87521

相应的返回结果

**JSON示例**

省略..

### 1.8任务意见

接口名称及接口说明

xxxx/uniformProc/taskComments

按条件查询项目信息 。

URL

**http://localhost:8**080/xxxx/uniformProc**/**taskComments

支持格式

**JSON**

HTTP请求方式

GET

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
|  | **必选** | **类型及范围** | **说明** |
| **taskId** | true | string | 任务id |

注意事项

输入参数示例

http://localhost:8080/xxxx/uniformProc/taskComments?taskId=87529

相应的返回结果

**JSON示例**

省略..

## 二 服务器端调用流程服务器后台的接口规范

注：接口规范仅仅是范例，具体应用场景可根据需求定制接口

注：数据仅仅具有参考意义，以实际执行数据为准

### 2.1流程开启及表单提交任务的执行接口

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("start");

applyAo.setBizId("11");

applyAo.setBizIdName("xxxxId");

applyAo.setUnitUser("11");

//返回的是流程实例id

ResponseData responseData = rpcInvoker.procAsist(applyAo);

String procInstId = responseData.getData().get("procInstId").toString();

### 2.2待办列表项

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("toDoList");

applyAo.setUnitUser("11");

applyAo.setTaskName("");

applyAo.setPageNo(1);

applyAo.setPageSize(10);

//执行返回的是TaskInfos对象

ResponseData responseData = rpcInvoker.procAsist(applyAo);

TaskInfos tasks = (TaskInfos)responseData.getData().get("taskInfos");

### 2.3审批接口

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("start");

applyAo.setUnitUser("11");

applyAo.setTaskId("55006");

applyAo.setBizIdName("xxxxId");

applyAo.setState(1);

applyAo.setArgs(argsModel.auditArgs());

applyAo.setComment("同意");

applyAo.setState(procVo.getState());

applyAo.setVariables(argsModel.auditVars());

//返回的是同意与否的boolean值

rpcInvoker.procAsist(applyAo);

### 2.4已办列表项

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("hasDoneList");

applyAo.setUnitUser("11");

applyAo.setTaskName("");

applyAo.setPageNo(1);

applyAo.setPageSize(10);

//执行返回的是TaskInfos对象

ResponseData responseData = rpcInvoker.procAsist(applyAo);

TaskInfos tasks = (TaskInfos)responseData.getData().get("taskInfos");

### 2.5流程意见

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("procCommentsQuery");

applyAo.setProcInstId("40006");

rpcInvoker.procAsist(applyAo);

### 2.6任务意见

**代码示例**

ApplyAo applyAo = new ApplyAo();

applyAo.setProcessName("reqLeave-GCA");

applyAo.setMethod("taskCommentsQuery");

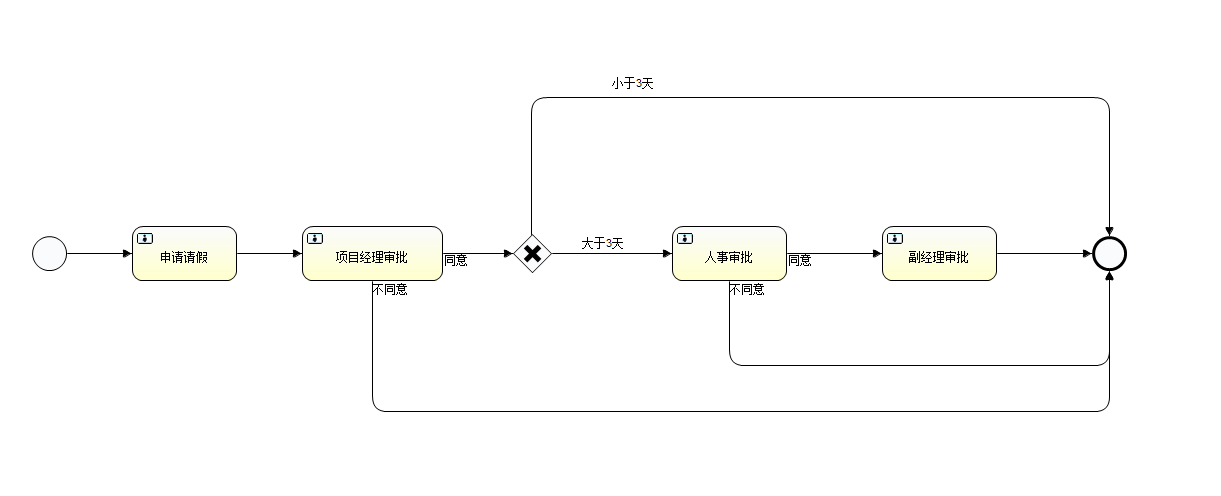
applyAo.setTaskId("60018");

rpcInvoker.procAsist(applyAo);

## 三 流程服务器端的开发规范。

注：接口规范仅仅是范例，具体应用场景可根据需求定制接口

### 3.1跟需求画流程图



### 3.2设计实现类实现客户端流程各个映射接口的实现

3.2.1 流程开启

**public** ResponseData start(ApplyAo applyAo) **throws** Exception {

ProcessHelper processHelper = (ProcessHelper)ContextUtil.*getBean*("processHelper");

ResponseData responseData = ResponseData.*ok*();

String procInstId = **null**;

**try** {

Map<String,Object> procVars=**new** HashMap<String,Object>();

procVars.put(applyAo.getBizIdName(), applyAo.getBizId());

String sql = "SELECT CHIEF\_ AS chiefId FROM ACT\_ID\_USER WHERE UNIT\_USER\_ = "+applyAo.getUnitUser()+"";

String chiefId = Persistence.*newPersistence*().queryChiefId(sql, "chiefId");

Map<String,Object> taskVars = **new** HashMap<>();

taskVars.put("chiefId", chiefId);

procInstId = processHelper.applyFormToAudit(applyAo.getProcessName(),procVars,applyAo.getBizId(),taskVars);

} **catch** (Exception e) {

**return** ResponseData.*customerError*().putDataValue("ececute", **false**).putDataValue("Error Reason", e.getMessage());

}

responseData.putDataValue("procInstId", procInstId);

**return** responseData;

}

3.2.2 待办

@Override

**public** ResponseData toDoList(ApplyAo applyAo) **throws** Exception {

TaskQueryService taskQuery = (TaskQueryService)ContextUtil.*getBean*("taskQueryService");

MemberShipService memberShipService = (MemberShipService)ContextUtil.*getBean*("memberShipService");

Map<String,Object> args=**new** HashMap<>();

args.put("unitUser", applyAo.getUnitUser());

MemberShip member = memberShipService.getActUserInfo(args);

**if**(member==**null**){

ResponseData.*customerError*().putDataValue("110", "该用户不存在审批身份信息！！");

}

String userId = member.getProcUser().getId();

String groupId = member.getGroup().getId();

PageInfo pageInfo = **new** PageInfo(applyAo.getPageNo(),applyAo.getPageSize());

Map<String,Object> map = **new** HashMap<>();

map.put("assignee", userId);

map.put("taskDefKey",groupId+ "Audit");

map.put("group", groupId);

map.put("pageNo", pageInfo.getPageNo());

map.put("pageSize", pageInfo.getPageSize());

TaskInfos taskInfos = **new** TaskInfos();

taskInfos.setTasks(taskQuery.toDoTaskQuery(map));

taskInfos.setTotal(taskQuery.toDoTaskCount(map));

**return** ResponseData.*ok*().putDataValue("taskInfos", taskInfos);

}

3.2.3 审批

@Override

**public** ResponseData audit(ApplyAo applyAo) **throws** Exception {

ProcessHelper processHelper = (ProcessHelper)ContextUtil.*getBean*("processHelper");

MemberShipService memberShipService = (MemberShipService)ContextUtil.*getBean*("memberShipService");

Map<String,Object> args=**new** HashMap<>();

args.put("unitUser", applyAo.getUnitUser());

MemberShip member = memberShipService.getActUserInfo(args);

**if**(member==**null**){

ResponseData.*customerError*().putDataValue("110", "该用户不存在审批身份信息！！");

}

List<Map<String,String>> auditArgs = applyAo.getArgs();

Task task = processHelper.findTaskById(applyAo.getTaskId());

String procDefId = task.getProcessDefinitionId();

String procDefName = procDefId.substring(0, procDefId.indexOf(":"));

Map<String,Object> variables = applyAo.getVariables();

//取出用户，角色信息

ProcUser currentUser = member.getProcUser();

Group currentGroup = member.getGroup();

String finalRoles = SysDicUtil.*getFieldDesc*("process", procDefName,"02");

List<String > roles = Arrays.*asList*(finalRoles.split(","));

//规则1是固定的

String rules01 = SysDicUtil.*getFieldDesc*("rule", procDefName,"01");

**boolean** rt01 = DynUtil.*getResult*(rules01, **new** Class[]{**int**.**class**}, **new** Object[]{applyAo.getState()});

//规则2是根据参数动态改变的

String rules02 = SysDicUtil.*getFieldDesc*("rule", procDefName,"02");

**int** argSize = auditArgs.size();

Class[] clazzArgs = **new** Class[argSize];

Object[] valArgs = **new** Object[argSize];

ArgsProcess argsProcs = ArgsProcess.*newArgsProcess*().processArgs(auditArgs, clazzArgs, valArgs);

**boolean** rt02 = DynUtil.*getResult*(rules02, argsProcs.getClazzArgs(), argsProcs.getValArgs());

//规则3也是根据组或用户来改变的

String rules03 = SysDicUtil.*getFieldDesc*("rule", procDefName,"03");

**boolean** rt03 = **false**;

String curRole = currentGroup.getName();

**for**(String role : roles){

**boolean** rt = DynUtil.*getResult*(rules03, **new** Class[]{String.**class**,String.**class**}, **new** Object[]{curRole,role});

rt03 = rt03 || rt;

}

**boolean** finalLessOK = rt01 && rt02;

**boolean** finalLessNO = rt01 == **false** && rt02;

**boolean** finalMoreOK = rt01 && rt03;

**boolean** finalMoreNO = rt01 == **false** && rt03;

**if**(finalLessOK || finalMoreOK){

variables = processHelper.finalAuditOk(applyAo.getTaskId(), applyAo.getBizIdName(), variables);

}**else** **if**(finalLessNO || finalMoreNO){

variables = processHelper.finalAuditNo(applyAo.getTaskId(), applyAo.getBizIdName(), variables);

}**else**{

**if**(rt01)

variables.put("msg", "通过");

**else**

variables = processHelper.finalAuditNo(applyAo.getTaskId(), applyAo.getBizIdName(), variables);

}

Authentication.*setAuthenticatedUserId*(currentUser.getFirstName()+currentUser.getLastName()+"["+currentGroup.getName()+"]");

processHelper.addComment(task, applyAo.getComment());

variables.put("state", applyAo.getState()==**null**?1:applyAo.getState());

processHelper.completeTask(task, variables);

**boolean** finalOK = finalLessOK || finalMoreOK;

**boolean** finalNO = !rt01;

**return** ResponseData.*ok*().putDataValue("agree", finalOK).putDataValue("oppose", finalNO);

}

3.2.4 已办

**public** ResponseData hasDoneList(ApplyAo applyAo) **throws** Exception {

TaskQueryService taskQuery = (TaskQueryService)ContextUtil.*getBean*("taskQueryService");

MemberShipService memberShipService = (MemberShipService)ContextUtil.*getBean*("memberShipService");

Map<String,Object> args=**new** HashMap<>();

args.put("unitUser", applyAo.getUnitUser());

MemberShip member = memberShipService.getActUserInfo(args);

**if**(member==**null**){

ResponseData.*customerError*().putDataValue("110", "该用户不存在审批身份信息！！");

}

String userId = member.getProcUser().getId();

String groupId = member.getGroup().getId();

PageInfo pageInfo = **new** PageInfo(applyAo.getPageNo(),applyAo.getPageSize());

Map<String,Object> map = **new** HashMap<>();

map.put("assignee", userId);

map.put("taskDefKey",groupId+ "Audit");

map.put("group", groupId);

map.put("pageNo", pageInfo.getPageNo());

map.put("pageSize", pageInfo.getPageSize());

TaskInfos taskInfos = **new** TaskInfos();

taskInfos.setTasks(taskQuery.hasDoneTaskQuery(map));

taskInfos.setTotal(taskQuery.hasDoneTaskCount(map));

**return** ResponseData.*ok*().putDataValue("taskInfos", taskInfos);

}

3.2.5 流程意见

@Override

**public** ResponseData procCommentsQuery(ApplyAo applyAo) **throws** Exception {

ProcessHelper processHelper = (ProcessHelper)ContextUtil.*getBean*("processHelper");

**if** (applyAo.getProcInstId() == **null**) {

**return** ResponseData.*ok*().putDataValue("result", "");

}

List<Comment> comments = processHelper.findCommentsByProcInstId(applyAo.getProcInstId());

**return** ResponseData.*ok*().putDataValue("comments", comments);

}

3.2.6 任务意见

@Override

**public** ResponseData taskCommentsQuery(ApplyAo applyAo) **throws** Exception {

ProcessHelper processHelper = (ProcessHelper)ContextUtil.*getBean*("processHelper");

String taskId = applyAo.getTaskId() == **null** ? "" : applyAo.getTaskId();

List<Comment> hisComments = processHelper.findHisCommentsByTaskId(taskId);

**return** ResponseData.*ok*().putDataValue("comments", hisComments);

}