服务管制-仓库中心接口文档

## 一、仓库中心 接口

### (一). 根据查询条件获取仓库信息

输入：

json={

"key":"repertory.metadata.search",

"page":{"pageNo":"1","pageSize":"20"},

"data":{ //data得有

"servicename":"rcs\_weka",//没有就不写

"serviceversion":"5.6.0",//没有就不写

"md5":"e089067fd20b5a323c04ff194213f01e",//没有就不写

"deploymentFlag":"1",//没有就不写，未发布1

"productname":"rcs"//没有就不写

“desployenv”:”4/1” //过滤,4—测试环境，1—运维镜像

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":[{"mwId":1197,"productname":"rcs",//服务类型，即属于哪个产品

"servicename":"rcs\_weka",

"servicefunction":"风控模型创建",

"servicetype":"2",

"serviceversion":"5.6.0",

"md5":"e089067fd20b5a323c04ff194213f01e",

"buildtime":"2014-09-26 11:13:59",

"svnversion":"47",//这个暂时没用

"bugids":"8491",

"svnpath":"svn://172.17.3.161/XGD-CXYW/risk\_control/trunk/src/rcs\_weka",

"deploymentFlag":"0",

"importtime":"2014-09-26 00:00:44",

"importpersion":"miaomiao",

"desploypath":"/root/adsad",

“javamain”:”xgd.com.fun.nac\_chn\_service.ServerLunsh”

}],

"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (二). 输入svn路径导入仓库数据

输入：

json= {

"key":"repertory.metadata.create",

"data":{

"svnurl":"svn://172.17.3.161/xgd\_rel/rcs/risk\_control/5.6.0/apps/rcs\_weka.tar.gz",//多个请以$隔开

"loginname":"miaomiao" //登陆用户名

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":-1,"pageSize":-1,"totalPage":-1,"total":-1}

### (三). 仓库中心显示树

输入：

json = {

"key":"repertory.metadata.tree",

"data":{

“envType”:”5”,

“desployenv”:”1/4”//过滤

"deploymentFlag":"1",//未发布1

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":[{"id":1178,

"productname":"rcs",

"child":[{"serviceid":1188,"servicename":"rcs\_weka",

“ss\_type”:”1”//1：C服务，2：web服务，3：java服务

,"desploypath":"/tmp/","defaultcfg":”config.xml”,"defaultcfgid":”1000”,

"datelist":["2014-09"]}]}],

"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (四). 点击树形结构里的服务名，获取查询结果

输入：

json = {

"key":"repertory.metadatasvr.search",

"page":{"pageNo":"1","pageSize":"20"},

"id":"1188"//服务名ID,

“data”:{

“desployenv”:”1/4”//过滤

"deploymentFlag":"1",//未发布1

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":[{"mwId":1197,"productname":"rcs","servicename":"rcs\_weka","servicefunction":"风控模型创建","servicetype":"2","serviceversion":"5.6.0","md5":"e089067fd20b5a323c04ff194213f01e","buildtime":"2014-09-26 11:13:59","svnversion":"47","bugids":"8491","svnpath":"svn://172.17.3.161/XGD-CXYW/risk\_control/trunk/src/rcs\_weka","deploymentFlag":"0","importtime":"2014-09-28 00:00:00","importpersion":"miaomiao",“javamain”:”xgd.com.fun.nac\_chn\_service.ServerLunsh”

}],

"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (五). 点击树形结构里的时间，获取查询结果

输入：

json = {

"key":"repertory.metadatatime.search",

"page":{"pageNo":"1","pageSize":"20"},

"data":{

"id":"1188",//服务名对应的ID

"time":"2014-09" //页面上的时间格式 “desployenv”:”1/4”//过滤1-运维镜像，2--生产，3--测试镜像，4--测试，5--开发

"deploymentFlag":"1",//未发布1

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":[{"mwId":1197,"productname":"rcs","servicename":"rcs\_weka","servicefunction":"风控模型创建","servicetype":"2","serviceversion":"5.6.0","md5":"e089067fd20b5a323c04ff194213f01e","buildtime":"2014-09-26 11:13:59","svnversion":"47","bugids":"8491","svnpath":"svn://172.17.3.161/XGD-CXYW/risk\_control/trunk/src/rcs\_weka","deploymentFlag":"0","importtime":"2014-09-28 00:00:00","importpersion":"miaomiao",“javamain”:”xgd.com.fun.nac\_chn\_service.ServerLunsh”

}],

"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (六)根据ID删除服务元数据

输入：

{

"key":"repertory.metadata.delete",

"id":"1197"

}

输出：{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (七)添加部署路径（修改部署路径也是这个）

输入：

json = {

"key":"repertory.desploypath.update",

"id":"1188",

"data":{

"path":"/home/oracle/test",

“cfgid”:”100”,

“envType”:”5”

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (八)获取默认的配置和默认的部署路径

输入：

json = {

"key":"repertory.defaultdesploy.get",

"id":"1188"

“data”:{  
"envType":"5"

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":{"path":"/home/oracle/omp\_zk","defaultcfgid":"1629","defaultcfg":"mytest"},"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (九)根据mwid查询操作历史记录

输入：

json = {

"page":{"pageNo":"1","pageSize":"20"},

"key":"repertory.history.search",

“data”:{

"id":"1188",  
"begintime":"2014-12-10",

"endtime":"2014-12-10",

"person":"admin”,

"operationtype":"1"

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":[{"opId":"5675","mwId":"5445","mwEntry":"admin","mwDateTime":null,"mwTime":"2014-12-17 14:14:20","mwOperation":"重新将omp-zookeeper\_0.0.1 导入仓库中心","mwOperationtype":"1"}],"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (十)根据产品id获取mw记录

输入：

json={

“key”:”repertory.metadataproduct.search”,

“id”:”1111”,//产品id

“data”:{

“desployenv”:”1/4”//过滤1-运维镜像，2--生产，3--测试镜像，4--测试，5--开发

"deploymentFlag":"1",//未发布1

},

“page”:{"pageNo":1,"pageSize":20}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":[{"opId":"5675","mwId":"5445","mwEntry":"admin","mwDateTime":null,"mwTime":"2014-12-17 14:14:20","mwOperation":"重新将omp-zookeeper\_0.0.1 导入仓库中心","mwOperationtype":"1"}],"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (十一)点击服务名获取web模块信息

输入：

json = {

"key":"repertory.modules.search",

“data”:{

" svrid":"1188",//服务名id

" svrname":"omp-zookeeper”,//服务名

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":{

" replace ":["resources","test/ooo"],//过滤的目录

" pluge ":["lib"]},

"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (十二)更新web模块信息

输入：

json = {

"key":" repertory.modules.update",

“data”:{

" svrid":"1188",//服务名id

" svrname":"omp-zookeeper”,//服务名

" replace":"Conf/config,Server/proper",//过滤的目录，多个逗号隔开

“pluge”:”Server/lib”,//一致的目录，多个逗号隔开

}

{"errorCode":0,//非0失败

"errorMessage":"SUCCESS",//非0时表示失败的原因

"data":"","pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (十三)设置元数据显示列

输入：

json = {

"key":" repertory.menusetting.update",

“data”:{

" menuSetting":"admin”,//登陆名

" menuSetting":"servicename,version……",//选中的列

}

输出：

{"errorCode":0,//非0失败

"errorMessage":"SUCCESS",//非0时表示失败的原因

"data":"","pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (十四)获取元数据显示列

输入：

json = {

"key":" repertory.modules.search",

“data”:{

" menuSetting":"admin”//登陆名

}

输出:

{"errorCode":0,"errorMessage":"SUCCESS","data":{"meId":"13884","menuSetting":"servicename,version","loginname":"miaomiao"},"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (十五)是否覆盖resources/init.sh目录

输入：

json = {

"key":" repertory.javaremodules.search",

“data”:{

"svrid":"13830",

"svrname":"omp\_groupweight",

"type":"resources"//或者init.sh

}

输出:

{"errorCode":0,"errorMessage":"SUCCESS","data":{

“2”//1—是2—否 3—是 4—否

},"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (十六)更新resources或init.sh是否覆盖

输入：

json = {

"key":" repertory.javaremodules.update",

“data”:{

"replace":"3",//1-是2-否(12表示resources更新)3-是4否(34表示init.sh更新)

"svrid":"13830",

"svrname":"omp\_groupweight"

}

输出:

{"errorCode":0,"errorMessage":"SUCCESS","data":{

“2”//1—是2—否 3—是 4—否

},"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

## 二、部署脚本接口

### (一). 添加部署脚本

输入：

json = {

"key":"repertory.shellscript.create",

"data":{

"smTyProperty":"00",

"smName":"c\_start.sh",

smAddress:"/home/oracle/haha",

"smParameters":"$1--name, $2--pwd, $3--svrname",

"smFunction":"start up c service",

"smDemo":"脚本执行方式"

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":-1,"pageSize":-1,"totalPage":-1,"total":-1}

### (二). 修改部署脚本

输入：

json = {

"key":"repertory.shellscript.update",

"id":"1203",

"data":{

"smTyProperty":"00",

"smName":"c\_start.sh",

smAddress:"/home/oracle/haha",

"smParameters":"$1--name, $2--pwd, $3--update",

"smFunction":"start up c service",

"smDemo":"脚本执行方式"

}

}

输出：{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":-1,"pageSize":-1,"totalPage":-1,"total":-1}

### (三). 删除部署脚本、

输入：

json = {

"key":"repertory.shellscript.delete",

"id":"1203"

}

输出：{"errorCode":0,"errorMessage":"SUCCESS","data":"","pageNo":-1,"pageSize":-1,"totalPage":-1,"total":-1}

### (四). 获取部署脚本信息(分页)

输入：

json = {

"key":"repertory.shellscript.search",

"page":{"pageNo":"1","pageSize":"20"}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS",

"data":[{"smId":1203,

"smTyProperty":"00",

"smName":"c\_start.sh",

"smAddress":"/home/oracle/haha",

"smParameters":"$1--name, $2--pwd, $3--svrname",

"smFunction":"start up c service",

"smDemo":"脚本执行方式"}],

"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (五). 根据ID获取单个详细信息

输入：

json = {

"key":"repertory.shellscript.get",

"id":"1204"

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":{"smId":1204,"smTyProperty":"01","smName":"c\_stop.sh","smAddress":"/home/oracle/haha","smParameters":"$1--name, $2--pwd, $3--svrname","smFunction":"stopup c service","smDemo":"注释"},"pageNo":0,"pageSize":0,"totalPage":0,"total":0}

### (六). 调用初始化脚本初始机器

输入：

json = {

"key":"repertory.initscript.create",

"data":{

"id":"6344,5847",//脚本id

“ipv”:”172.17.3.85,172.17.3.86”,

“person”:”admin”

}

}

输出：//脚本执行结果在历史记录中查看

{"errorCode":0,//非0失败

"errorMessage":"SUCCESS" }

### (七). 获取初始化机器历史记录（分页）

输入：

json = {

"key":"repertory.scripthistory.search",

"page":{"pageNo":"1","pageSize":"20"}

“data”:{

“ipv”:”172.17.3.84”,

“person”:”admin”,

“begintime”:”2014-10-10”,

“endtime”:”2014-11-11”

}

}

输出：

{"errorCode":0,"errorMessage":"SUCCESS","data":[{" inId ":"5675"," inEntry":"操作人" ," oparateTime ":"2014-12-17 14:14:20"," inIpv ":"172.17.3.84"," inResult ":"操作结果"," inLog ":"脚本日志"}],"pageNo":1,"pageSize":20,"totalPage":1,"total":1}

### (八). 获取初始化脚本操作日志

输入：

json = {

"key":" repertory.scripthistory.get",

“id”:”12222”

}

输出：

{errorCode=0, errorMessage=SUCCESS, data=<br>shelltest1.sh:<br>tart to exec shelltest1...

172.17.3.84

root

123456

<br><br>shelltest2.shÈÕÖ¾Îª :<br>tart to exec shelltest1...

172.17.3.84

root

123456

<br><br>shelltest3.shÈÕÖ¾Îª :<br>tart to exec shelltest1...

172.17.3.84

root

123456

<br>, pageNo=0, pageSize=0, totalPage=0, total=0}