1.（rms客户部门）客户端服务，directUrl="127.0.0.1：9100意思就是调用接口时，开放给本机的9100端口给服务端（服务端ip必须跟客户端ip一致）监听"即机器人（callback服务，）把*RobotTaskCallbackFacade（submit）的当作端口9100，用来让服务器监听，当客户机调用接口的submit方法时，就会调用motan服务的实现接口RobotTaskCallbackFacadeImpl中的重写submit方法即第二步*

<!-- RobotTaskCallback -->

<motan:referer id=*"robotTaskCallbackFacade"* interface=*"com.geekplus.athena.facade.RobotTaskCallbackFacade"* directUrl=*"127.0.0.1：9100 "* requestTimeout=*"3000"* />（验证仿真时需要把/usr/local/geekplus/tomcat-rms/webapps/athena/WEB-INF/classes/config/spring/spring-motan-rpc.xml端口改成28.45本机监听）

2（模拟wms服务器部分）服务器服务，

<!-- athena调用的服务 -->监听*com.geekplus.athena.facade.RobotTaskCallbackFacade 9100端口，执行实现接口*

*Class=com.geekplus.test.athena.facade.impl.RobotTaskCallbackFacadeImpl的方法（重写）可以添加(模拟confirm,wms发过来的)*

<bean id=*"robotTaskCallbackFacadeImpl"* class=*"com.geekplus.test.athena.facade.impl.RobotTaskCallbackFacadeImpl"* />

<motan:service interface=*"com.geekplus.athena.facade.RobotTaskCallbackFacade"* ref=*"robotTaskCallbackFacadeImpl"* export=*"9100"* />

执行重写接口*RobotTaskCallbackFacadeImpl（submit，可以加入confirm（）方法）*

**public** **class** RobotTaskCallbackFacadeImpl **implements** RobotTaskCallbackFacade {

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(RobotTaskCallbackFacadeImpl.**class**);

RPCclient client = **new** RPCclient();

**private** **static** List<RobotTask> *responseTasks* = **new** ArrayList<RobotTask>();

**public** **static** List<RobotTask> getResponseTasks() {

**return** *responseTasks*;

}

**public** **int** submit(RobotTask task) {

***logger***.info("submit task:{}", JSONUtil.*objToJson*(task));

**if**(TaskPhase.***SHELF\_ARRIVED***.equals(task.getTaskPhase())&&SHELF\_STATE.***READY***.equals(task.getShelfState())){

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

task.setShelfState(SHELF\_STATE.***GO\_RETURN***);

//responseTasks.add(task);

client.confirmTask(task);

}

//client.confirmTask(task);

**return** 0;

}

**public** **int** swapShelfPlacement(String arg0, Double arg1, String arg2, Double arg3) {

// **TODO** Auto-generated method stub

**return** 0;

}