Bs3学习笔记

## Main方法,框架的启动入口

整个框架的学习，从main函数开始

com.umpay.front3.core.Main

|  |
| --- |
| public class Main {  public static void main(String[] args) throws Exception {  if (args == null || args.length == 0) {  System.out.println("请输入应用上下文配置文件路径!");  System.exit(-1);  }  addSecurityProvider();  String[] context = args[0].split(",", -1);  Context4Spring SprintContext = Context4Spring.getInstance(context);  Service4Inf r = (Service4Inf) SprintContext.getBean("ServiceMain");  r.start();  }  /\*\*  \* 增加SecurityProvider  \*/  private static void addSecurityProvider() {  try {  Security.addProvider(new com.sun.crypto.provider.SunJCE());  Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());  } catch (Exception e) {  }  }  } |

Front3\_common.xml

|  |
| --- |
| <!--主服务-->  <bean id="ServiceMain" class="com.bs2.core.ServiceMain" singleton="true">  <description>主服务：自动启动所有子服务</description>  <property name="\_context"><ref bean="ServiceContext" /></property>  <property name="\_manager"><ref bean="**ServiceGroup**" /></property>  <property name="\_exithook"><ref bean="ServiceHooking" /></property>  </bean>  <bean id="ServiceContext" class="com.bs2.core.Context4Spring" factory-method="getInstance" singleton="true" />  <bean id="ServiceHooking" class="com.umpay.front3.core.Hooking"></bean> |

app\_my.xml

|  |
| --- |
| <bean id="**ServiceGroup**" class="com.bs2.core.ServiceGroup">  <property name="\_context">  <ref bean="ServiceContext" />  </property>  <property name="\_namelist">  <list>  <!--1.服务队列-->  <value>recvUMQ</value>  <value>send2UMQ</value>  <value>recvRemoteQ</value>  <value>send2RemoteQ</value>  <!--2.C/S FOR UM-->  <value>Client\_UM\_HTTP</value>  <!--3.C/S FOR RM-->  <value>Server\_RM\_HTTP</value>  <!--6.定时器-->  <value>TickletGroup</value>  <!--7.本地模拟测试-->  <!--<value>BusiTrigger</value>-->  </list>  </property>  </bean> |

看下com.bs2.core.ServiceMain.start() 做了什么事情

|  |
| --- |
| public void start() {  int DL = Util.getDynamicLoading();// 动态加载级别，Ticklet/QObj;  // 显示运行状态  Runtime rt = Runtime.getRuntime();  StringBuffer sbInfo = new StringBuffer();  sbInfo.append("============= ServiceMain v3.4.110526 ============= ")  .append(Util.strDateTime()).append(CRLF);  sbInfo.append(Util.pn("Start Time", PLEN)).append(": ")  .append(Util.strDateTime()).append(CRLF);  sbInfo.append(Util.pn("CopyLeft", PLEN))  .append(": (1997-2011) Nike.Lius@GMail.Com").append("/DL:")  .append(DL).append(CRLF);  ………………….  ………………….  // 依次初始化各种服务 services  Object[] serviceIds = \_manager.getNamesAsArray();  if (\_context != null) {  for (int i = 0; i < serviceIds.length; i++) {  if (serviceIds[i] == null)  continue;  String svcId = (String) serviceIds[i];  Service4Inf svcOne = (Service4Inf) \_context.getBean(svcId,  Service4Inf.class);  \_log.debug("### START 服务(ID:" + svcId + ")/CLASS:"  + svcOne.getClass().getName());  svcOne.start(); // 启动线程运行！启动顺序由配置先后决定！  String svcName = Util.strBlank(svcId, 22);// 18,20  String svcInfo = svcOne.getInfo(); // 确保先start后getInfo！  if (i < 10)  sbInfo.append(' ');  sbInfo.append(i).append('.').append(svcName).append(": ");  sbInfo.append(svcInfo).append(CRLF);  }  }  sbInfo.append("=================================================== ")  .append(Util.strDateTime()).append(CRLF);  System.out.print(sbInfo); // 输出显示  if (\_context != null) {  Set beanset = \_context.getAllBeanNames(); // 导致创建所有的Bean，无法LazyCreate  if (\_log.isInfoEnabled())  \_log.info("Found Define Beans " + beanset.size() + ":" + beanset);// Spring Beans  Map beanmap = \_context.getAllActiveBeans(); // 导致创建所有的Bean，无法LazyCreate  if (\_log.isInfoEnabled())  \_log.info("Found Active Beans " + beanmap.size() + ":" + beanmap.keySet());  if (\_log.isInfoEnabled())  \_log.info("# start()...ok/Services:" + serviceIds.length "/DL-Level:" + DL);  }  } |