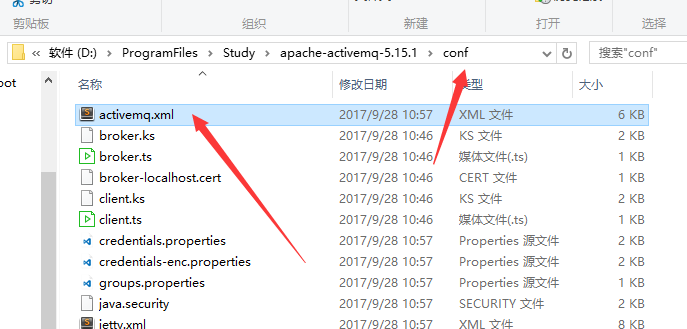
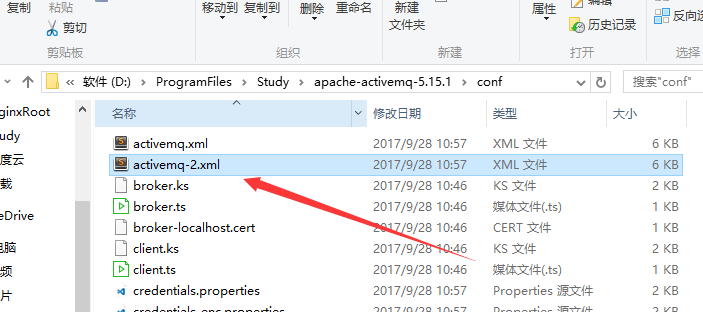
# 1、重新选择activemq启动的配置文件

## 1、正常的activemq启动时依靠的配置文件为

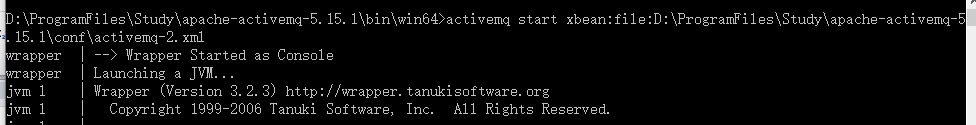


## 2、重新创建一个配置文件，并利用它启动

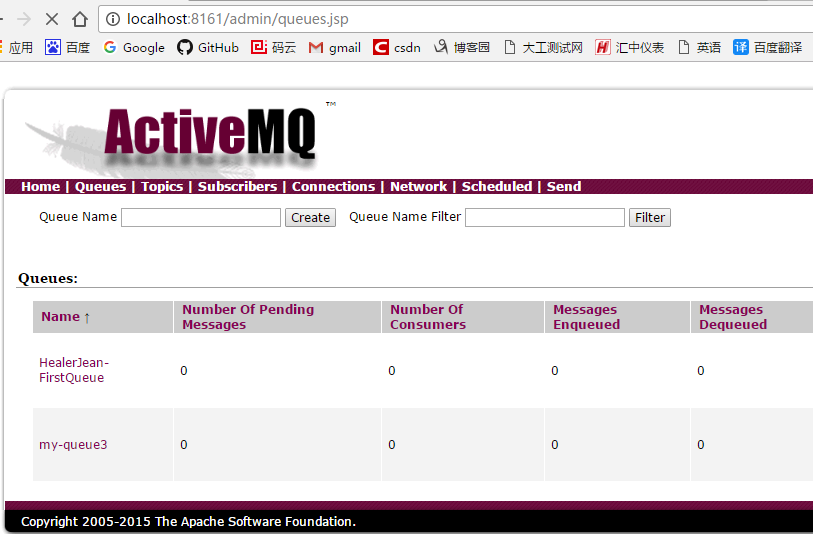


## 3、命令行启动

activemq start xbean:file: D:\ProgramFiles\Study\apache-activemq-5.15.1\conf\activemq-2.xml



## 4、启动成功

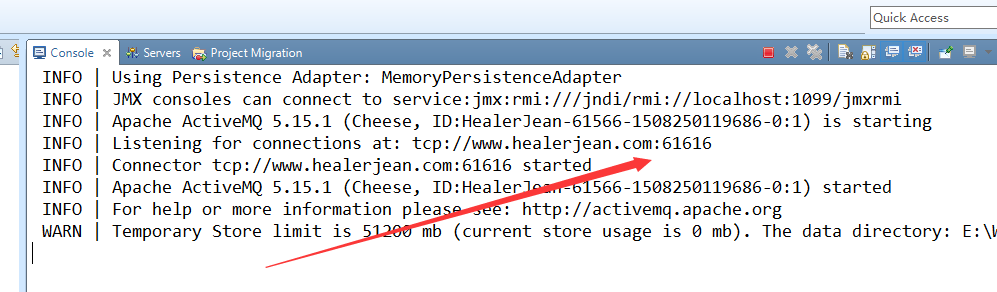


# 2、Java中内嵌启动broker

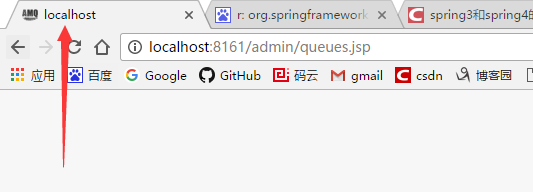
## 1、第一种方式

### 1、BrokerService启动代码

|  |
| --- |
| **public** **class** BrokerOne {  **public** **static** **void** main(String[] args)**throws** Exception {    BrokerService broker = **new** BrokerService();    broker.setUseJmx(**true**);    broker.addConnector("tcp://localhost:61616");    broker.start();    }  } |



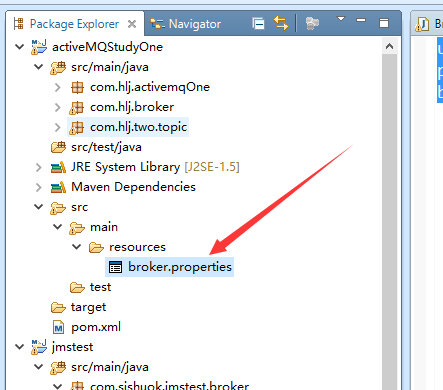
### 2、这个时候运行生产者和消费者是没有问题的，但是浏览器中没有管理页面了



## 2、第二种，读取properties方式（src或者resource资源文件下）

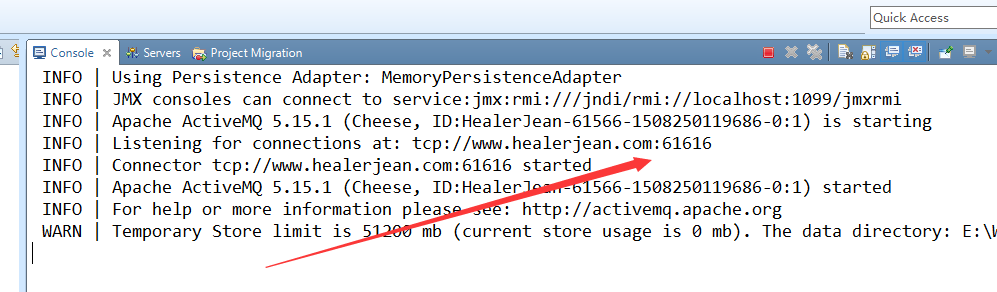
### 1、properties配置文件为

|  |
| --- |
| useJmx=true  persistent=false #要不要持久化  brokerName=Cheese |



### 2、BrokerFactory启动代码

|  |
| --- |
| **public** **class** BrokerTwo {  **public** **static** **void** main(String[] args)**throws** Exception {    String Uri = "properties:broker.properties";  BrokerService broker1 = BrokerFactory.*createBroker*(**new** URI(Uri));  broker1.addConnector("tcp://localhost:61616");  broker1.start();    }  } |



## 3、第三种，spring配置文件xml的方式

## 1、启动代码

|  |
| --- |
| **public** **class** BrokerThreeSprignXml {  **public** **static** **void** main(String[] args)**throws** Exception {    ApplicationContext context = **new** ClassPathXmlApplicationContext("broker-applicationContext.xml");    }  } |

## 2、相关spring配置文件如下

|  |
| --- |
| <bean id=*"broker"* class=*"org.apache.activemq.broker.BrokerService"* init-method=*"start"* destroy-method=*"stop"*>  <property name=*"brokerName"* value=*"myBroker"* />  <property name=*"persistent"* value=*"false"* />  <property name=*"transportConnectorURIs"*>  <list>  <value>tcp://localhost:61616</value>  </list>  </property>  </bean> |