# Step by Step guide to deploy Spring Batch Admin application on DB2/Websphere environment

# Tools used:

## Apache-maven-3.0.4 & Java

JAVA\_HOME C:\Program Files\Java\jdk1.7.0\_45

M2 %M2\_HOME%\bin;

M2\_HOME C:\springsource\apache-maven-3.0.4

MAVEN\_OPTS -Xms256m -Xmx512m

## Update env PATH

%M2%;%JAVA\_HOME%\bin;

## Download a suitable batch source from the below repository

<http://docs.spring.io/downloads/nightly/release-download.php?project=BATCH>

# Building Spring Batch Admin application

The suggested modifications are capable of using an existing datasource connection or create a new datasource of its own to connect to DB2 database on a websphere 5.1 application server.

## Overriding the bean by placing a new context file to use an existing datasource

**Modify web.xml**

<resource-ref>

<res-ref-name>**jdbc/reportingManagerCustDbDataSource**</res-ref-name>

<res-type>**javax.sql.DataSource**</res-type>

<res-auth>**Container**</res-auth>

<res-sharing-scope>**Shareable**</res-sharing-scope>

</resource-ref>

**C:\software\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-sample\src\main\resources\META-INF\spring\batch\override\myer-context.xml**

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<beans xmlns=**"http://www.springframework.org/schema/beans"** xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"** xmlns:jee=**"http://www.springframework.org/schema/jee"**

xmlns:batch=**"http://www.springframework.org/schema/batch"** xmlns:p=**"http://www.springframework.org/schema/p"** xmlns:jdbc=**"http://www.springframework.org/schema/jdbc"**

xsi:schemaLocation=**"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd**

**http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-3.0.xsd**

**http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-2.5.xsd**

**http://www.springframework.org/schema/batch http://www.springframework.org/schema/batch/spring-batch-2.1.xsd"**>

<batch:job-repository id=**"jobRepository"** table-prefix=**"FOO\_"**/>

<bean id=**"jobExplorer"** class=**"org.springframework.batch.core.explore.support.JobExplorerFactoryBean"**

p:dataSource-ref=**"dataSource"** p:tablePrefix=**"FOO\_"**/>

<bean id=**"jobService"** class=**"org.springframework.batch.admin.service.SimpleJobServiceFactoryBean"**>

<property name=**"jobRepository"** ref=**"jobRepository"** />

<property name=**"jobLauncher"** ref=**"jobLauncher"** />

<property name=**"jobLocator"** ref=**"jobRegistry"** />

<property name=**"dataSource"** ref=**"dataSource"** />

<property name=**"databaseType"** value=**"db2"** />

<property name=**"tablePrefix"** value=**"REPMAN.BATCH\_"** />

</bean>

<bean id=**"dataSource"** class=**"org.springframework.jndi.JndiObjectFactoryBean"**>

<property name=**"jndiName"** value=**"java:comp/env/jdbc/reportingManagerCustDbDataSource"**/>

</bean>

<jdbc:initialize-database data-source=**"dataSource"** enabled=**"false"** ignore-failures=**"DROPS"**>

<jdbc:script location=**"${batch.schema.script}"**/>

</jdbc:initialize-database>

</beans>

## Overriding the bean by placing a new context file to create a NEW datasource

**C:\software\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-sample\src\main\resources\META-INF\spring\batch\override\myer-context.xml**

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<beans xmlns=**"http://www.springframework.org/schema/beans"** xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**

xmlns:batch=**"http://www.springframework.org/schema/batch"** xmlns:p=**"http://www.springframework.org/schema/p"**

xsi:schemaLocation=**"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd**

**http://www.springframework.org/schema/batch http://www.springframework.org/schema/batch/spring-batch-2.1.xsd"**>

<batch:job-repository id=**"jobRepository"** table-prefix=**"FOO\_"**/>

<bean id=**"jobExplorer"** class=**"org.springframework.batch.core.explore.support.JobExplorerFactoryBean"**

p:dataSource-ref=**"dataSource"** p:tablePrefix=**"FOO\_"**/>

<bean id=**"jobService"** class=**"org.springframework.batch.admin.service.SimpleJobServiceFactoryBean"**>

<property name=**"jobRepository"** ref=**"jobRepository"** />

<property name=**"jobLauncher"** ref=**"jobLauncher"** />

<property name=**"jobLocator"** ref=**"jobRegistry"** />

<property name=**"dataSource"** ref=**"dataSource"** />

<property name=**"databaseType"** value=**"db2"** />

<property name=**"tablePrefix"** value=**"REPMAN.BATCH\_"** />

</bean>

</beans>

# Add new batch properties for DB2

C:\software\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-sample\src\main\resources\batch-db2.properties

# Placeholders batch.\*

# for DB2:

batch.jdbc.driver**=**com.ibm.db2.jcc.DB2Driver

batch.jdbc.url**=**jdbc:db2://XXXXXXXX:1000/CUSTDB:currentSchema=REPMAN;

batch.jdbc.user**=**db2inst1

batch.jdbc.password**=**XXXXXX

batch.schema**=**REPMAN

batch.jndi.name**=**

batch.naming.factory.initial**=**

batch.naming.provider.url**=**

batch.database.incrementer.class**=**org.springframewo rk.jdbc.support.incrementer.DB2SequenceMaxValueInc rementer

batch.lob.handler.class**=**org.springframework.jdbc.support.lob.DefaultLobHandler

batch.schema.script**=**classpath\*:/org/springframework/batch/core/schema-db2.sql

batch.drop.script**=**classpath\*:/org/springframework/batch/core/schema-drop-db2.sql

batch.business.schema.script**=**classpath:/business-schema-db2.sql

batch.data.source.init**=**false

# Bean Properties for override

# when not using sequences:

incrementerParent.columnName**=**ID

## You may need to add the db2 jar file for runtime execution

Update the top level pom.xml with the driver details

<dependency>

<groupId>**com.ibm.db2**</groupId>

<artifactId>**db2jcc\_license\_cu**</artifactId>

<version>**9.7.0.4**</version>

<scope>**runtime**</scope>

</dependency>

<dependency>

<groupId>**com.ibm.db2**</groupId>

<artifactId>**db2jcc4**</artifactId>

<version>**9.7.0.4**</version>

<scope>**runtime**</scope>

</dependency>

Download and install the dirver to maven repository

mvn install:install-file -Dfile=C:\myjavaworks\spring-batch-admin-master\db2jcc\_license\_cu-3.8.47.jar -DgroupId=com.ibm.db2 -DartifactId=db2jcc\_license\_cu -Dversion=9.7.0.4 -Dpackaging=jar -DgeneratePom=true

mvn install:install-file -Dfile=C:\myjavaworks\spring-batch-admin-master\db2jcc3.8.47.jar -DgroupId=com.ibm.db2 -DartifactId=db2jcc4 -Dversion=9.7.0.4 -Dpackaing=jar -DgeneratePom=true

## stax-api is also required while running on websphere

<dependency>

<groupId>**javax.xml.stream**</groupId>

<artifactId>**stax-api**</artifactId>

<version>**1.0-2**</version>

<scope>**runtime**</scope>

</dependency>

## If you hit java version incompatibility issues, try this

<plugin>

<groupId>**org.apache.maven.plugins**</groupId>

<artifactId>**maven-compiler-plugin**</artifactId>

<version>**3.1**</version>

<configuration>

<source>**1.5**</source>

<target>**1.5**</target>

</configuration>

</plugin>

# Deploying war file in websphere

Set the system variable for websphere .

Application servers > server1 > Process Definition > Java Virtual Machine > Custom Properties

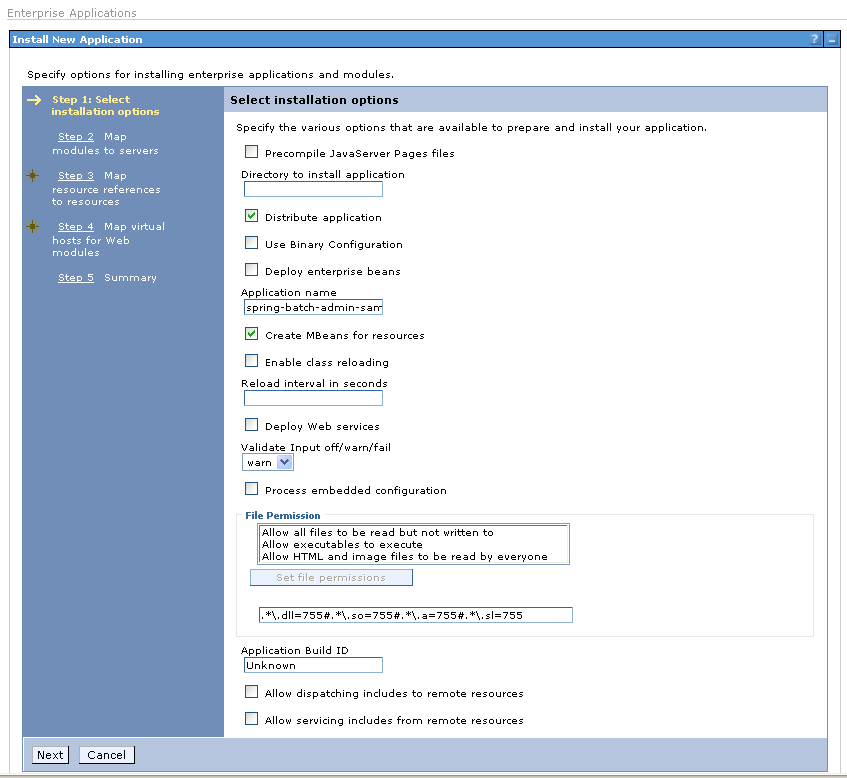
ENVIRONMENT = db2

C:\software\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-1.2.1.RELEASE\spring-batch-admin-sample\target\spring-batch-admin-sample-1.2.1.RELEASE.war

Context Root: spring\_admin\_console

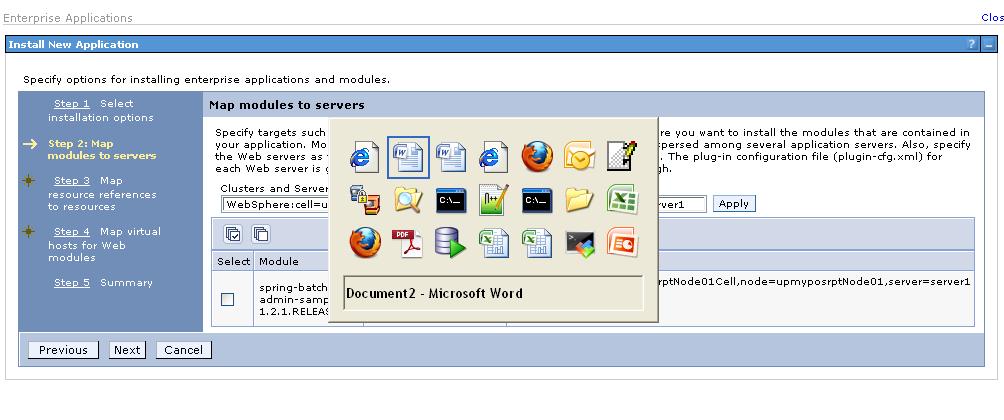
## Step 1:

No Changes



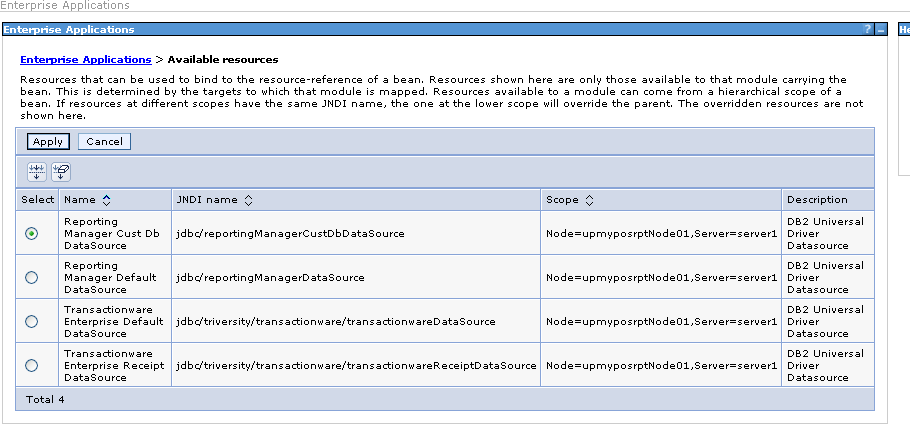
## Step 2 :

No Changes

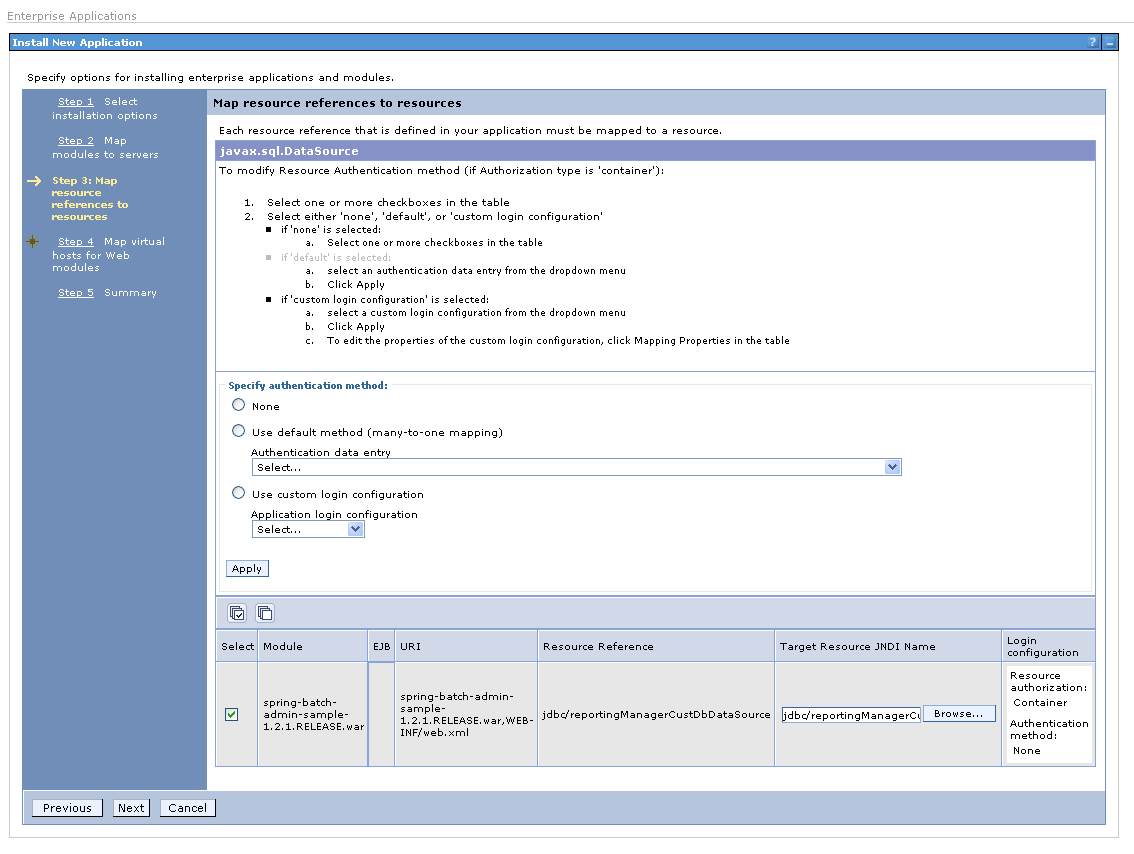


## Step 3:

Select the resource references and select the module

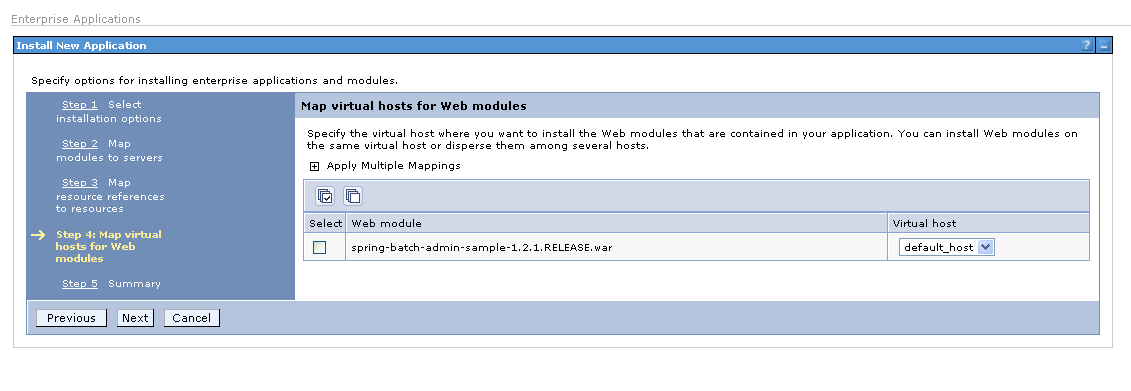


Apply

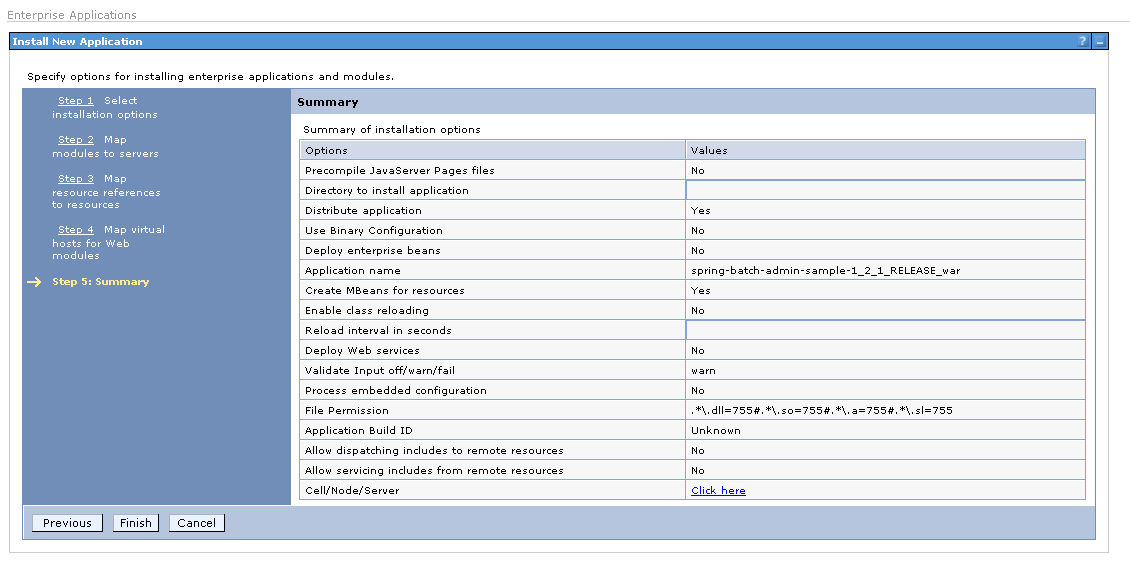


## Step4

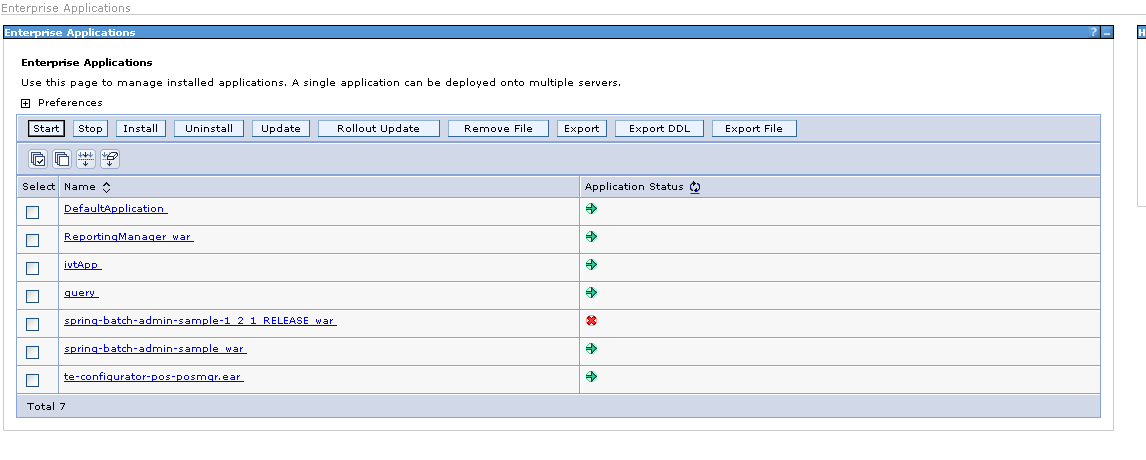
No Changes



## Finish



## Save

Select the application again for modifying the class loader priorities



## Select class loading and update detection

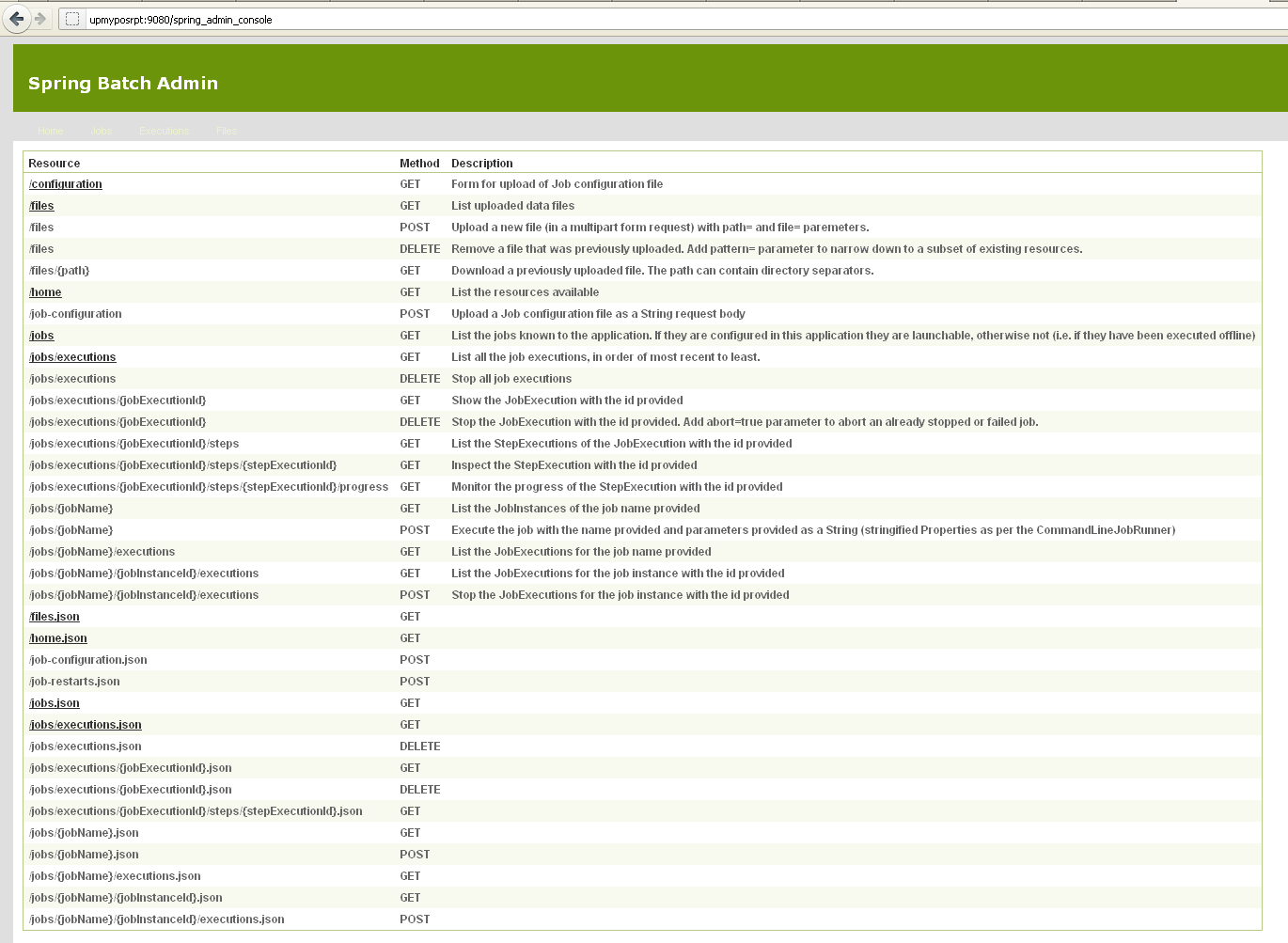


Choose >Class loaded with application class loader first

And Save the Configuration

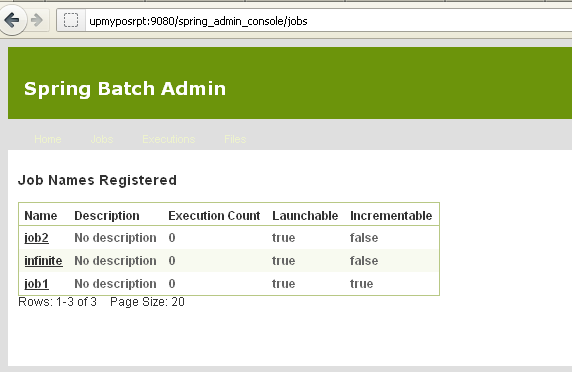
# Wait for the application start process and check the application url

<http://upmyposrpt:9080/spring_admin_console>



## Check your Jobs

<http://upmyposrpt:9080/spring_admin_console/jobs>



## Check your executions

<http://upmyposrpt:9080/spring_admin_console/jobs/executions>