**How can you find in which JARs/libraries some specific classes are?**

<http://www.findjar.com/index.x>

achine generated alternative text:
findJAR 
@com 
org.apache.hadoop.mapred.JobConf 
JAR/Class Search 
lueries: SSLSessionManager org.springframework.jdbc.core.simple org.springframework.core.annotation.AnnotationUtils 
findJAR.com is a JAR search engine that helps Java developers to find 
containing required 
JAR libraries 
Java classes. 

You see a few options - pick the first (we don't care about JSPs):

achine generated alternative text:
findJAR 
.com 
org.apache.hadoop.mapred.JobConf 
JAR/Class Search 
haus.jackson.annotate.JsonClass org.apache.axis.types.UnsignedLong org.springframework.jndi.Jr 
Search results for: 
org.apache.hadoop.mapred.JobConf 
LCLASS) org.apache.hadoop.mapred.JobConf 
LCL-ASS) org.apache.hadoop.mapred.jobconf_jsp 
LCL-ASS) org.apache.hadoop.mapred.JobConfigurable 
LCLASS) org.apache.hadoop.mapred.jobconf_005fhistory_jsp 

You see a few possibilities:

achine generated alternative text:
(MAVEN2J hadoop- 
LMAVEN2J hadoop- 
LMAVEN2J hadoop- 
LMAVEN2J hadoop- 
LMAVEN2J hadoop- 
findJAR 
.com 
JAR/Class Search 
1-8.2-504.jdbc4 AbstractBaseLayoutManager org.jboss.wsf.framework.management.ManagedEndpoi 
Information on class org.apache.hadoop.mapred.JobConf: 
Containing JAR files: 
core- 
core- 
core- 
core- 
core- 
o. 
o. 
o. 
o. 
o. 
19.0.jar 
19.1.jar 
19.1-hudson-l .jar 
19.1-hudson-2.jar 
19.1-hudson-3.jar 

It gives you an idea what to look for - most likely something like hadoop-core-XXX.jar

**Viewing jar content:**

In Eclipse - you can see Library and jar content:

achine generated alternative text:
MRv2_JDK8 
test 
*JRE System Library 
[Java SE 8 [1.8.0_25]] 
v *CDH5.5 mrv2 full 
hadoop-common-2.6.O-cdh5.5.1 .jar - /Users/marinag 
hadoop-mapreduce-cl ient-core-2.6.O-cdh5.5.1 .jar 
commons-cli-l .2.jar - /Users/marinapopova/Marinar 
commons-logging-I .1 .3.jar - /Users/marinapopova/N 
guava-11.0.2.jar - /Users/marinapopova/Marina/Tool! 
commons-collections-3.2.2.jar - /Users/marinapop0'e 
activation-I . 1 .jar - /Users/marinapopova/Marina/TooI 
apacheds-il 8n-2.O.O-M1 5.jar - /Users/marinapopova 
apacheds-kerberos-codec-2.O.O-M15.jar - /LJsers/mæ 
apl-asnl -api-l .0.O-M20.jar - /Users/marinapopova/B.' 
api-util-l .0.O-M20.jar - /Users/marinapopova/Marina 
asm-3.2.jar - /Users/marinapopova/Marina/Tools/hac 
avro-l .7.6-cdh5.5.1 .jar - /Users/marinapopova/Marir 
commons-beanutils-l .7.0.jar - [Users/marinapopova 
commons-beanutils-core-l .8.O.jar - /Users/marinapo 

achine generated alternative text:
MRv2_JDK8 
test 
*JRE System Library 
[Java SE 8 [1.8.0_25]] 
V *CDH5.5 mrv2 full 
hadoop-common-2.6.O-cdh5.5.1 .jar - 
org.apache.hadoop 
60 HadooplllegalArgumentException.class 
org.apache.hadoop.conf 
org.apache.hadoop.crypto 
org.apache.hadoop.crypto.key 
org.apache.hadoop.crypto.key.kms 
org.apache.hadoop.crypto.random 

**Java regular JAR vs Runnable JAR**

The runnable jar contains a **MANIFEST.MF** file, which defines the Main class to be executed when the jar is run.

Non-runnable jars are just libraries of classes, that can be added to the classpath so that code is reused (it also contains the manifest file, but no main class there)

If you are curios - unjar one of the JARs you've created and take a look at the META-INF/MANIFEST.MF file they contain