*name* := **"Loader"***version* := **"0.1"***scalaVersion* := **"2.11.8"**

*// to include repository  
resolvers* += **"RSI"** at **"http://localhost:8081/repository/SAMPLE\_REL/"***libraryDependencies* ++= *Seq*(  
 **"org.apache.spark"** %% **"spark-core"** % **"2.1.0"** % **"provided"**,  
 **"org.apache.spark"** %% **"spark-sql"** % **"2.1.0"** % **"provided"**,  
 **"org.apache.spark"** %% **"spark-hive"** % **"2.1.0"** %**"test"**,  
 **"org.json4s"** %% **"json4s-jackson"** % **"3.2.11"** % **"provided"**,  
 **"com.typesafe"** % **"config"** % **"1.3.1"**,  
 **"org.apache.kafka"** % **"kafka-clients"** % **"2.0.0"**,  
 **"com.rsi.commons.utils"** % **"CommonUtils"** % **"1.0"**,  
 **"com.rsi.audit"** % **"AuditMessage"** % **"1.0"**)  
*// to include sbt project***lazy val** auditMessage = *RootProject*(*file*(**"../AuditMessage"**))  
**val** main = *Project*(id = **"loader"**, base = *file*(**"."**)).dependsOn(auditMessage)

*// to compile and test included project frombase projct  
//val main = Project(id = "loader", base = file(".")).dependsOn(auditMessage % "compile->compile;test->test")*

*test* in *assembly* := {}  
  
*assemblyOption* in *assembly* := (*assemblyOption* in *assembly*).value.copy(includeScala = **false**)  
  
*assemblyMergeStrategy* in *assembly* := {  
 **case** *PathList*(**"META-INF"**, xs @ \_\*) => *MergeStrategy*.*discard* **case** x => *MergeStrategy*.*first*}