Spark 编程模型（下）
--IDEA 搭建及开发

目 录

Spark 编程模型（下）

1 安装 IntelliJ IDEA

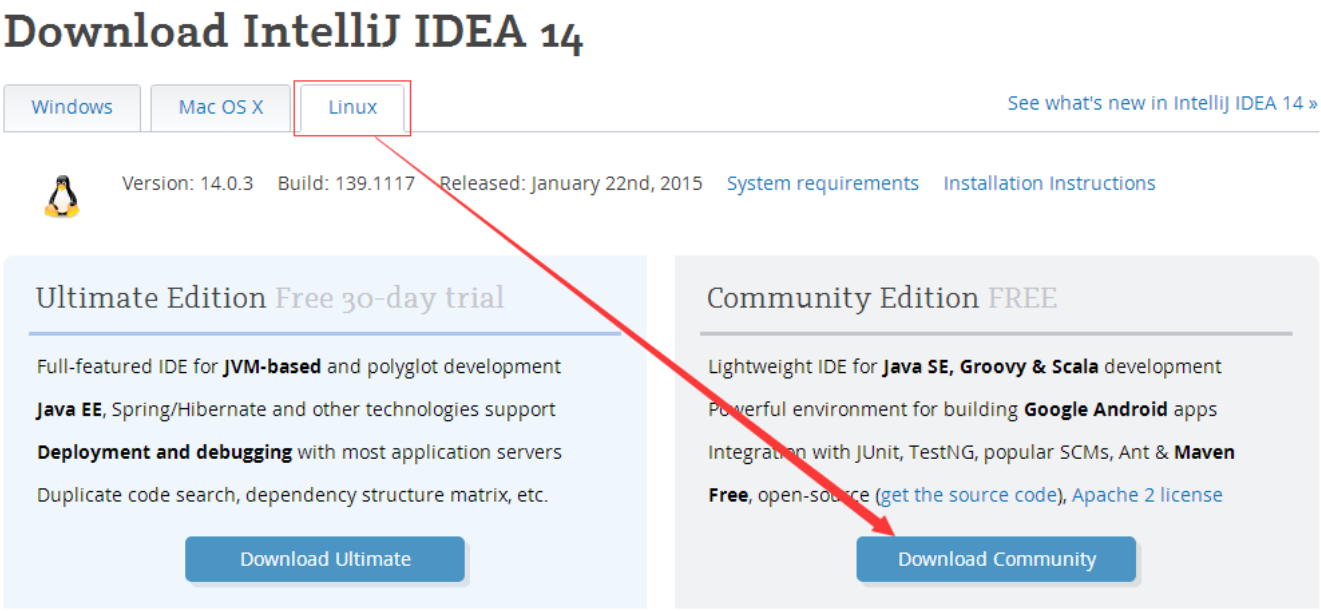
IDEA 全称 IntelliJ IDEA，是 java 语言开发的集成环境，IntelliJ 在业界被公认为最好的 java 开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、Ant、JUnit、CVS 整合、代码审查、 创新的 GUI 设计等方面的功能可以说是超常的。IDEA 是 JetBrains 公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。

IDEA 每个版本提供 Community 和 Ultimate 两个版本，如下图所示，其中 Community 是完全免费的，而 Ultimate 版本可以使用 30 天，过这段时间后需要收费。从安装后使用对比来看，下载一个 Community 版本足够了。

1.1 安装软件

1.1.1 下载 IDEA 安装文件

可以到 Jetbrains 官网 http://www.jetbrains.com/idea/download/，选择最新的安装文件。由于以后的练习需要在 Linux 开发 Scala 应用程序，选择 Linux 系统 IntelliJ IDEA14，如下图所示：



【注】在该系列配套资源的 install 目录下分别提供了 ideaIC-14.0.2.tar.gz(社区版)和 ideaIU-14.0.2.tar.gz(正式版)安装文件，对于 Scala 开发来说两个版本区别不大

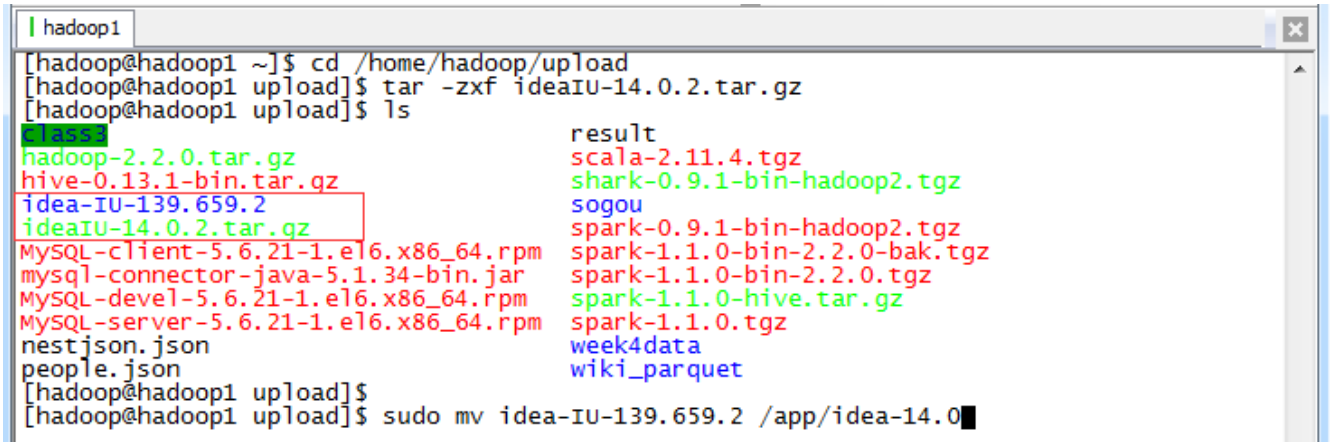
1.1.2 解压缩并移动目录

把下载的安装文件上传到目标机器，用如下命令解压缩 IntelliJ IDEA 安装文件，并迁移到/app 目录下：

cd /home/hadoop/upload

tar -zxf ideaIU-14.0.2.tar.gz

sudo mv idea-IU-139.659.2 /app/idea-IU



1.1.3 配置/etc/profile 环境变量

使用如下命令打开/etc/profile 文件：

sudo vi /etc/profile

确认 JDK 配置变量正确配置（参见第 2 节《Spark 编译与部署》中关于基础环境搭建介绍）：

export JAVA_HOME=/usr/lib/java/jdk1.7.0_55

export PATH=$PATH:$JAVA_HOME

```
export JAVA_HOME=/usr/lib/java/jdk1.7.0_55
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

export PIG_HOME=/usr/local/pig-0.13.0
export PIG_CLASSPATH=/usr/local/hadoop-1.1.2/conf
export PATH=$PATH:/usr/local/hadoop-1.1.2/bin:$PIG_HOME/bin
```

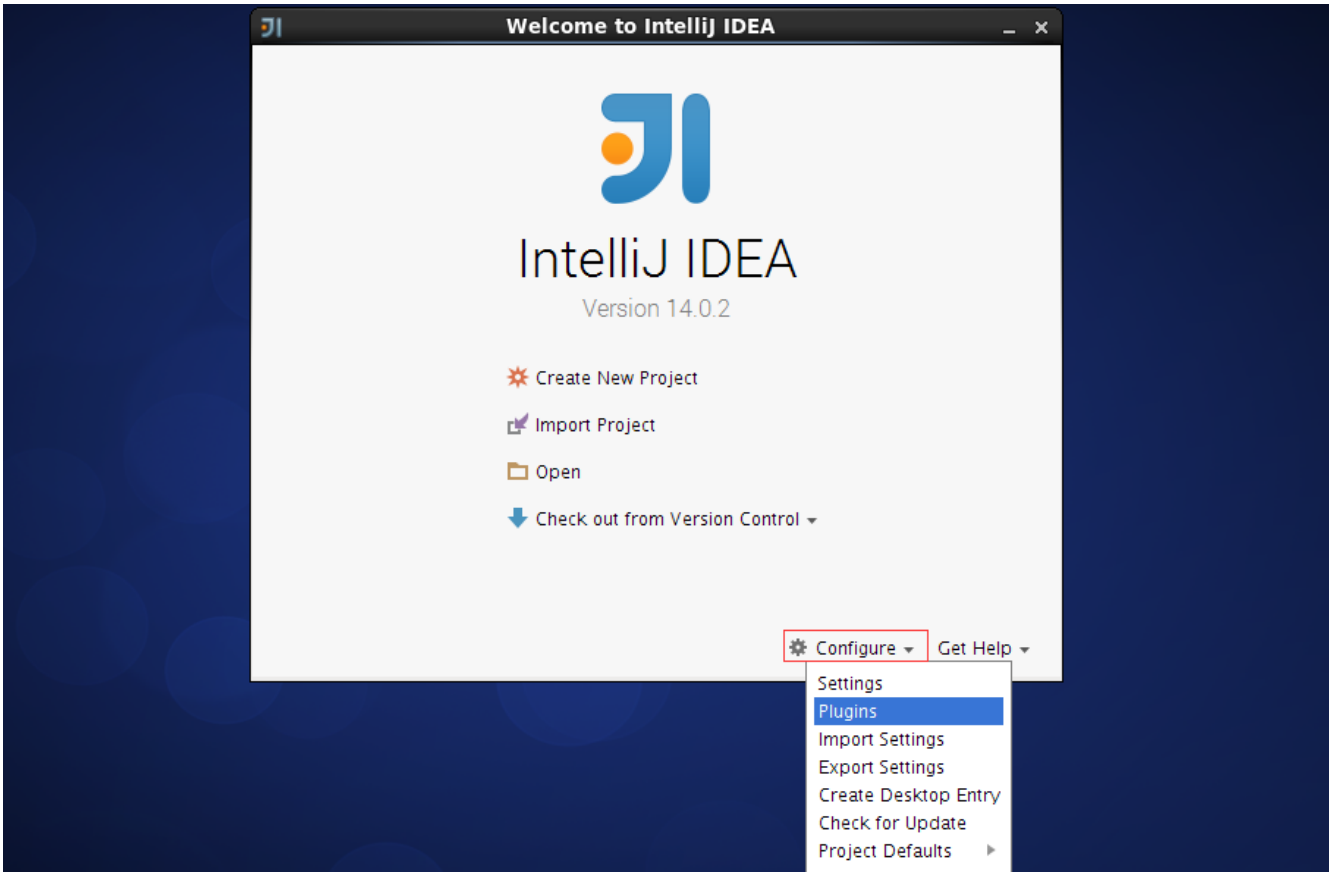
1.2 配置 Scala 环境

1.2.1 启动 IntelliJ IDEA

可以通过两种方式启动 IntelliJ IDEA：

- 到 IntelliJ IDEA 安装所在目录下，进入 bin 目录双击 idea.sh 启动 IntelliJ IDEA；
- 在命令行终端中，进入$IDEA_HOME/bin 目录，输入./idea.sh 进行启动

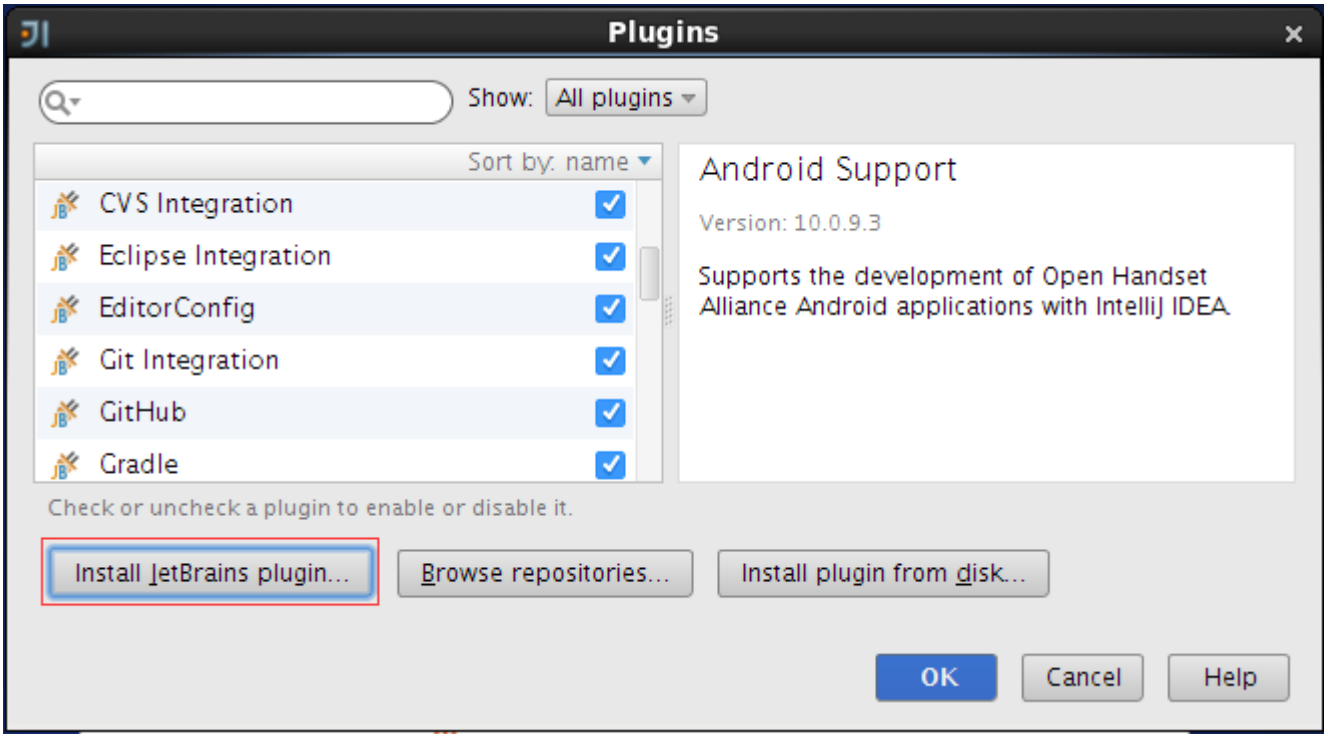
IDEA 初始启动目录如下，IDEA 默认情况下并没有安装 Scala 插件，需要手动进行安装，安装过程并不复杂，下面将演示如何进行安装。



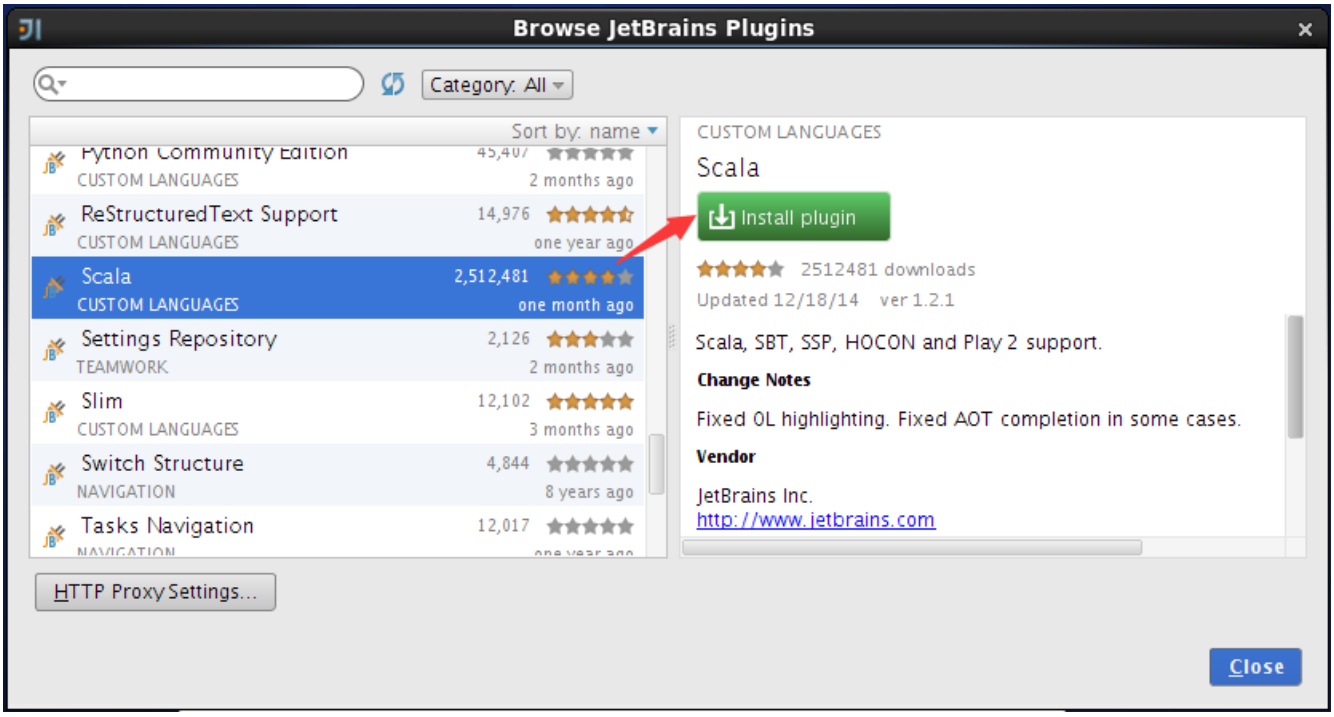
1.2.2 下载 Scala 插件

参见上图，在启动界面上选择“Configure-->Plugins"选项，然后弹出插件管理界面，在该界面上列出了所有安装好的插件，由于 Scala 插件没有安装，需要点击”Install JetBrains

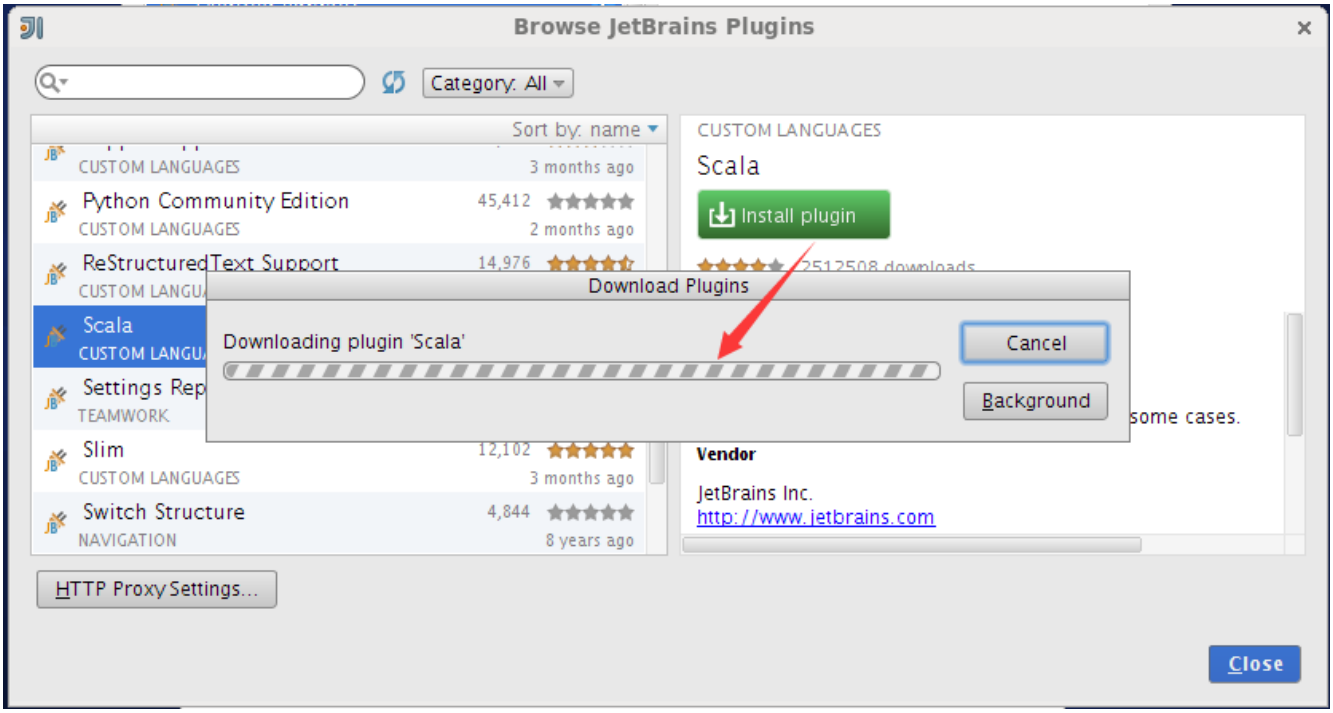
plugins"进行安装，如下图所示：



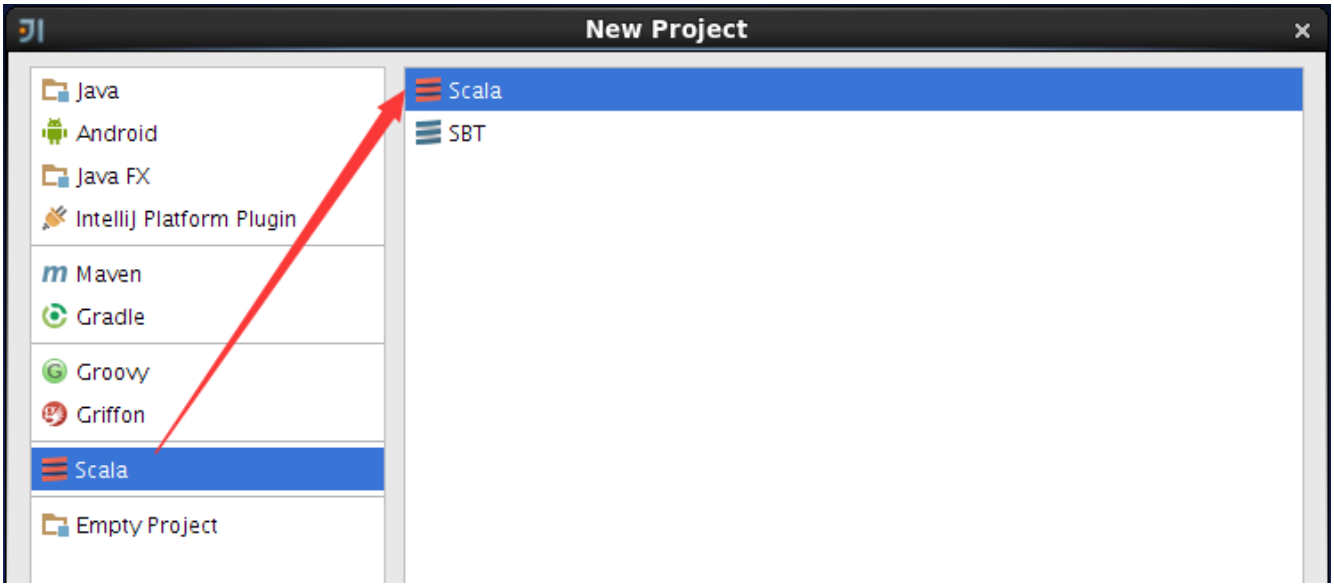
待安装的插件很多，可以通过查询或者字母顺序找到 Scala 插件，选择插件后在界面的右侧出现该插件的详细信息，点击绿色按钮"Install plugin”安装插件，如下图所示：



安装过程将出现安装进度界面，通过该界面了解插件安装进度，如下图所示：

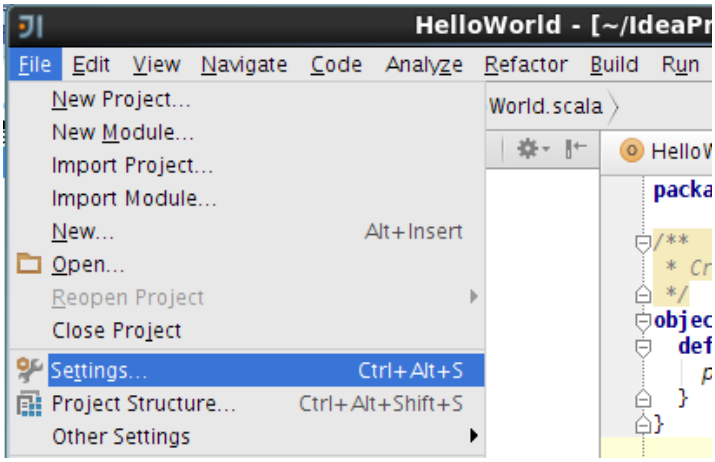


安装插件后，在启动界面中选择创建新项目，弹出的界面中将会出现"Scala"类型项目，选择后将出现提示创建的项目是仅 Scala 代码项目还是 SBT 代码项目，如下图所示：

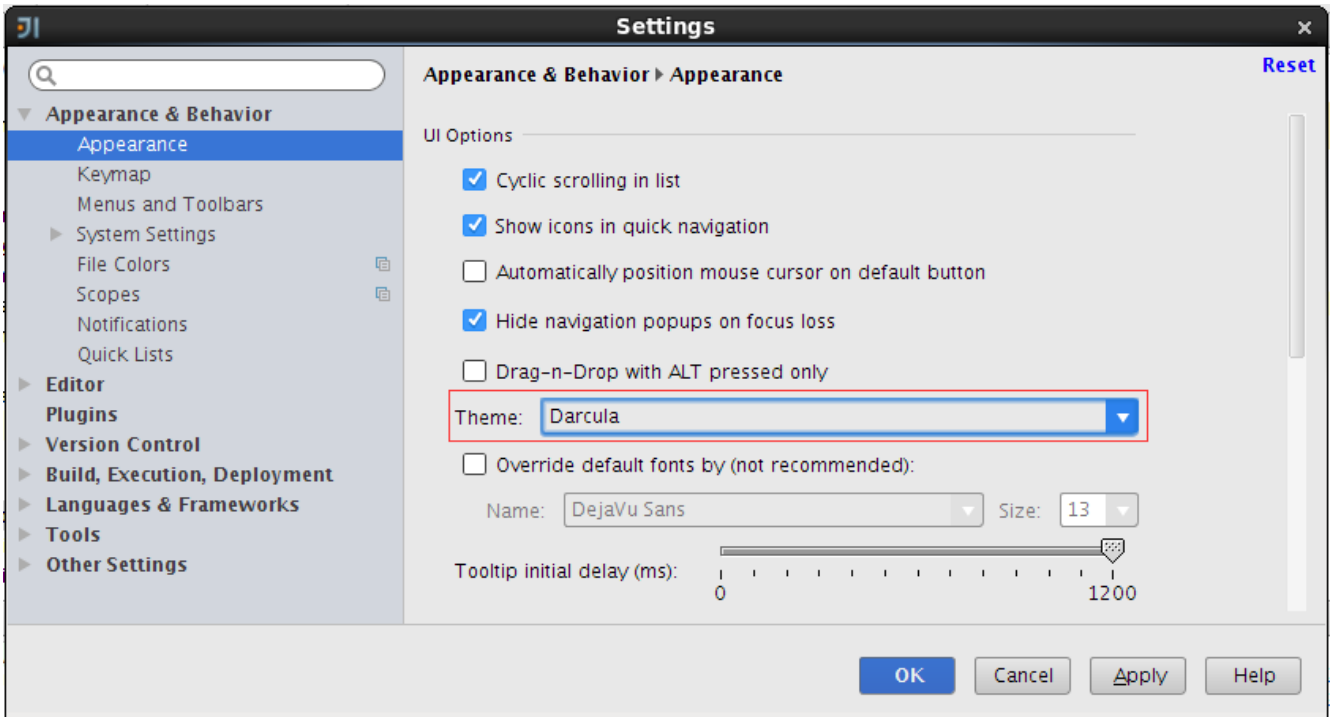


1.2.3 设置界面主题

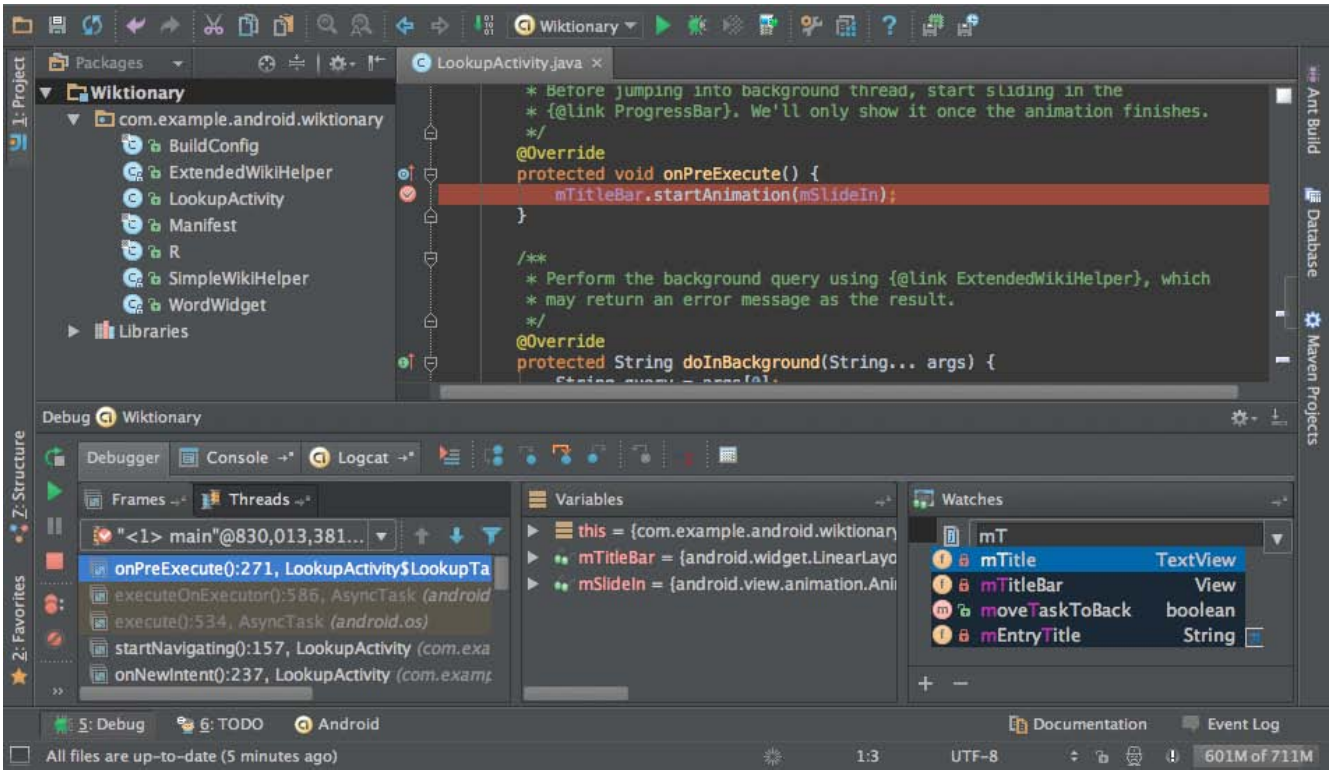
从 IntelliJ IDEA12 开始起推出了 Darcula 主题的全新用户界面，该界面以黑色为主题风格得到很多开发人员的喜爱，下面我们将介绍如何进行配置。在主界面中选择 File 菜单，然后选择 Setting 子菜单，如下图所示：



在弹出的界面中选择 Appearance &Behavior 中 Appearance，其中 Theme 中选择 Darcula 主题，如下图所示：



保存该主题重新进入，可以看到如下图样式的开发工具，是不是很酷！

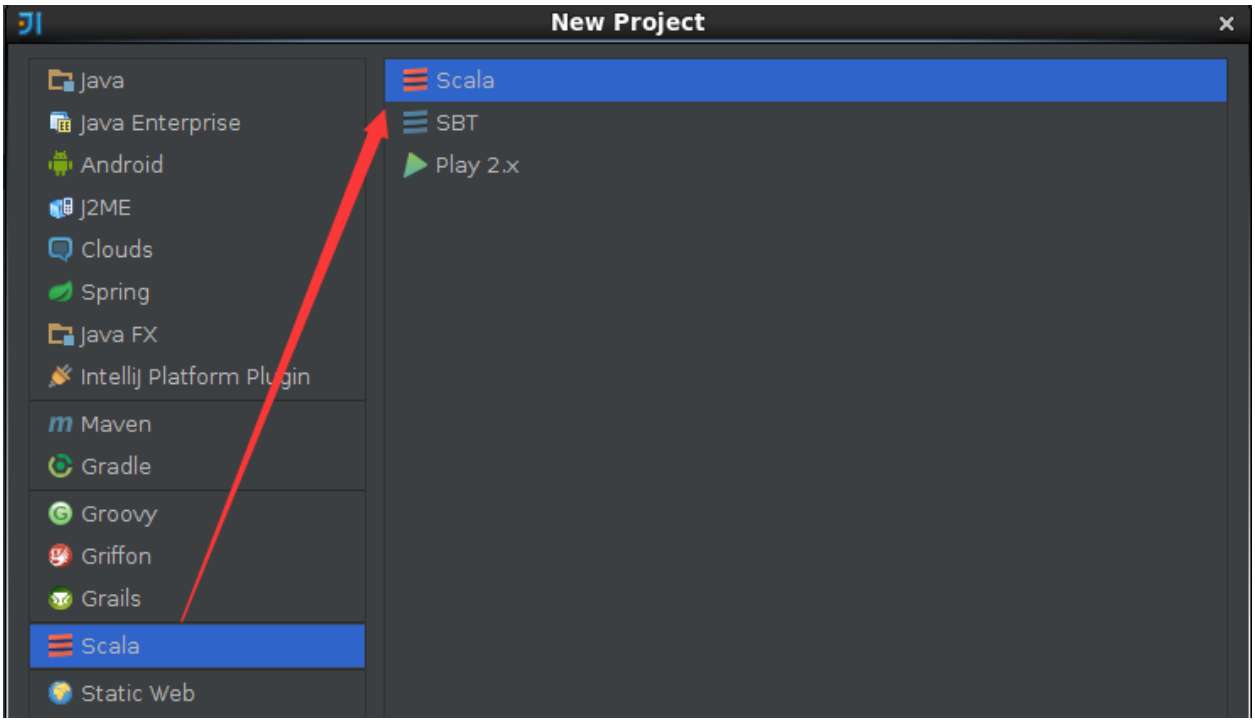


2 使用 IDEA 编写例子

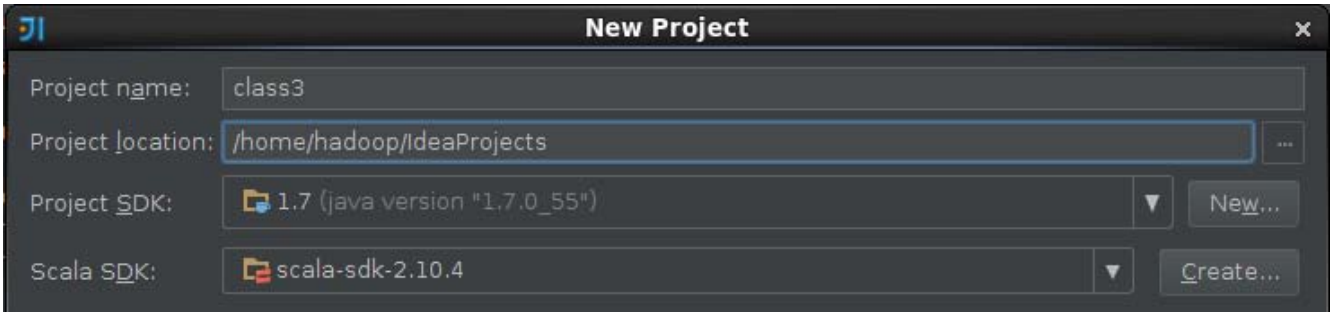
2.1 创建项目

2.1.1 设置项目基本信息

在 IDEA 菜单栏选择 File->New Project，出现如下界面，选择创建 Scala 项目：

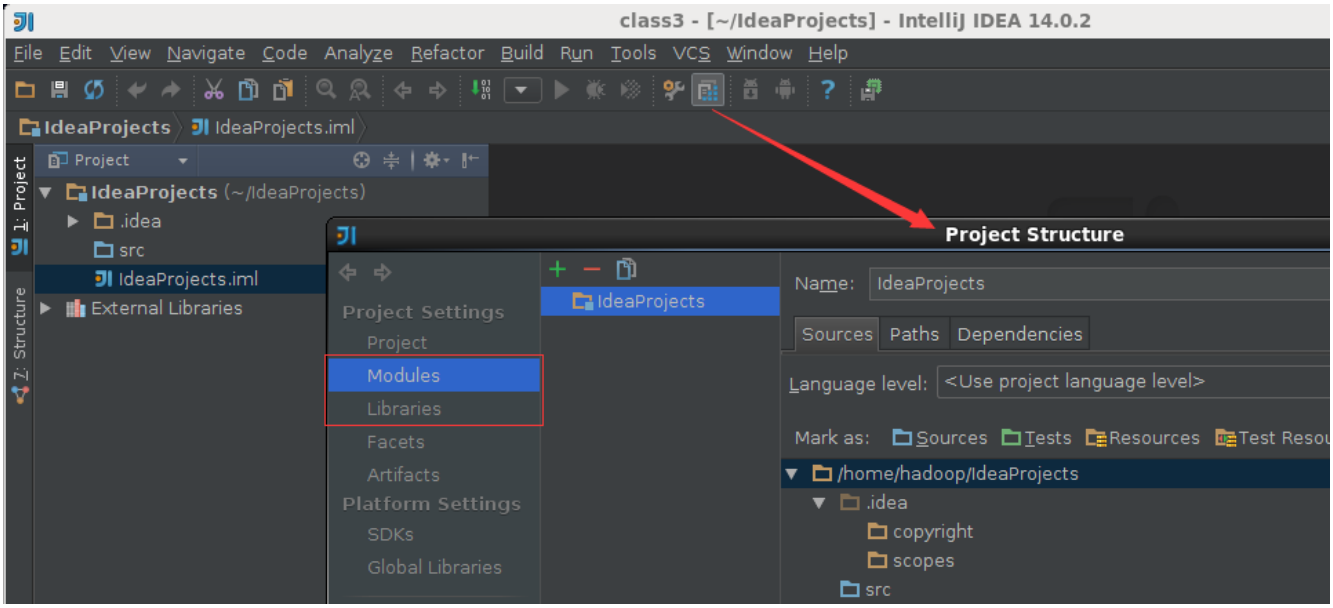


在项目的基本信息填写项目名称、项目所在位置、Project SDK 和 Scala SDK，在这里设置项目名称为 class3，关于 Scala SDK 的安装参见第 2 节《Spark 编译与部署》下 Spark 编译安装介绍：

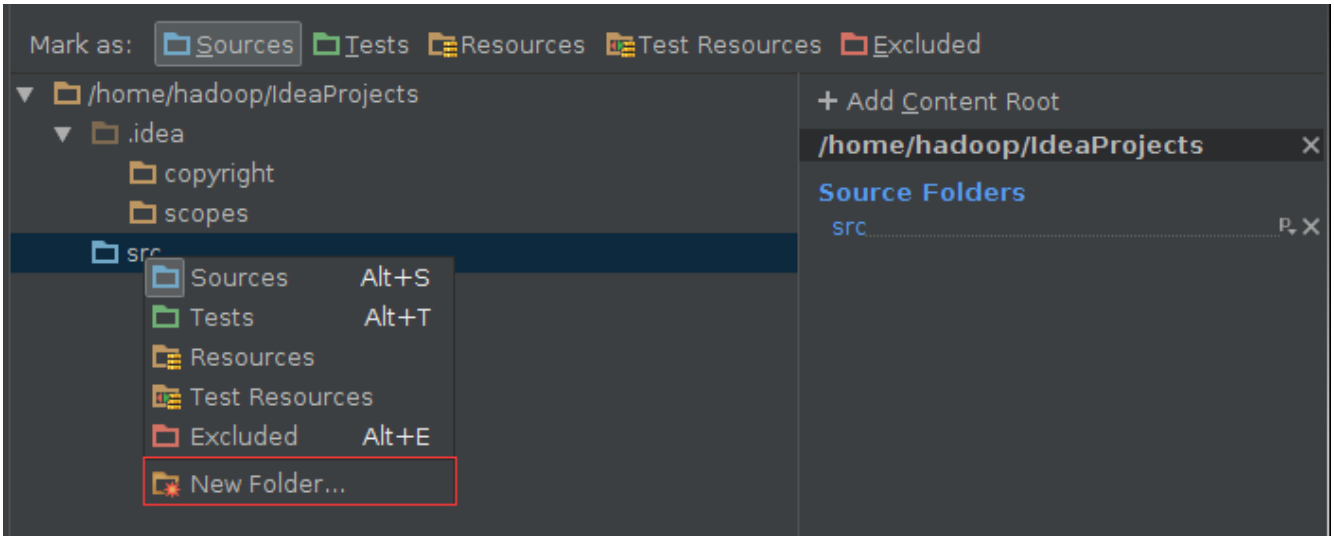


2.1.2 设置 Modules

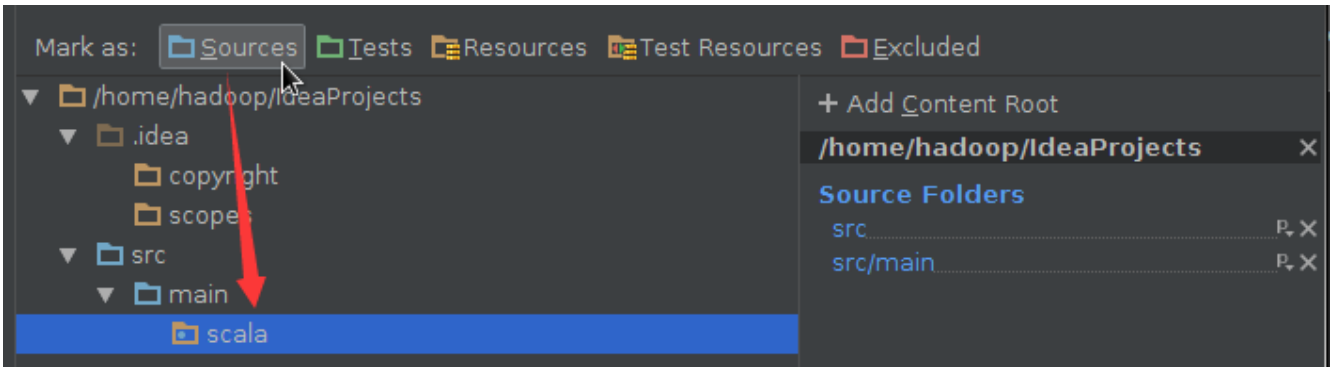
创建该项目后，可以看到现在还没有源文件，只有一个存放源文件的目录 src 以及存放工程其他信息的杂项。通过双击 src 目录或者点击菜单上的项目结构图标打开项目配置界面，如下图所示：



在 Modules 设置界面中，src 点击右键选择“新加文件夹”添加 src->main->scala 目录：

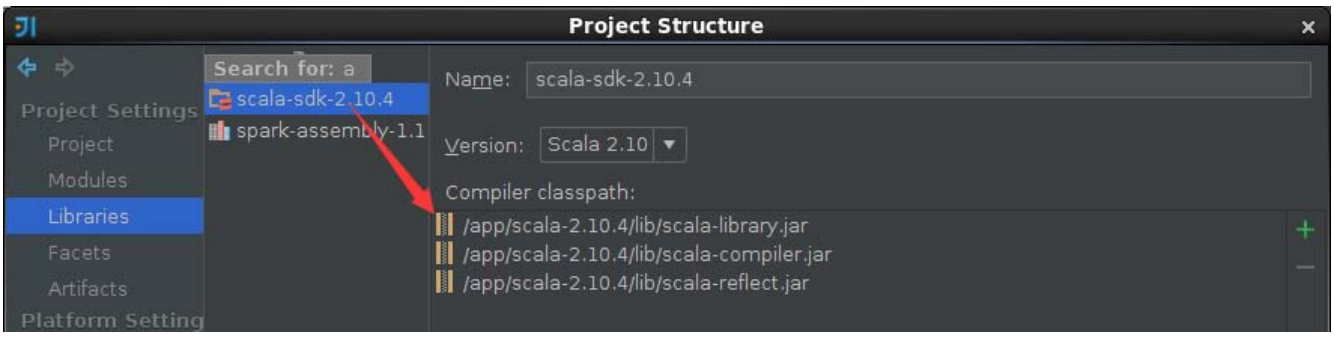


在 Modules 设置界面中，分别设置 main->scala 目录为 Sources 类型：

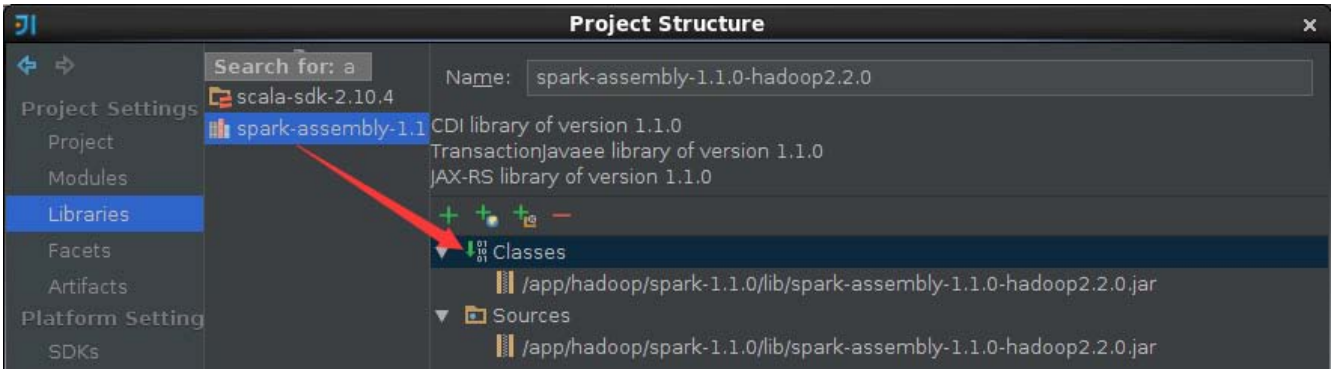


2.1.3 配置 Library

选择 Library 目录，添加 Scala SDK Library，这里选择 scala-2.10.4 版本



添加 Java Library，这里选择的是在 $SPARK_HOME/lib/spark-assembly-1.1.0-hadoop2.2.0.jar 文件，添加完成的界面如下：



2.2 例子 1：直接运行

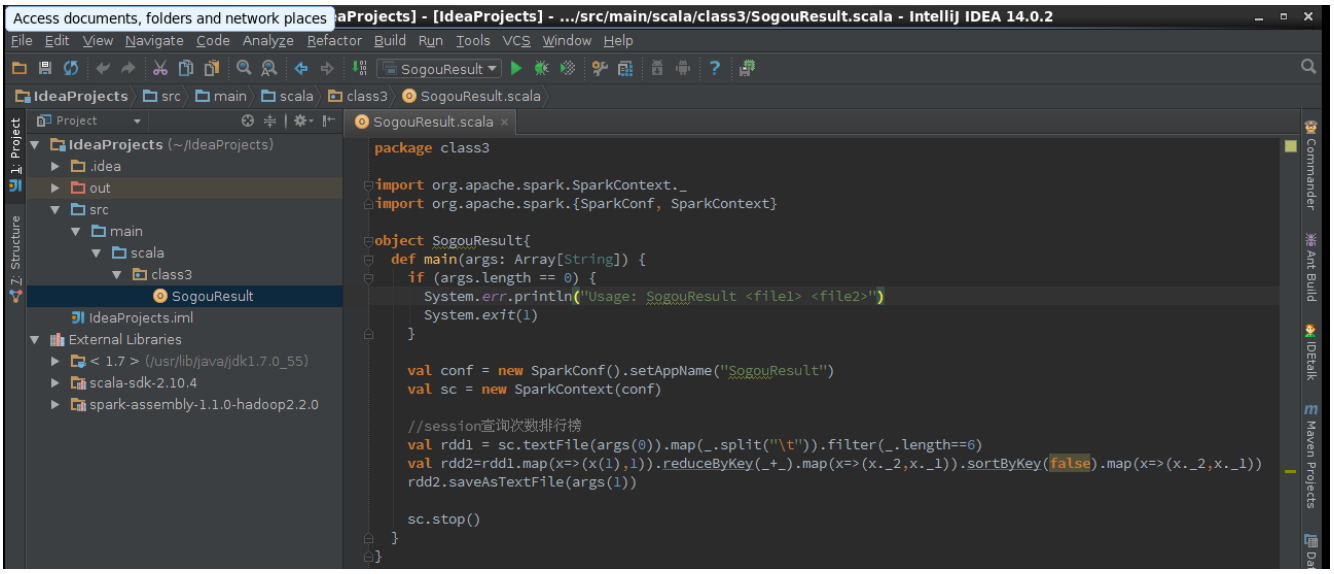
《Spark 编程模型（上）--概念及 Shell 试验》中使用 Spark-Shell 进行了搜狗日志的查询，在这里我们使用 IDEA 对 Session 查询次数排行榜进行重新练习，可以发现借助专业的开发工具可以方便快捷许多。

2.2.1 编写代码

在 src->main->scala 下创建 class3 包，在该包中添加 SogouResult 对象文件，具体代码如下：

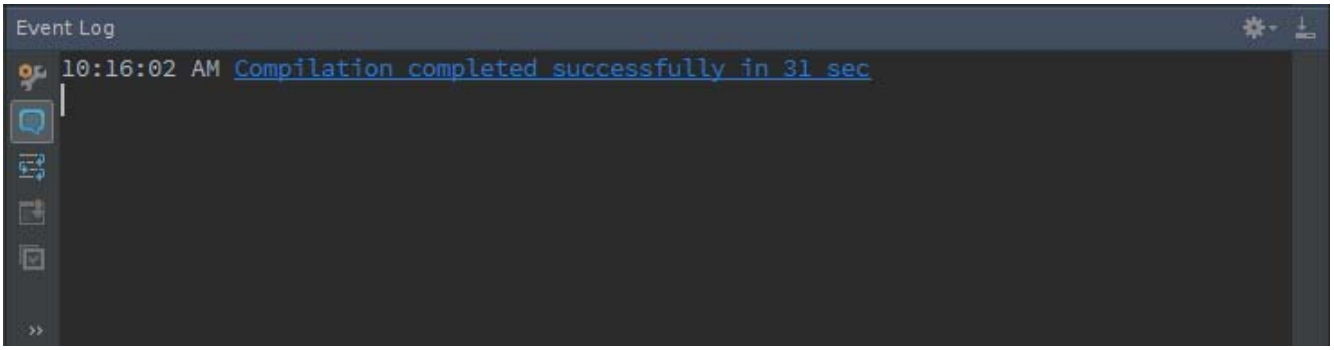
```
package class3

import org.apache.spark.SparkContext._
import org.apache.spark.{SparkConf, SparkContext}

object SogouResult{
  def main(args: Array[String]) {
    if (args.length == 0) {
      System.err.println("Usage: SogouResult <file1> <file2>")
      System.exit(1)
    }

    val conf = new SparkConf().setAppName("SogouResult").setMaster("local")
    val sc = new SparkContext(conf)

    //session 查询次数排行榜
    val rdd1 = sc.textFile(args(0)).map(_.split("\t")).filter(_.length==6)
    val
    rdd2=rdd1.map(x=>(x(1),1)).reduceByKey(_+_).map(x=>(x._2,x._1)).sortByKey(false)
    .map(x=>(x._2,x._1))
```

```
    rdd2.saveAsTextFile(args(1))
    sc.stop()
  }
}
```



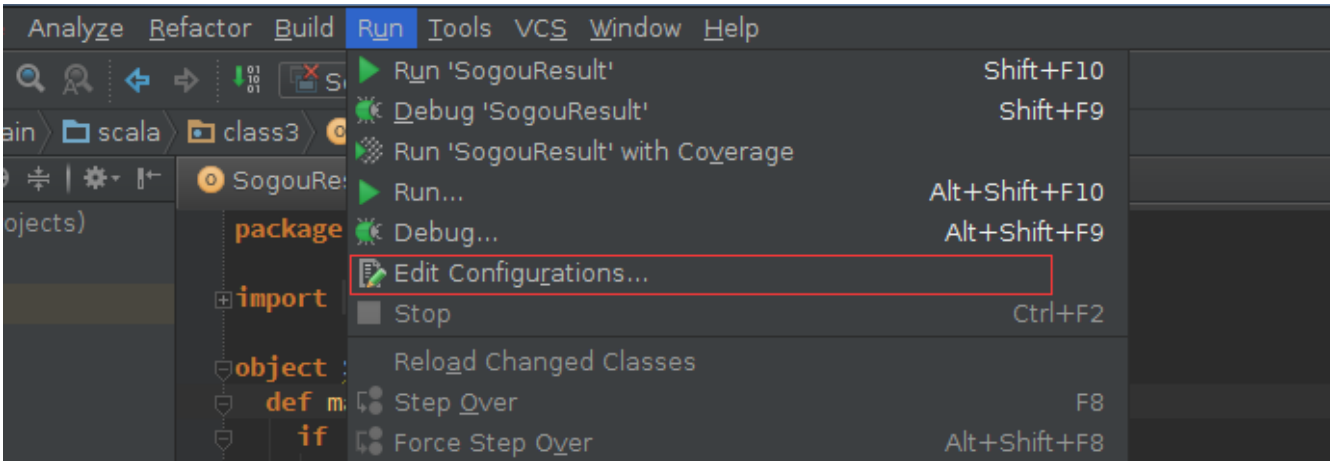
2.2.2 编译代码

代码在运行之前需要进行编译，可以点击菜单 Build->Make Project 或者 Ctrl+F9 对代码进行编译，编译结果会在 Event Log 进行提示，如果出现异常可以根据提示进行修改



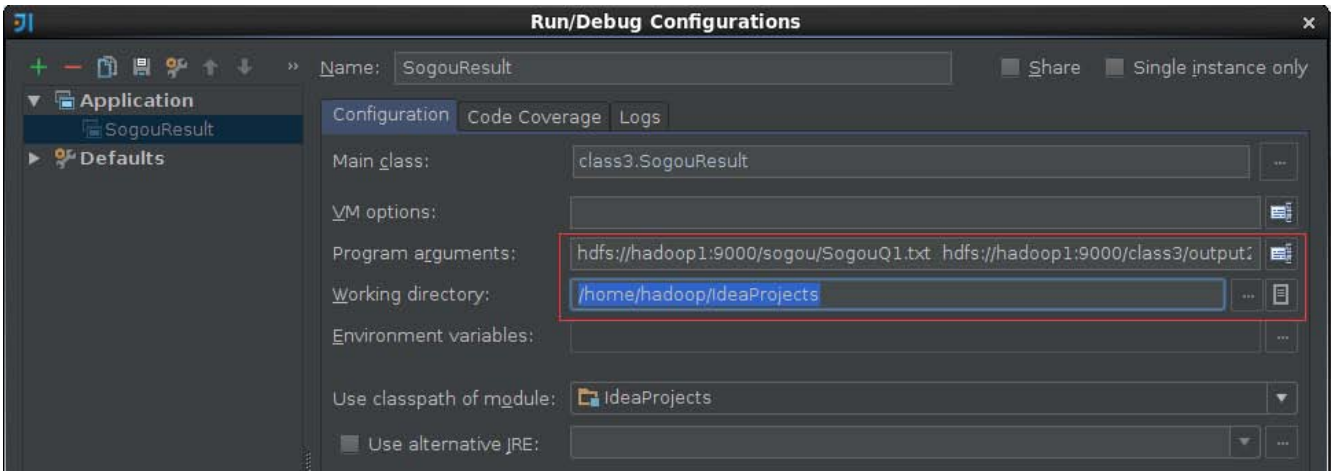
2.2.3 运行环境配置

SogouResult 首次运行或点击菜单 Run->Edit Configurations 打开"运行/调试 配置界面"



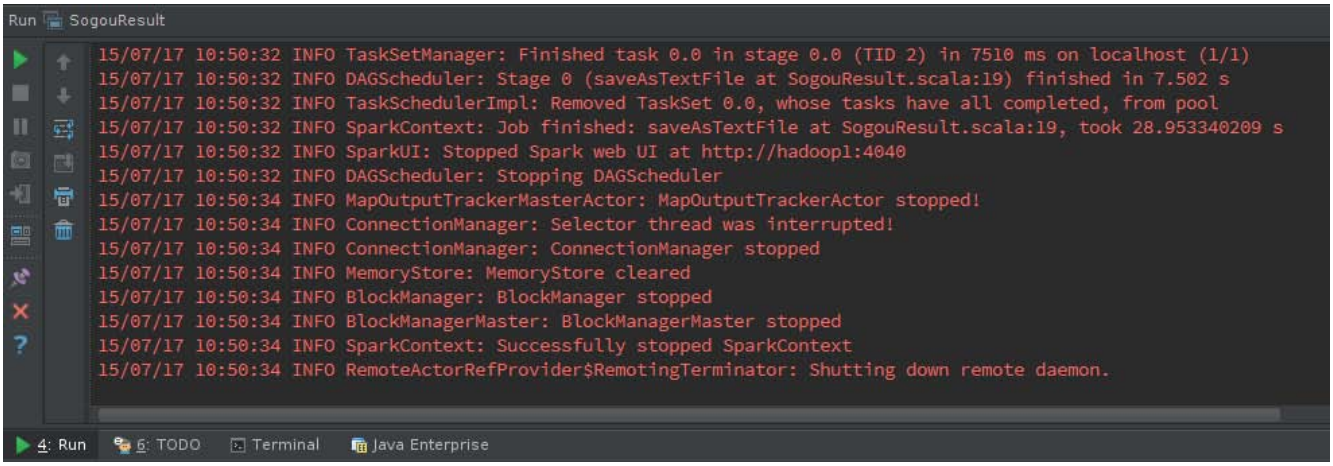
运行 SogouResult 时需要输入搜狗日志文件路径和输出结果路径两个参数，需要注意的是 HDFS 的路径参数路径需要全路径，否则运行会报错：

- 搜狗日志文件路径：使用上节上传的搜狗查询日志文件 hdfs://hadoop1:9000/sogou/SogouQ1.txt
- 输出结果路径：hdfs://hadoop1:9000/class3/output2



2.2.4 运行结果查看

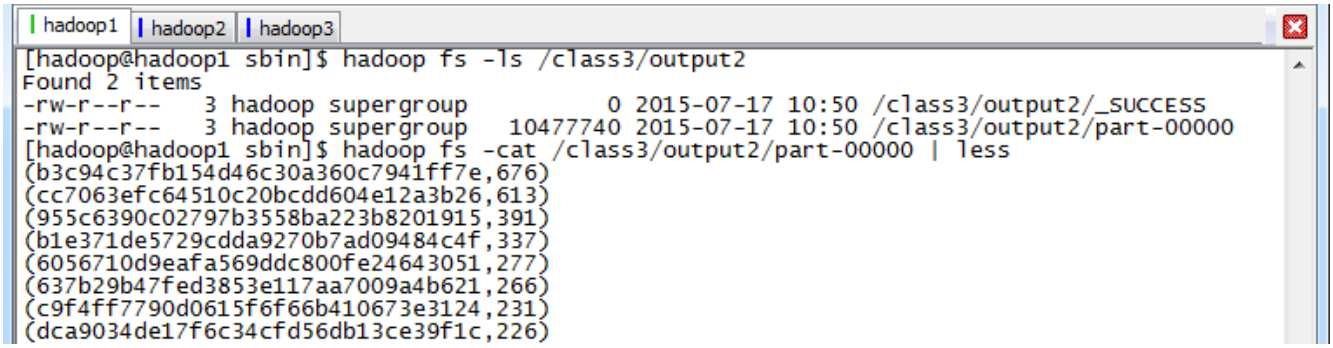
启动 Spark 集群，点击菜单 Run->Run 或者 Shift+F10 运行 SogouResult，在运行结果窗口可以运行情况。当然了如果需要观察程序运行的详细过程，可以加入断点，使用调试模式根据程序运行过程。



使用如下命令查看运行结果，该结果和上节运行的结果一致

hadoop fs -ls /class3/output2

hadoop fs -cat /class3/output2/part-00000 | less



2.3 例子 2：打包运行

上个例子使用了 IDEA 直接运行结果，在该例子中将使用 IDEA 打包程序进行执行

2.3.1 编写代码

在 class3 包中添加 Join 对象文件，具体代码如下：

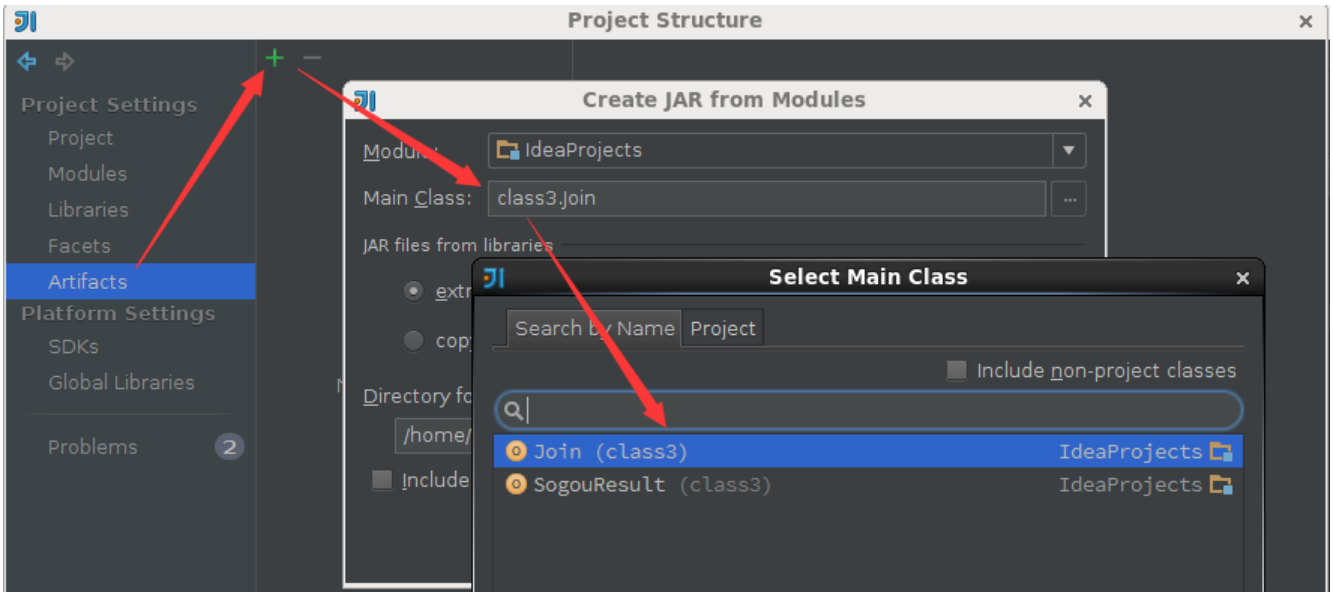
```
package class3

import org.apache.spark.SparkContext._
import org.apache.spark.{SparkConf, SparkContext}

object Join{
  def main(args: Array[String]) {
    if (args.length == 0) {
      System.err.println("Usage: Join <file1> <file2> ")
      System.exit(1)
    }
```

```
    val conf = new SparkConf().setAppName("Join").setMaster("local")
    val sc = new SparkContext(conf)

    val format = new java.text.SimpleDateFormat("yyyy-MM-dd")
    case class Register (d: java.util.Date, uuid: String, cust_id: String, lat: Float,lng: Float)
    case class Click (d: java.util.Date, uuid: String, landing_page: Int)
    val reg = sc.textFile(args(0)).map(_.split("\t")).map(r => (r(1), Register(format.parse(r(0)), r(1),
   r(2), r(3).toFloat, r(4).toFloat)))
    val clk = sc.textFile(args(1)).map(_.split("\t")).map(c => (c(1), Click(format.parse(c(0)), c(1),
    c(2).trim.toInt)))
    reg.join(clk).take(2).foreach(println)

    sc.stop()
  }
}
```

2.3.2 生成打包文件

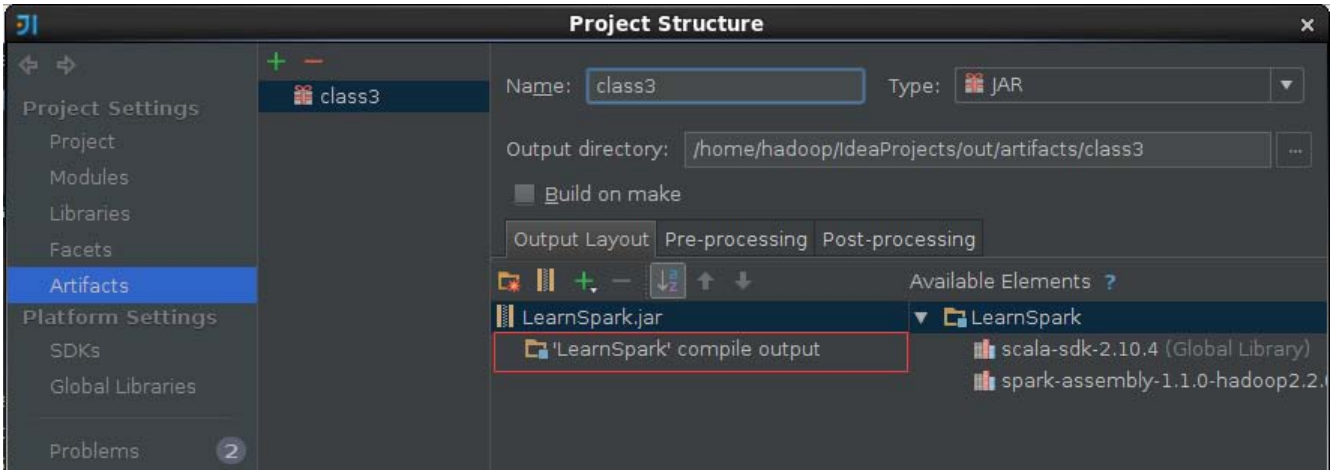
第一步 配置打包信息

在项目结构界面中选择"Artifacts"，在右边操作界面选择绿色"+"号，选择添加 JAR 包的"From modules with dependencies"方式，出现如下界面，在该界面中选择主函数入口为 Join：



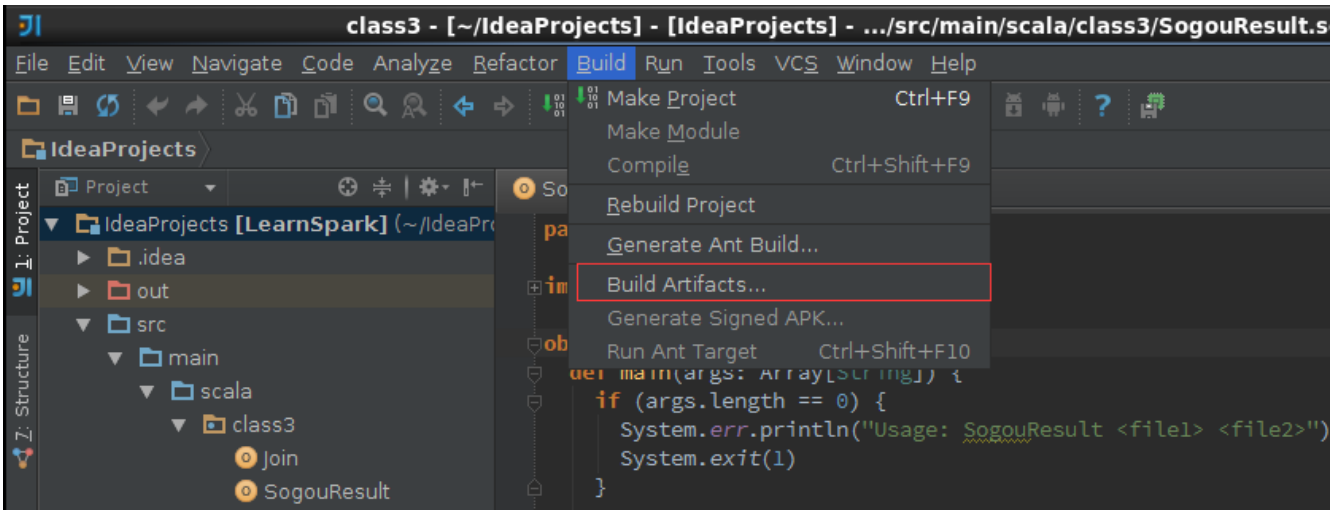
第二步 填写该 JAR 包名称和调整输出内容

【注意】的是默认情况下"Output Layout"会附带 Scala 相关的类包，由于运行环境已经有 Scala 相关类包，所以在这里去除这些包只保留项目的输出内容



第三步 输出打包文件

点击菜单 Build->Build Artifacts，弹出选择动作，选择 Build 或者 Rebuild 动作

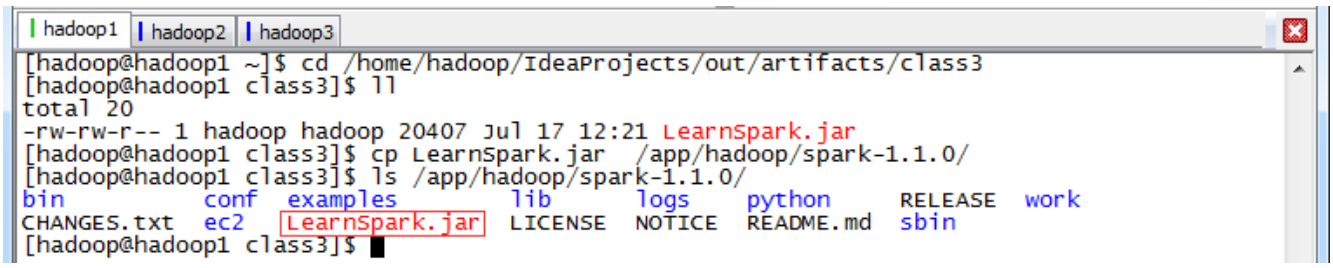


第四步 复制打包文件到 Spark 根目录下

cd /home/hadoop/IdeaProjects/out/artifacts/class3

cp LearnSpark.jar /app/hadoop/spark-1.1.0/

ls /app/hadoop/spark-1.1.0/



2.3.3 运行查看结果

通过如下命令调用打包中的 Join 方法，运行结果如下：

cd /app/hadoop/spark-1.1.0

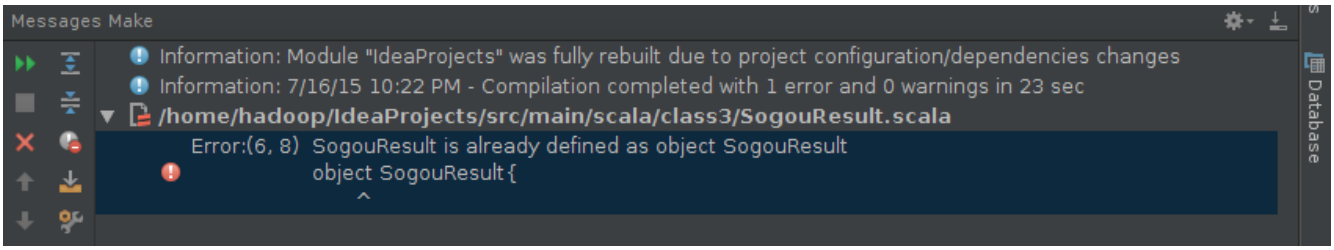
bin/spark-submit --master spark://hadoop1:7077 --class class3.Join --executor-memory 1g LearnSpark.jar hdfs://hadoop1:9000/class3/join/reg.tsv hdfs://hadoop1:9000/class3/join/clk.tsv

```
15/07/17 12:43:48 INFO Executor: Finished task 0.0 in stage 0.0 (TID 2). 1384 bytes result sent to driver
15/07/17 12:43:48 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 2) in 126 ms on localhost (1/1)
15/07/17 12:43:48 INFO DAGScheduler: Stage 0 (take at Join.scala:21) finished in 0.129 s
15/07/17 12:43:48 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
15/07/17 12:43:48 INFO SparkContext: Job finished: take at Join.scala:21, took 1.321228527 s
(81da510acc4111e387f3600308919594,(Register(Tue Mar 04 00:00:00 CST 2014,81da510acc4111e387f3600308919594,2,33.85701,-117.85574)
,Click(Thu Mar 06 00:00:00 CST 2014,81da510acc4111e387f3600308919594,61)))
(15dfb8e6cc4111e3a5bb600308919594,(Register(Sun Mar 02 00:00:00 CST 2014,15dfb8e6cc4111e3a5bb600308919594,1,33.659943,-117.95812
),Click(Tue Mar 04 00:00:00 CST 2014,15dfb8e6cc4111e3a5bb600308919594,11)))
15/07/17 12:43:48 INFO SparkUI: Stopped Spark web UI at http://hadoop1:4040
15/07/17 12:43:48 INFO DAGScheduler: Stopping DAGScheduler
15/07/17 12:43:49 INFO MapOutputTrackerMasterActor: MapOutputTrackerActor stopped!
15/07/17 12:43:49 INFO ConnectionManager: Selector thread was interrupted!
15/07/17 12:43:49 INFO ConnectionManager: ConnectionManager stopped
15/07/17 12:43:49 INFO MemoryStore: MemoryStore cleared
15/07/17 12:43:49 INFO BlockManager: BlockManager stopped
15/07/17 12:43:49 INFO BlockManagerMaster: BlockManagerMaster stopped
15/07/17 12:43:49 INFO SparkContext: Successfully stopped SparkContext
15/07/17 12:43:49 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
15/07/17 12:43:49 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote trans
ports.
15/07/17 12:43:49 INFO Remoting: Remoting shut down
15/07/17 12:43:49 INFO RemoteActorRefProvider$RemotingTerminator: Remoting shut down.
[hadoop@hadoop1 spark-1.1.0]$
```

3 问题解决

3.1 出现"*** is already defined as object ***"错误

编写好 SogouResult 后进行编译，出现"Sogou is already as object SogouResult"的错误，



出现这个错误很可能不是程序代码的问题，很可能是使用 Scala JDK 版本问题，作者在使用 scala-2.11.4 遇到该问题，换成 scala-2.10.4 后重新编译该问题得到解决，需要检查两个地方配置：Libraries 和 Global Libraries 分别修改为 scala-2.10.4