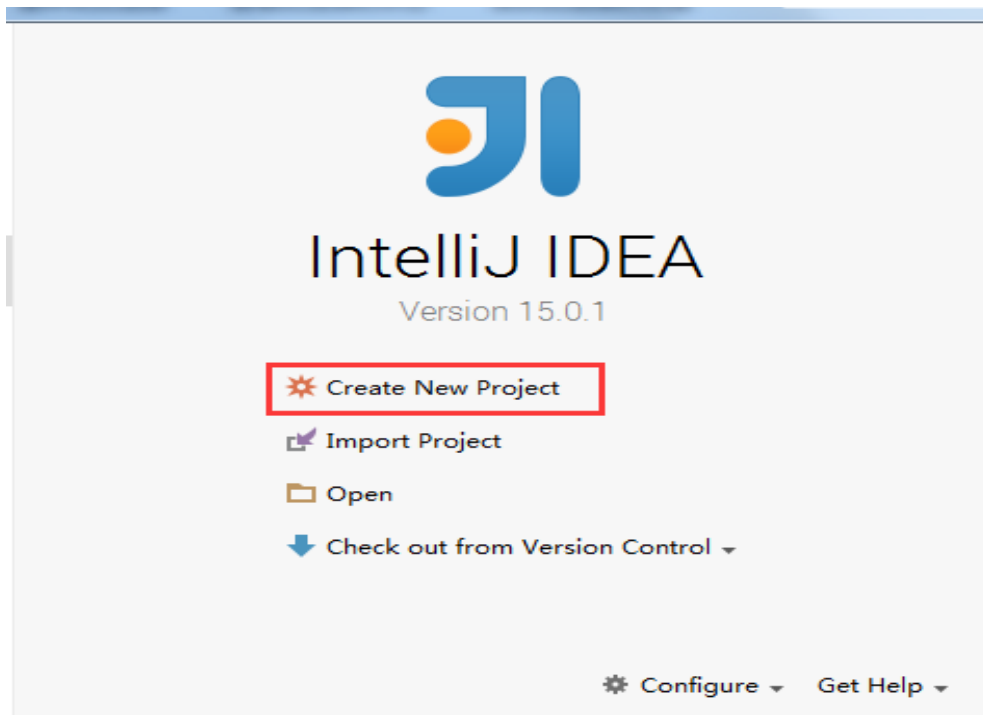


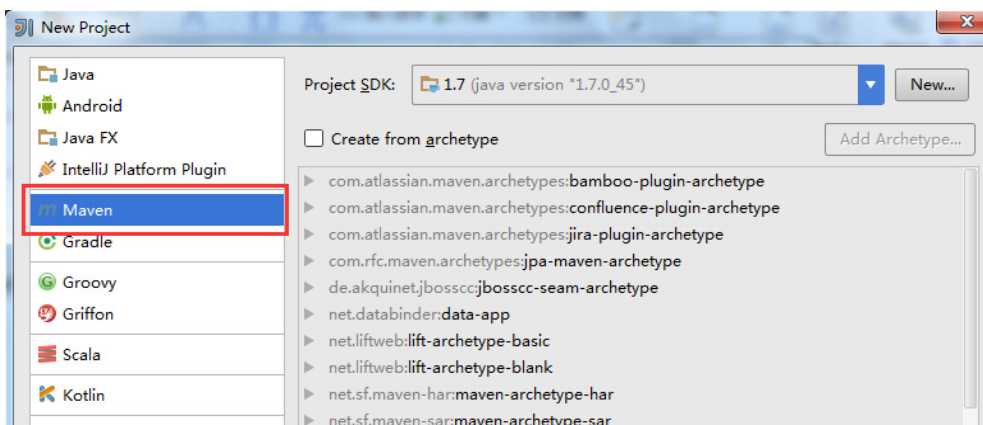
1. 在 IDEA 中编写 WordCount 程序

spark shell 仅在测试和验证程序时使用较多，而生产环境通常会在 IDEA 中编写程序，然后打成 jar 包提交到集群运行。最常用的是创建一个 Maven 项目，利用 Maven 来管理 jar 包的依赖。

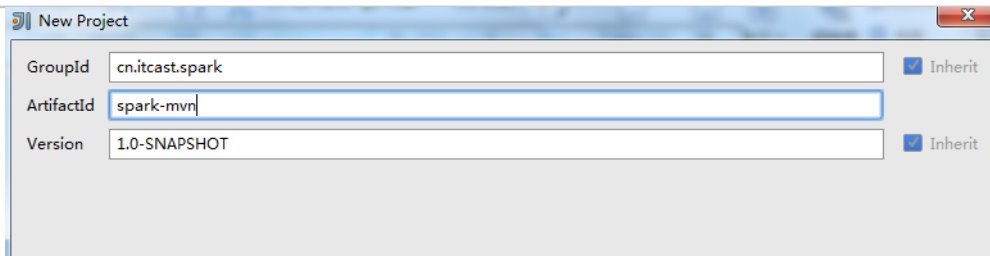
1. 创建项目



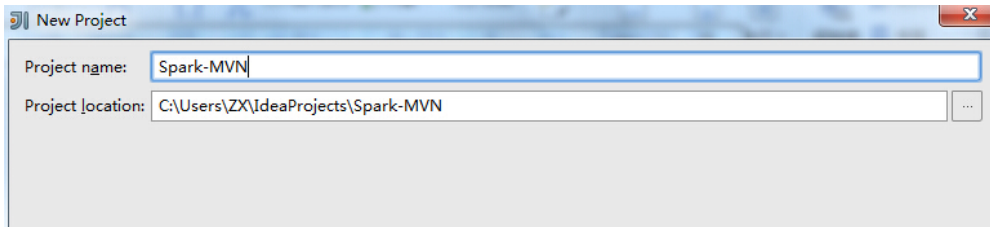
2. 选择 Maven 项目，然后点击 next



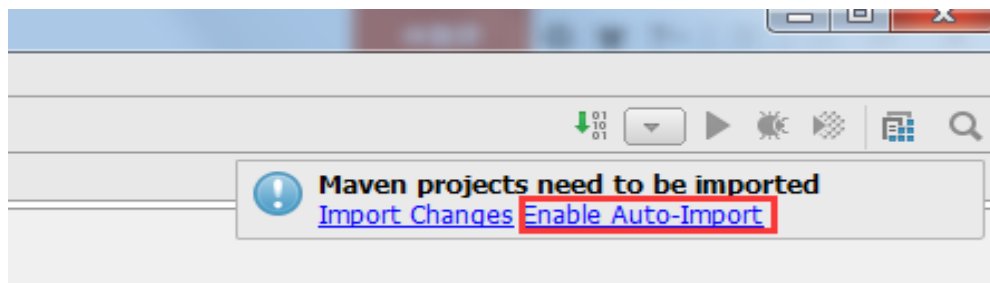
3. 填写 maven 的 GAV，然后点击 next



4.填写项目名称，然后点击 finish



5.创建好 maven 项目，点击 Enable Auto-Import



6.配置 Maven 的 pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>cn.itcast.spark</groupId>
  <artifactId>spark-mvn</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
    <encoding>UTF-8</encoding>
    <scala.version>2.10.6</scala.version>
    <scala.compat.version>2.10</scala.compat.version>
  </properties>
```

```
<dependencies>
  <dependency>
    <groupId>org.scala-lang</groupId>
    <artifactId>scala-library</artifactId>
    <version>${scala.version}</version>
  </dependency>

  <dependency>
    <groupId>org.apache.spark</groupId>
    <artifactId>spark-core_2.10</artifactId>
    <version>1.5.2</version>
  </dependency>

  <dependency>
    <groupId>org.apache.spark</groupId>
    <artifactId>spark-streaming_2.10</artifactId>
    <version>1.5.2</version>
  </dependency>

  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-client</artifactId>
    <version>2.6.2</version>
  </dependency>
</dependencies>

<build>
  <sourceDirectory>src/main/scala</sourceDirectory>
  <testSourceDirectory>src/test/scala</testSourceDirectory>
  <plugins>
    <plugin>
      <groupId>net.alchim31.maven</groupId>
      <artifactId>scala-maven-plugin</artifactId>
      <version>3.2.0</version>
      <executions>
        <execution>
          <goals>
            <goal>compile</goal>
```

```
        <goal>testCompile</goal>
      </goals>
    <configuration>
      <args>
        <arg>-make:transitive</arg>
        <arg>-dependencyfile</arg>
        <arg>${project.build.directory}/.scala_dependencies</arg>
      </args>
    </configuration>
  </execution>
</executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.18.1</version>
  <configuration>
    <useFile>>false</useFile>
    <disableXmlReport>>true</disableXmlReport>
    <includes>
      <include>**/*Test.*</include>
      <include>**/*Suite.*</include>
    </includes>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-shade-plugin</artifactId>
  <version>2.3</version>
  <executions>
    <execution>
      <phase>package</phase>
      <goals>
        <goal>shade</goal>
      </goals>
      <configuration>
        <filters>
```

```

        <filter>
          <artifact>*:*</artifact>
          <excludes>
            <exclude>META-INF/*.SF</exclude>
            <exclude>META-INF/*.DSA</exclude>
            <exclude>META-INF/*.RSA</exclude>
          </excludes>
        </filter>
      </filters>
      <transformers>
        <transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
          <mainClass>cn.itcast.spark.WordCount</mainClass>
        </transformer>
      </transformers>
    </configuration>
  </execution>
</executions>
</plugin>
</plugins>
</build>
</project>

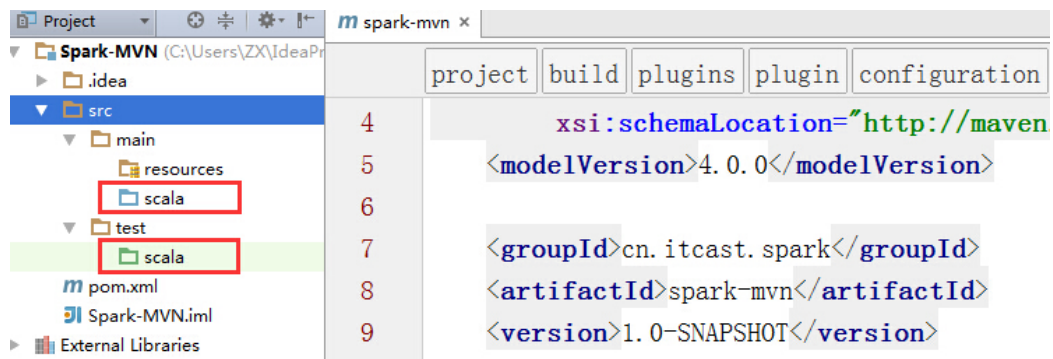
```

7.将 src/main/java 和 src/test/java 分别修改成 src/main/scala 和 src/test/scala，与 pom.xml 中的配置保持一致

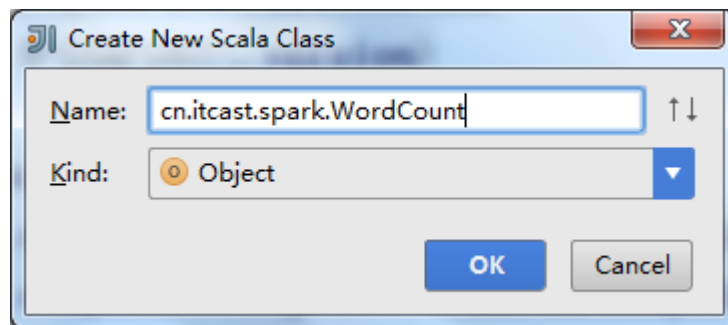
```

44
45 <build>
46   <sourceDirectory>src/main/scala</sourceDirectory>
47   <testSourceDirectory>src/test/scala</testSourceDirectory>
48   <plugins>
49     <plugin>
50       <groupId>net.alchim31.maven</groupId>
51       <artifactId>scala-maven-plugin</artifactId>
52       <version>3.2.0</version>
53       <executions>

```



8.新建一个 scala class，类型为 Object



9.编写 spark 程序

```
package cn.itcast.spark

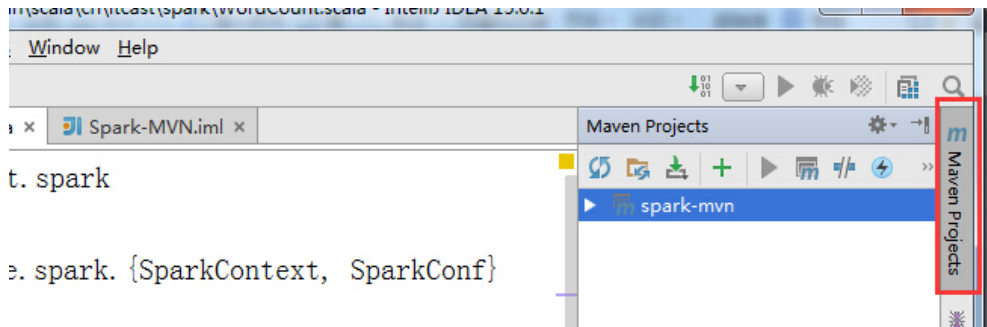
import org.apache.spark.{SparkContext, SparkConf}

object WordCount {
  def main(args: Array[String]) {
    //创建 SparkConf() 并设置 App 名称
    val conf = new SparkConf().setAppName("WC")
    //创建 SparkContext, 该对象是提交 spark App 的入口
    val sc = new SparkContext(conf)
    //使用 sc 创建 RDD 并执行相应的 transformation 和 action
    sc.textFile(args(0)).flatMap(_.split(" ")).map((_, 1)).reduceByKey(_+_ , 1).sortBy(_._2,
false).saveAsTextFile(args(1))
    //停止 sc, 结束该任务
    sc.stop()
  }
}
```

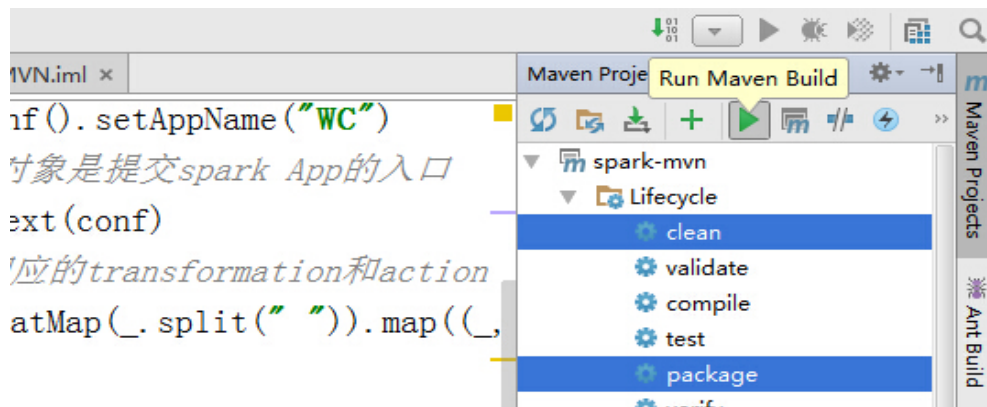
10.使用 Maven 打包：首先修改 pom.xml 中的 main class

```
<transformers>
  <transformer implementation="org.apache.maven.plugins.shade.resource.ManifestRes
    <mainClass>cn.itcast.spark.WordCount</mainClass>
  </transformer>
</transformers>
```

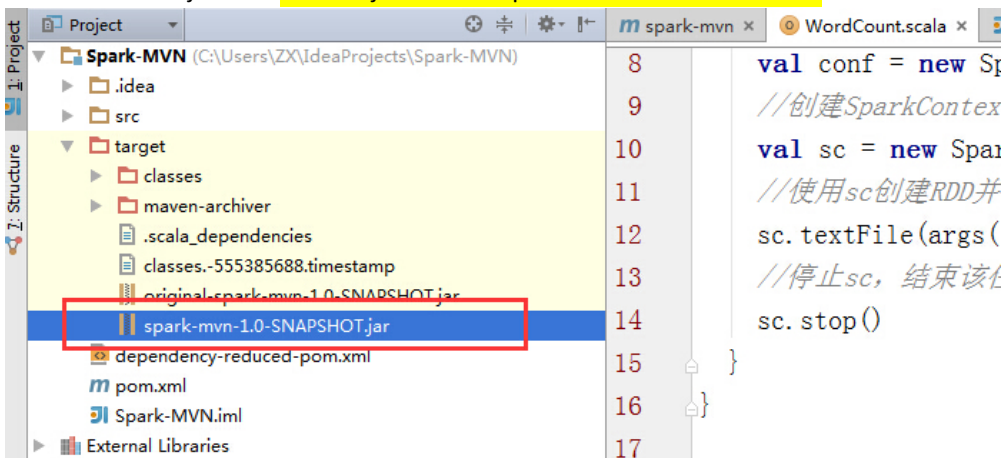
点击 idea 右侧的 Maven Project 选项



点击 Lifecycle，选择 clean 和 package，然后点击 Run Maven Build



11.选择编译成功的 jar 包，并将该 jar 上传到 Spark 集群中的某个节点上



12.首先启动 hdfs 和 Spark 集群

启动 hdfs

```
/usr/local/hadoop-2.6.1/sbin/start-dfs.sh
```

启动 spark

```
/usr/local/spark-1.5.2-bin-hadoop2.6/sbin/start-all.sh
```

13.使用 spark-submit 命令提交 Spark 应用（注意参数的顺序）

```
/usr/local/spark-1.5.2-bin-hadoop2.6/bin/spark-submit \
```

```
--class cn.itcast.spark.WordCount \
```

```
--master spark://node1.itcast.cn:7077 \
```

```
--executor-memory 2G \
--total-executor-cores 4 \
/root/spark-mvn-1.0-SNAPSHOT.jar \
hdfs://node1.itcast.cn:9000/words.txt \
hdfs://node1.itcast.cn:9000/out
```

14. 查看程序执行结果

```
hdfs dfs -cat hdfs://node1.itcast.cn:9000/out/part-00000
```

(hello,6)

(tom,3)

(kitty,2)

(jerry,1)

```
[root@mini1 app0]# spark-submit --class com.wolf.spark.wordcount.ScalaWordCount --master spark://mini1:7077 --executor-memory 1g --total-executor-cores 2 /root/apps/spark-1.0-SNAPSHOT.jar hdfs://mini1:9000/wordcount/input hdfs://mini1:9000/wordcount/output/spark/scala
/root/apps/spark-1.0.2-bin-hadoop2.6/conf/spark-env.sh: line 72: unexpected EOF while looking for matching `"'
/root/apps/spark-1.0.2-bin-hadoop2.6/conf/spark-env.sh: line 74: syntax error: unexpected end of file
java.lang.ClassNotFoundException: com.wolf.spark.wordcount.ScalaWordCount
    at java.net.URLClassLoader.findClass(URLClassLoader.java:301)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:348)
    at org.apache.spark.util.Utils$.addClassFromName(Utils.scala:175)
    at org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:689)
    at org.apache.spark.deploy.SparkSubmit$.doRunMain(SparkSubmit.scala:181)
    at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:206)
    at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:121)
    at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
```

在 spark 集群提交任务运行时报错：找不到 scala 程序的主类。

解决方案：pom 文件中添加 scala 插件。

```
<build>
  <sourceDirectory>src/main/scala</sourceDirectory>
  <testSourceDirectory>src/test/scala</testSourceDirectory>
  <plugins>
    <plugin>
      <groupId>net.alchim31.maven</groupId>
      <artifactId>scala-maven-plugin</artifactId>
      <version>3.2.0</version>
      <executions>
        <execution>
          <goals>
            <goal>compile</goal>
            <goal>testCompile</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```