

数加平台 OPEN_MR 的使用示例

-----by 余音(383700092)

---比赛交流群 155167917

目录

1 配置 MR 编程环境	1
maven 安装	1
添加远程 repository	2
新建项目	3
2 数加平台上使用 MR 参考:	5
编写代码	6
点击 export 导出 Jar	9
平台操作	12

1 配置 MR 编程环境

这个环境的基于御膳房建的，数加可以用来提交 OPEN_MR 程序

Eclipse 进行 MR 开发

maven 安装以及 eclipse 配置 maven

下载 maven 的 bin，在 apache 官方网站可以下载。

<http://www.apache.org/dyn/closer.cgi/maven/binaries/apache-maven-3.0.4-bin.tar.gz>

maven 安装

win7 解压

新建环境变量

MAVEN_HOME

F:\Program Files (x86)\apache-maven-3.3.3

PATH 后面添加

%MAVEN_HOME%\bin;

测试 cmd

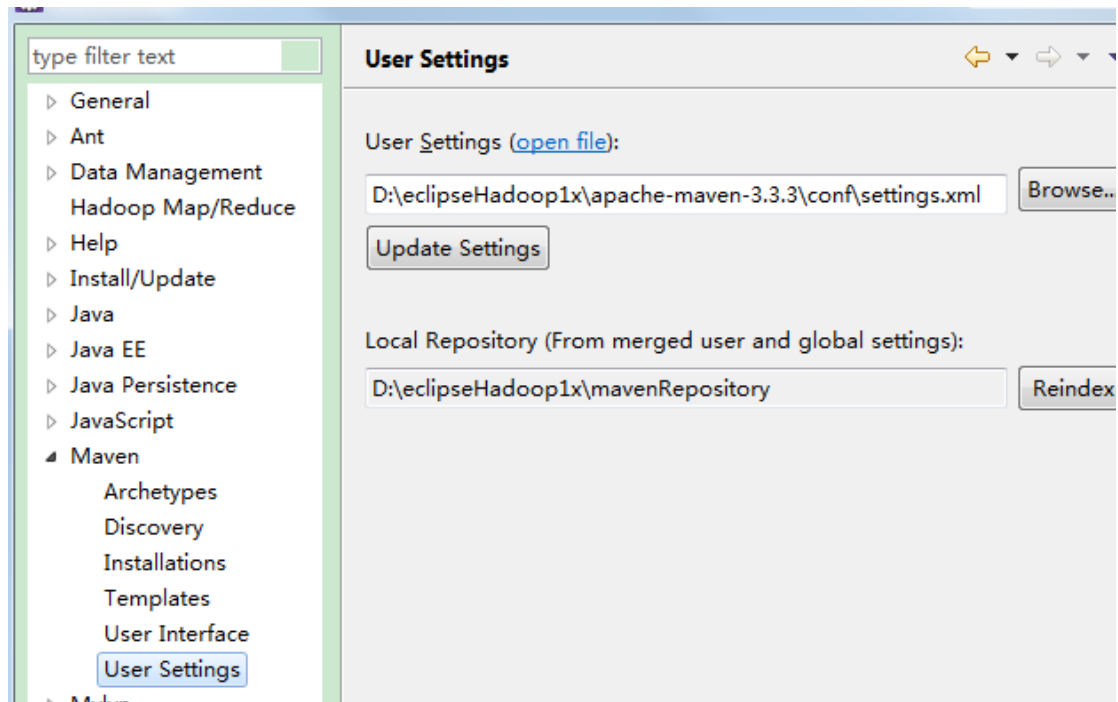
mvn -v

安装 Eclipse 的 Maven 插件：参考 <http://www.eclipse.org/m2e/>(或百度 eclipse 安装 maven 插件)因为每个人 eclipse 版本情况无法保证每一种都能成功

可参考： <http://jingyan.baidu.com/article/60ccbceb01de4d64cbb19756.html>

http://blog.csdn.net/wode_dream/article/details/38052639

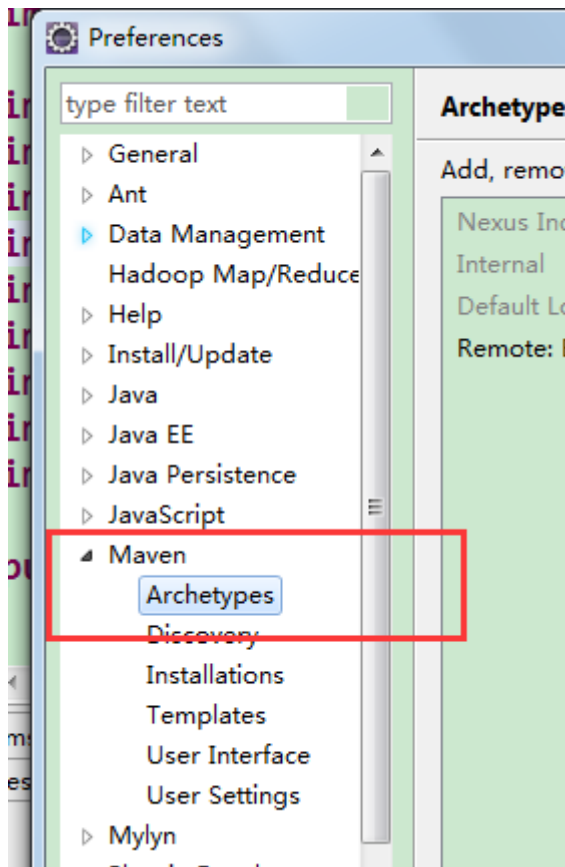
Maven 的 Eclipse 插件配置



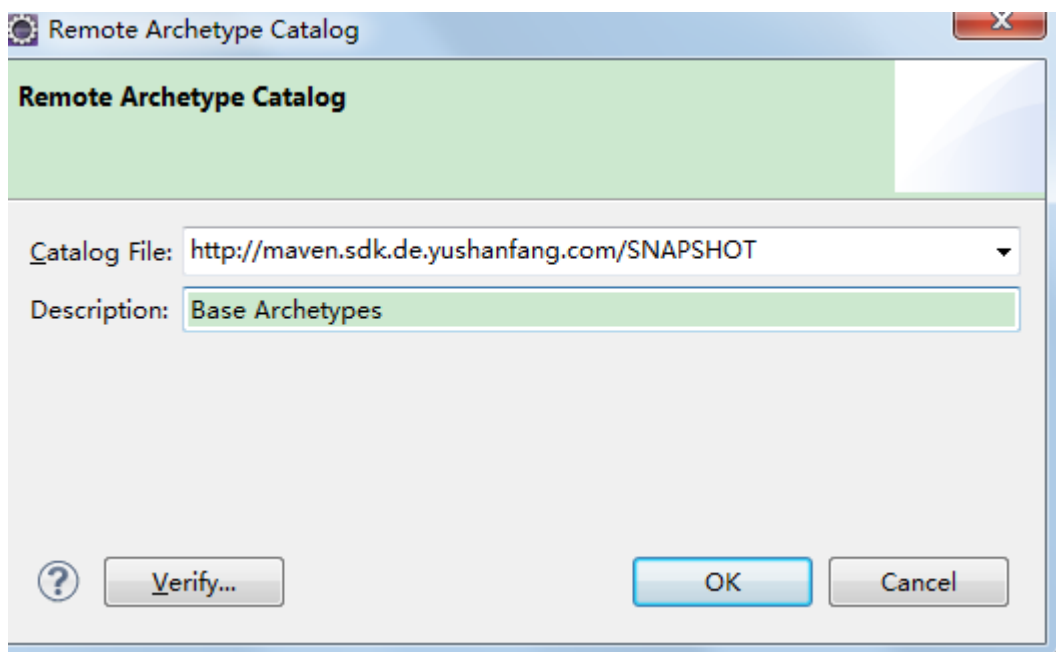
添加远程 repository

首先安装 maven，并且 Eclipse 中已安装 M2Eclipse 插件。

Eclipse 中依次点击 window-> Preferences -> Maven -> Arthetypes ，在打开的对话框中点击 Add Remote Catalog 按钮



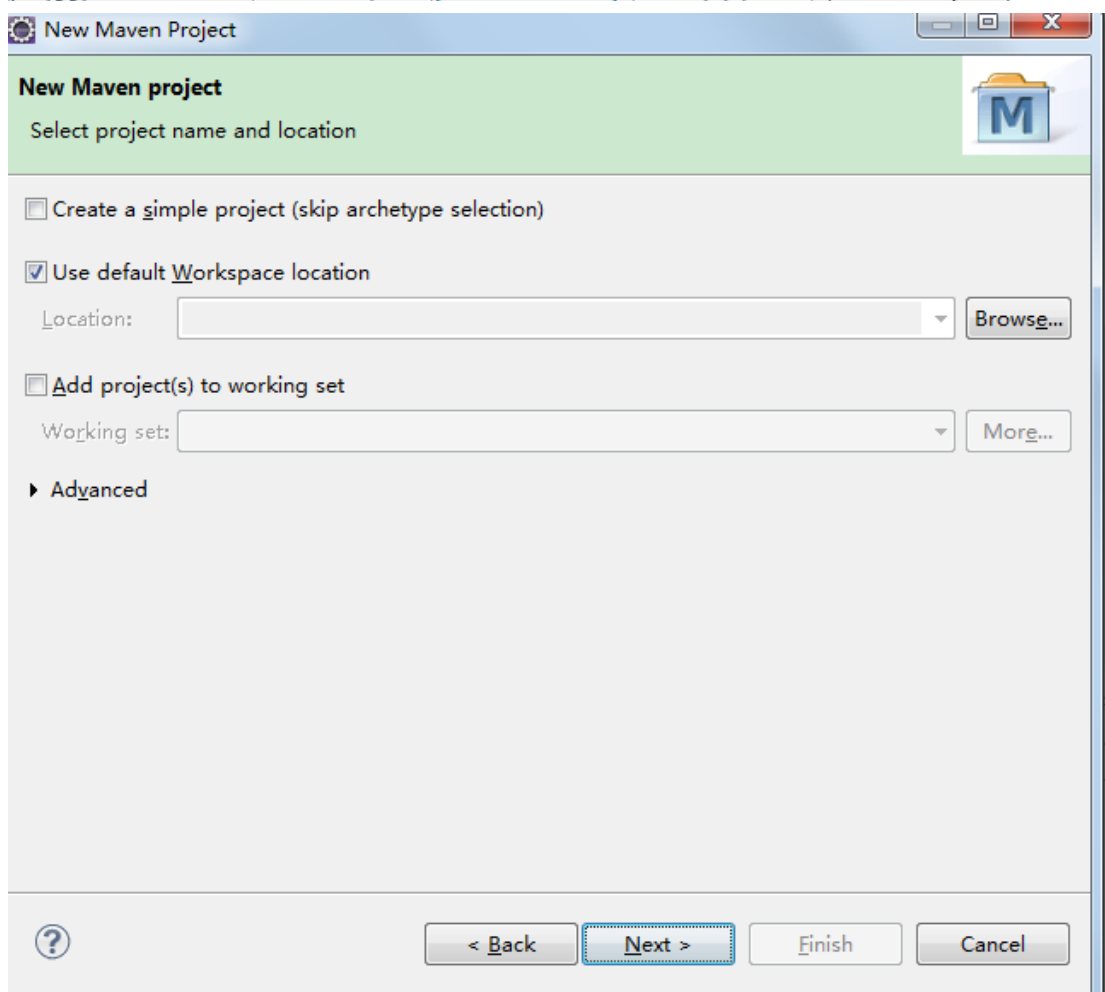
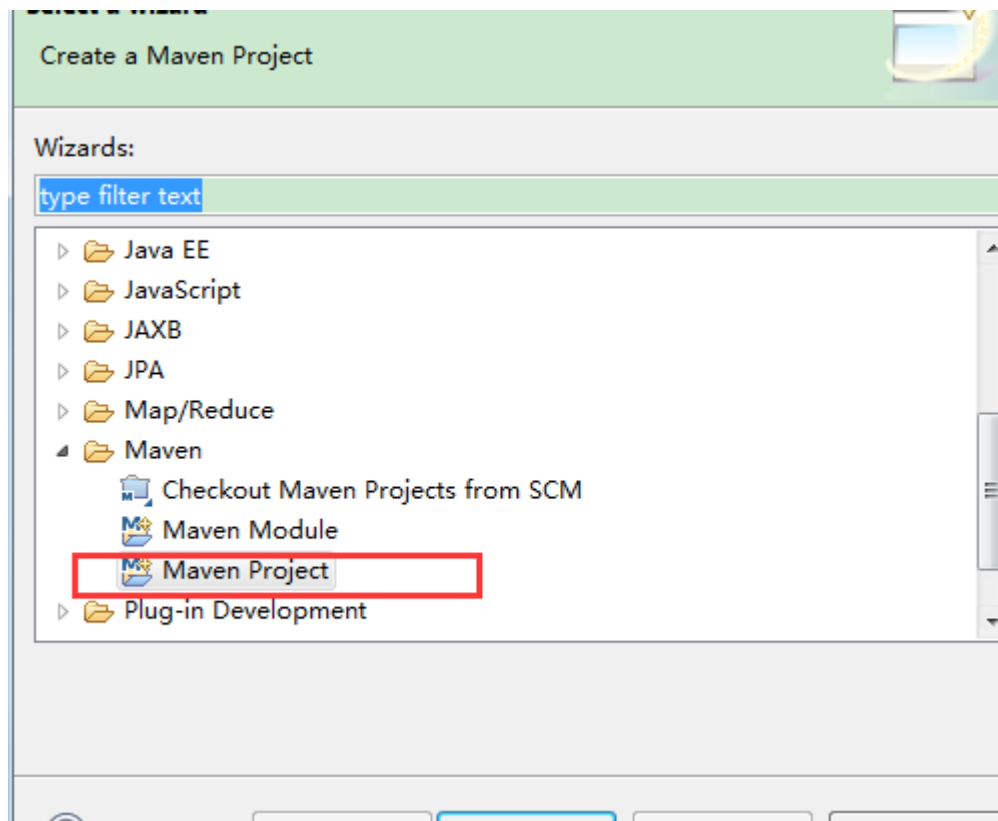
在打开的对话框中， Catalog File 填入 <http://maven.sdk.de.yushanfang.com/SNAPSHOT> ，
Description 填入 Base Archetypes



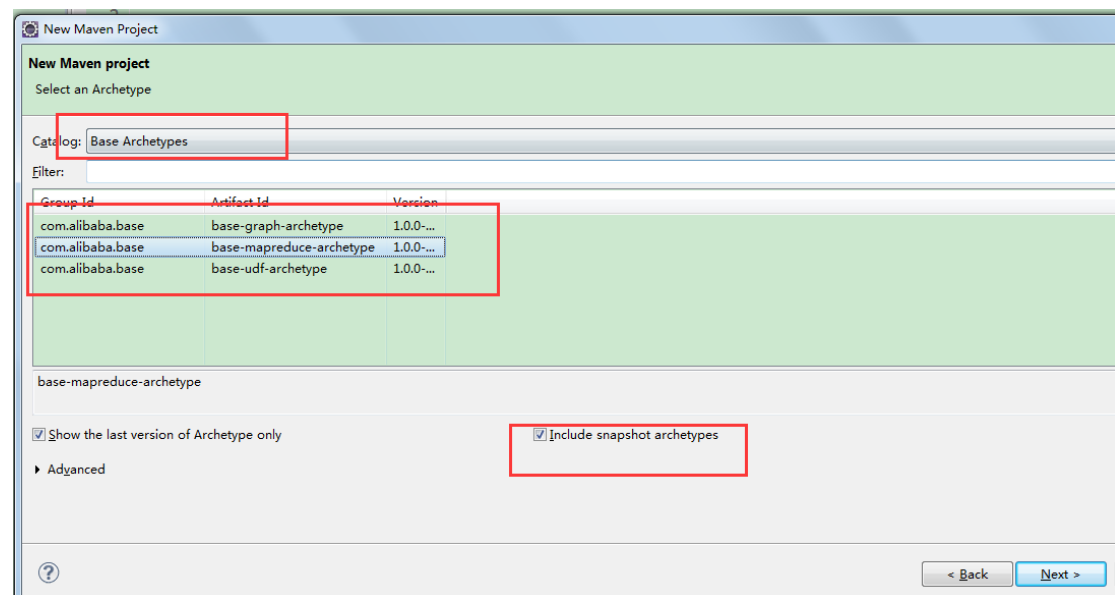
新建项目

在 Eclipse 中依次点击 File -> New -> Project.

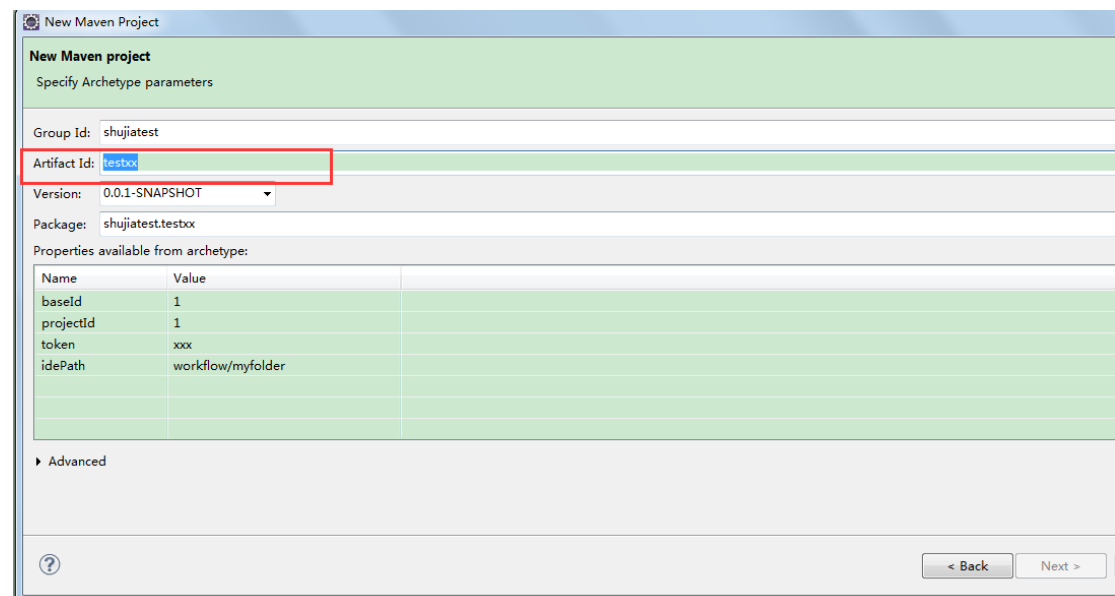
选择 maven 项目



按照下图选择



填入项目名



至此环境搭建 OK

2 数加平台上使用 MR 参考:

写在开头: OPEN_MR 在数加平台上可用首先, 建立一个 Hadoop 项目, 可以使用御膳房的文档搭建环境(目的使代码能够编译通过), 后面附一份我的御膳房项目(配置 maven 后可直接导入 eclipse 使用)。---配置好以上环境搭建的直接跳过就行

使用御膳房的文档搭建环境

<http://setting.tenant.yushanfang.com/portal/help/doc.html?spm=0.0.0.0.LBzMEE&file=MrUdfLo>

calDev

下载源码

<http://download.csdn.net/detail/q383700092/9550686>

编写代码

附上示例代码(来自官方):

```
package hadoop.TianChiMapreduce;
```

```
import java.io.IOException;
```

```
import java.util.Iterator;
```

```
import com.aliyun.odps.data.Record;
```

```
import com.aliyun.odps.data.TableInfo;
```

```
import com.aliyun.odps.mapred.JobClient;
```

```
import com.aliyun.odps.mapred.MapperBase;
```

```
import com.aliyun.odps.mapred.ReducerBase;
```

```
import com.aliyun.odps.mapred.TaskContext;
```

```
import com.aliyun.odps.mapred.conf.JobConf;
```

```
import com.aliyun.odps.mapred.utils.InputUtils;
```

```
import com.aliyun.odps.mapred.utils.OutputUtils;
```

```
import com.aliyun.odps.mapred.utils.SchemaUtils;
```

```
public class WordCount {
```

```
    public static class TokenizerMapper extends MapperBase {
```

```
        private Record word;
```

```
        private Record one;
```

```
        @Override
```

```
        public void setup(TaskContext context) throws IOException {
```

```
            word = context.createMapOutputKeyRecord();
```

```
            one = context.createMapOutputValueRecord();
```

```
            one.set(new Object[] { 1L });
```

```
            System.out.println("TaskID:" + context.getTaskID().toString());
```

```
        }
```

```
        @Override
```

```
        public void map(long recordNum, Record record, TaskContext context)
```

```
            throws IOException {
```

```
            for (int i = 0; i < record.getColumnCount(); i++) {
```

```

        word.set(new Object[] { record.get(0).toString() });
        context.write(word, one);
    }
}

/**
 * A combiner class that combines map output by sum them.
 */
public static class SumCombiner extends ReducerBase {
    private Record count;

    @Override
    public void setup(TaskContext context) throws IOException {
        count = context.createMapOutputValueRecord();
    }

    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext context)
        throws IOException {
        long c = 0;
        while (values.hasNext()) {
            Record val = values.next();
            c += (Long) val.get(0);
        }
        count.set(0, c);
        context.write(key, count);
    }
}

/**
 * A reducer class that just emits the sum of the input values.
 */
public static class SumReducer extends ReducerBase {
    private Record result = null;

    @Override
    public void setup(TaskContext context) throws IOException {
        result = context.createOutputRecord();
    }

    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContext context)
        throws IOException {

```

```

        long count = 0;
        while (values.hasNext()) {
            Record val = values.next();
            count += (Long) val.get(0);
        }
        result.set(0, key.get(0));
        result.set(1, count);
        context.write(result);
    }
}

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println("Usage: WordCount <in_table> <out_table>");
        System.exit(2);
    }

    JobConf job = new JobConf();

    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(SumCombiner.class);
    job.setReducerClass(SumReducer.class);

    job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));
    job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));

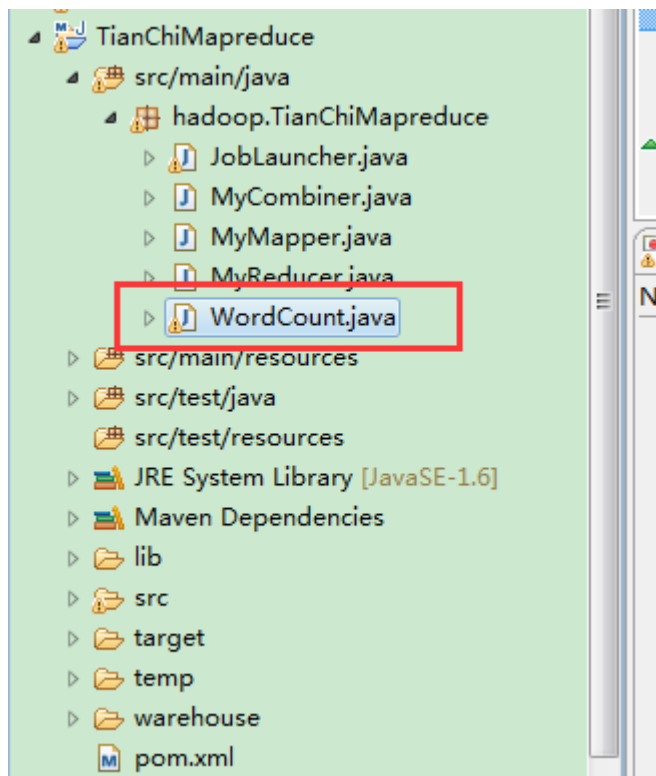
    InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(), job);
    OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(), job);

    JobClient.runJob(job);
}
}

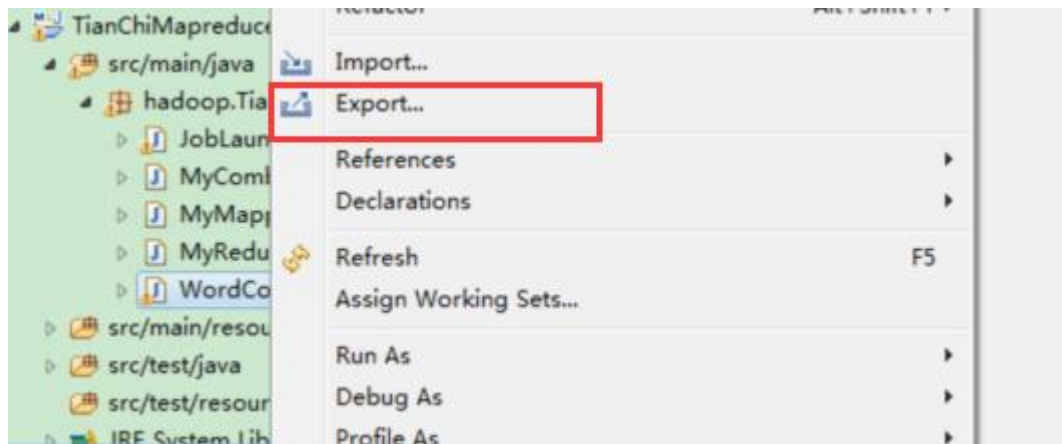
```

项目结构如下：

留一个标红的代码就行，其他没用。
















点击 export 导出 Jar



Export resources into a JAR file on the local file system.

Select an export destination:

type filter text

-  JAR file
-  Javadoc
-  Runnable JAR file
- ▲  Java EE
 -  App Client JAR file
 -  EAR file
 -  RAR file
- ▶  Plug-in Development
- ▶  Remote Systems
- ▶  Run/Debug
- ▶  Tasks
- ▶  Team
- ▶  Web

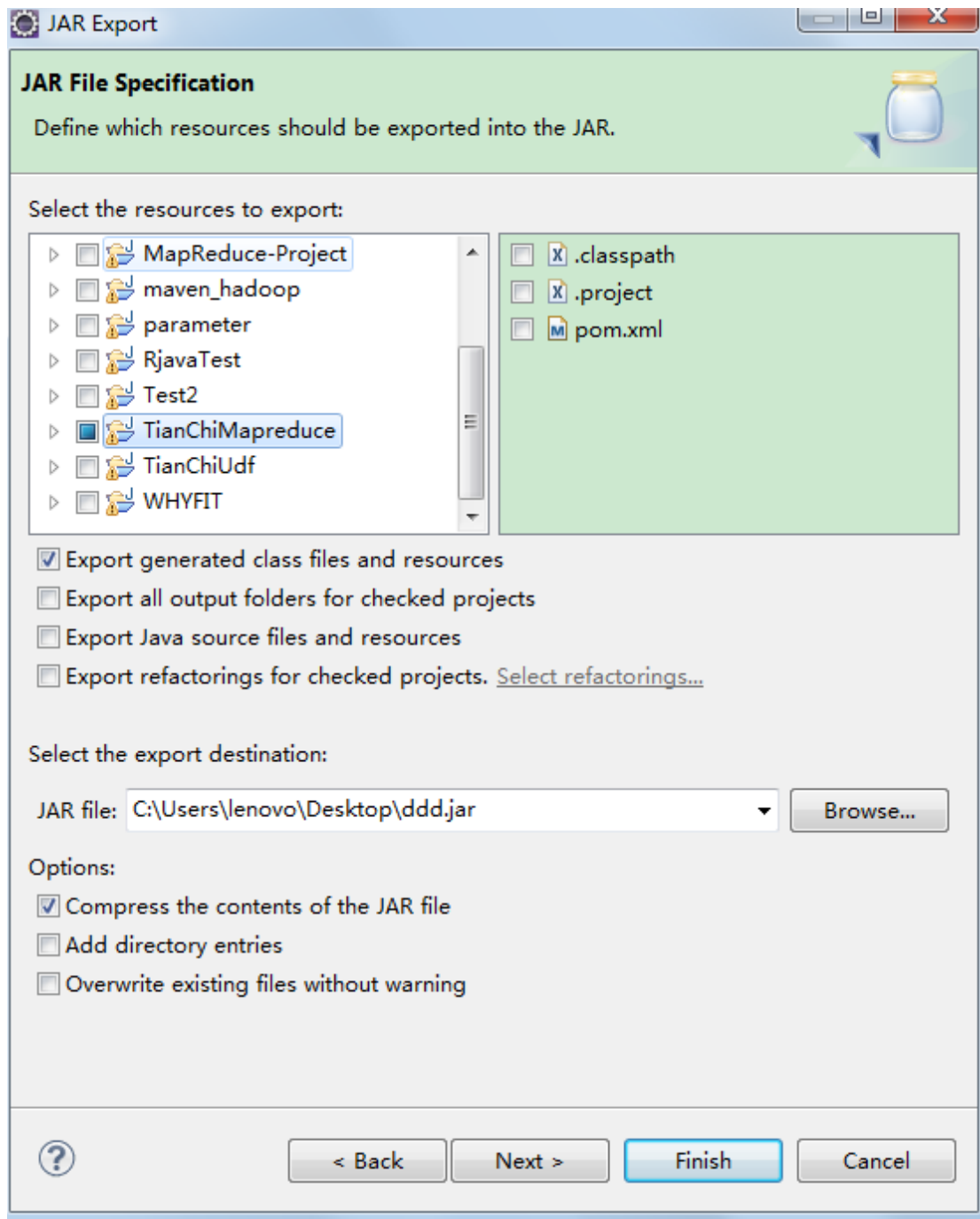


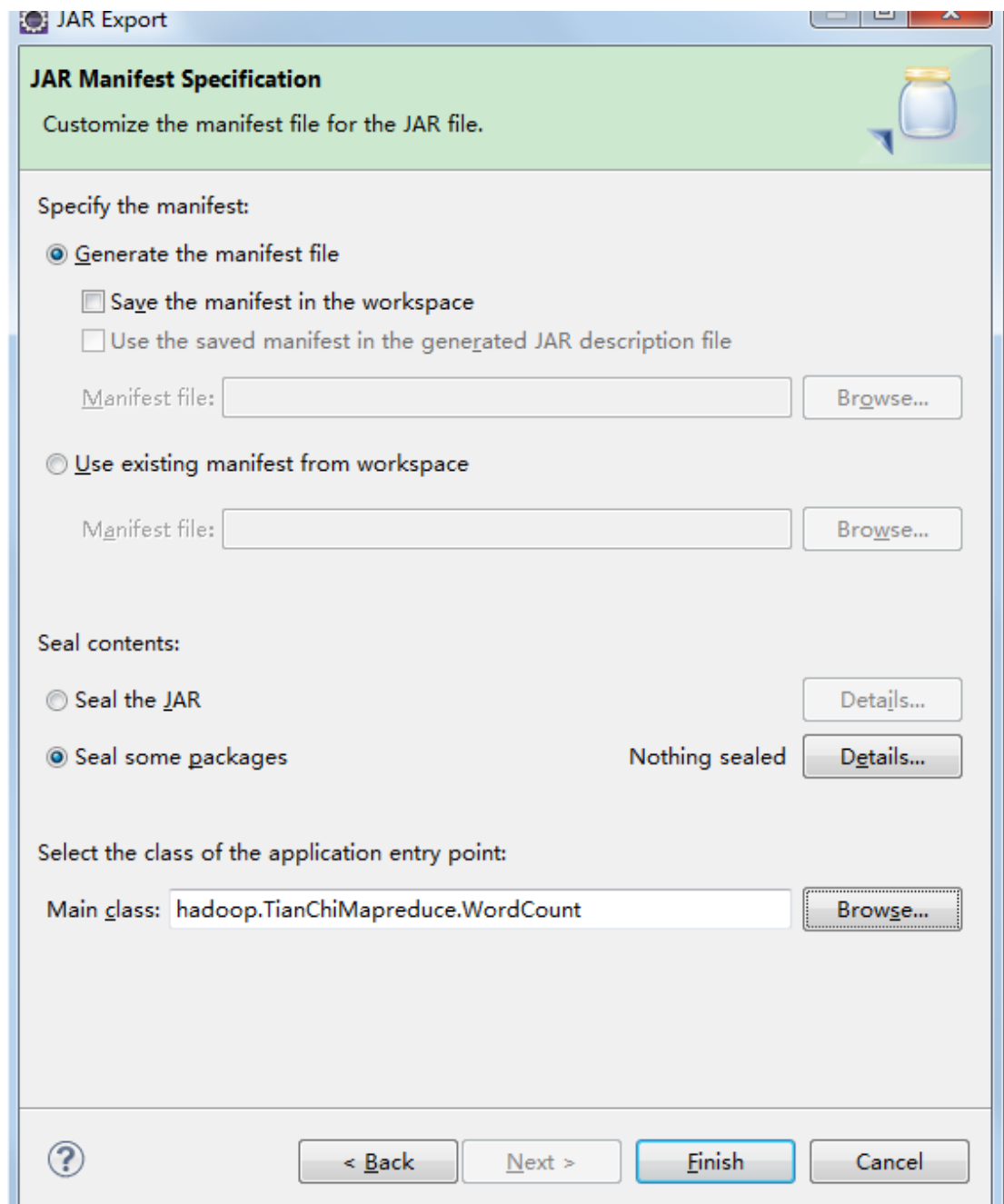
< Back

Next >

Finish

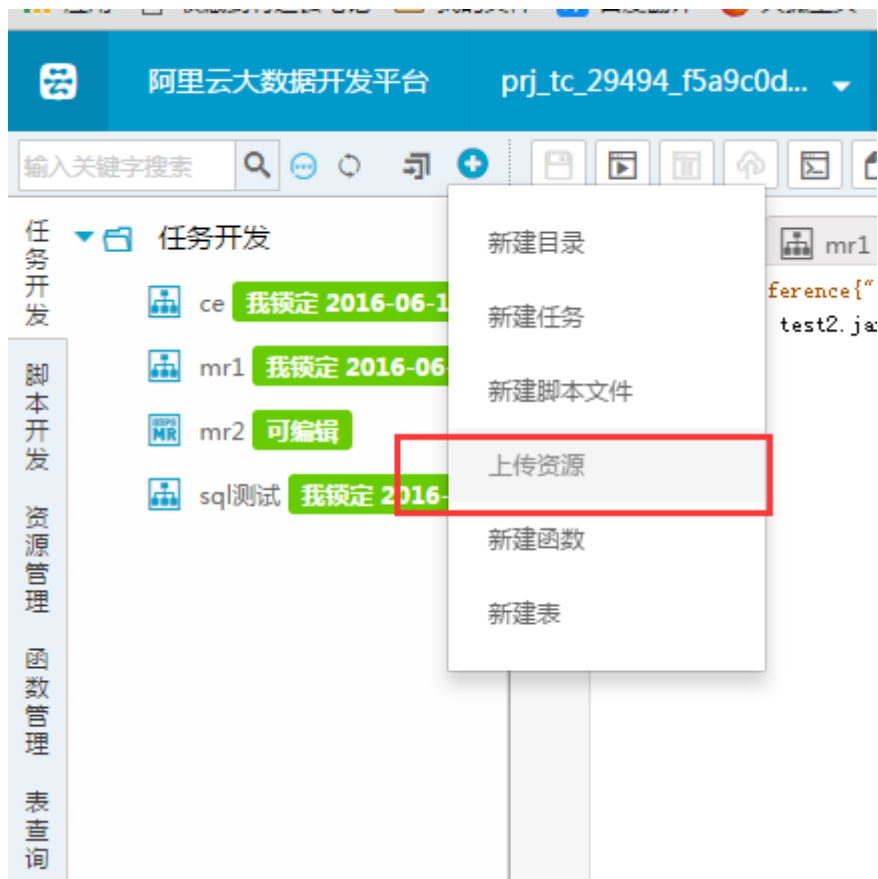
Cancel





平台操作

打开数加平台点击上传资源



资源上传

>

*名称:

*类型:

*上传:

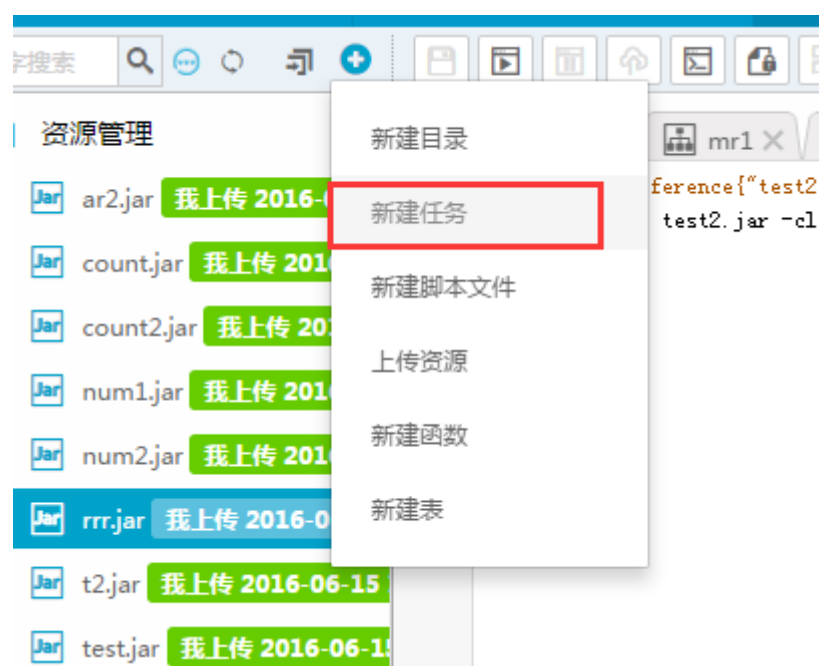
描述:

☒ 上传为ODPS资源

选择目录:

提交

取消



新建任务

*名称：

描述：

*任务类型：☒ 工作流任务 ☐ 节点任务

*调度类型：☐ 一次性调度 ☒ 周期调度

选择目录：

/

▶ ☐ 任务开发

创建

取消

拖动 OPEN_MR 组件到右侧空白



新建节点

*名称:	<input type="text" value="测试1"/>
*类型:	<input type="text" value="OPEN_MR"/>
描述:	<input type="text" value="请输入描述信息"/>

双击组件-按如下填写，前两个是你需要运行的 jar，mapper 和 reducer 是你需要运行的 map 类和 reudce 类，输出表要提前创建好，key 和 value 对应输出表的字段

建表 sql: create table if not exists wcout1 (key string,cnt bigint);

MRJar包

ddd.jar

+

-

资源

ddd.jar

+

输入表

avg53

+

-

mapper

hadoop.TianChiMapreduce.WordCount\$TokenizerMapper

必选

reducer

hadoop.TianChiMapreduce.WordCount\$SumReducer

combiner

选填。Combiner class全名。如果不填则没有combine步骤

输出表

wcout1

输出Key

key:string

输出Val

cnt:bigint

填写好之后保存运行

附录:

wcout1 的表结构

Field	Type	Label	Comment
key	string		
cnt	bigint		

运行成功

```
Running job in console.
2016-06-15 19:27:15 start to get jobId:
2016-06-15 19:27:15 get jobid:20160615112715401gg97idm2
InstanceId: 20160615112715401gg97idm2
http://logview.odps.aliyun.com/logview/?h=http://service-all.ext.odps.aliyun-ir
FBTX09CTzpwNF8yNTA1OTUwNjU5MDQwNzg2NzUsMTQ2NjU5NDgzNSx7I1N0YXR1bWVudCI6W3siQWNK
1jMGRkNWZYS9pbnN0YW5jZXMvMjAxNjA2MTUxMTI3MTU0MDFnZzk3aWRtMiJdfV0sI1ZlcnNpb24iO
...
...
2016-06-15 19:27:30 M1_job0:0/0/1[0%] R2_1_job0:0/0/1[0%]
2016-06-15 19:27:38 M1_job0:0/1/1[100%] R2_1_job0:0/0/1[0%]
2016-06-15 19:27:47 M1_job0:0/1/1[100%] R2_1_job0:0/1/1[100%]
2016-06-15 19:27:55 M1_job0:0/1/1[100%] R2_1_job0:0/1/1[100%]
2016-06-15 19:28:03 M1_job0:0/1/1[100%] R2_1_job0:0/1/1[100%]
```

结果

日志	结果[1] ×	结果[2] ×	结果[3] ×	
序号	key	cnt		
1	031460402eed3e	180		
2	05f476c6f043a73	180		
3	0693f5e1c570d96	180		
4	088ff4b427effea3	180		
5	0a5d7c11c11c711	180		
6	0c80008b0a28d3	180		
7	140ef1f57652ab4	180		
8	151e09cf1ad8a2f	180		
9	16b019a469052c	180		