实验 HDFS 的 Java API 操作实验报告

江昱峰 21009200038

2023 年 11 月 13 日

1 背景介绍

HDFS 在生产应用中主要是客户端的开发，其核心步骤是从 HDFS 提供的 API 中构造一个 HDFS 的访问客户端对象，然后通过该客户端对象操作（增删改查）HDFS 上的文件。

2 实验目的

实践并掌握 HDFS 的 Java API 操作，具体包括以下四部分内容：

- Linux 下 Eclipse 连接 Hadoop；
- 上传查看文件操作;
- 修改删除文件操作;
- 小文件合并。

3 实验知识

- API 介绍：API（Application Programming Interface，应用程序接口）是一些预先定义的接口（如函数、HTTP 接口），或指软件系统不同组成部分衔接的约定。用来提供应用程序与开发人员基于某软件或硬件得以访问的一组例程，而又无需访问源码，或理解内部工作机制的细节。可简单理解 API 是一个中间件，我们的程序通过它实现与系统交互，把各种数据，应用和设备连系在一起，进行协同工作。

- 操作 HDFS 文件 API 概述：Hadoop 中关于文件操作类基本上全部是在 org.apache.hadoop.fs 包中，这些 API 能够支持的操作包含：打开文件，读写文件，删除文件等。
- 小文件合并：由于 Hadoop 擅长存储大文件，因为大文件的元数据信息比较少，如果 Hadoop 集群当中有大量的小文件，那么每个小文件都需要维护一份元数据信息，会大大的增加集群管理元数据的内存压力，所以在实际工作当中，如果有必要一定要将小文件合并成大文件进行一起处理, 可以在上传的时候将小文件合并到一个大文件里面去。

4 实验要求

完成 HDFS 的 Java API 操作，具体包括以下四部分任务：

- Linux 下 Eclipse 连接 Hadoop；
- 上传查看文件操作;
- 修改删除文件操作;
- 小文件合并。

5 实验环境

本次实验实验环境为青椒课堂平台的 Linux（Ubuntu 20.04）操作系统。

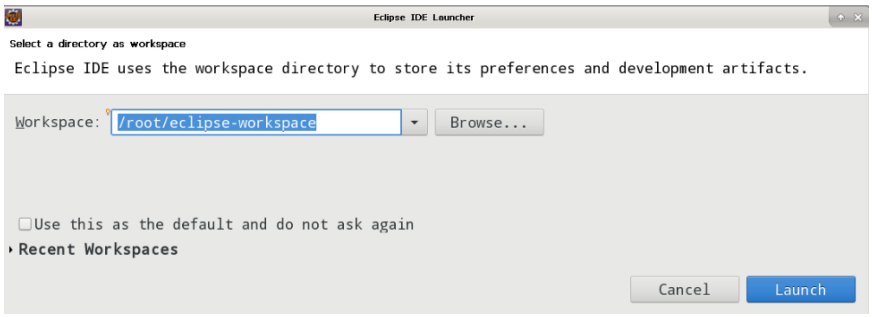
6 实验步骤

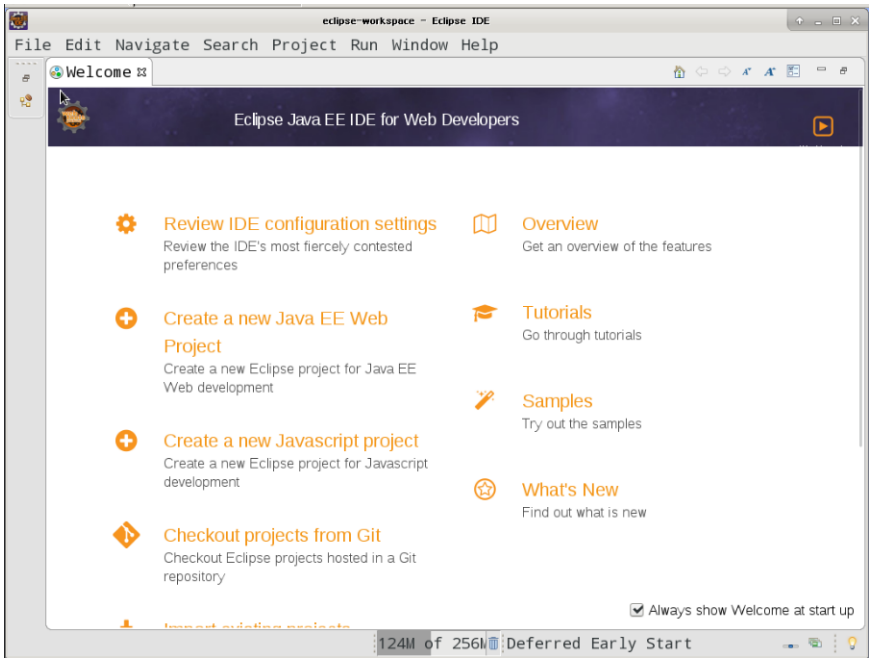
6.1 Linux 下 Eclipse 连接 Hadoop

6.1.1 任务 1：Linux 下 Eclipse 连接 Hadoop

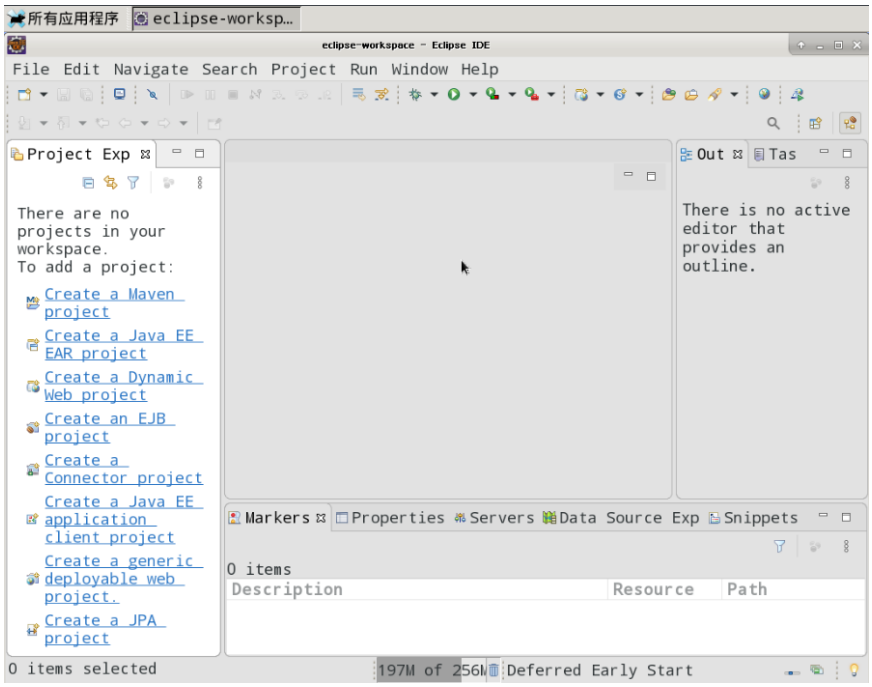
Linux 下 Eclipse 连接 Hadoop：

1. 已上传 hadoop-eclipse-plugin-2.7.7.jar，路径已设置为 plugins 目录。
2. 启动 Eclipse，选择默认路径即可。

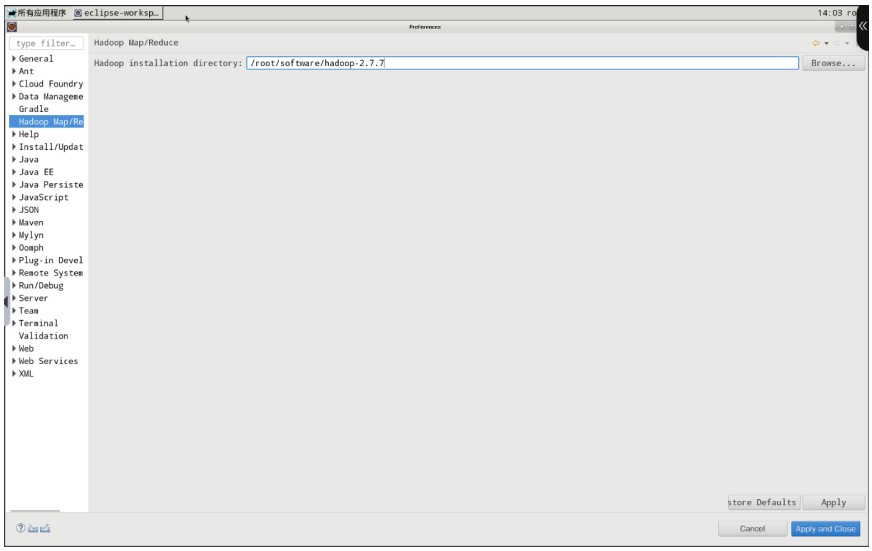




3. 关闭 Welcome 页面。

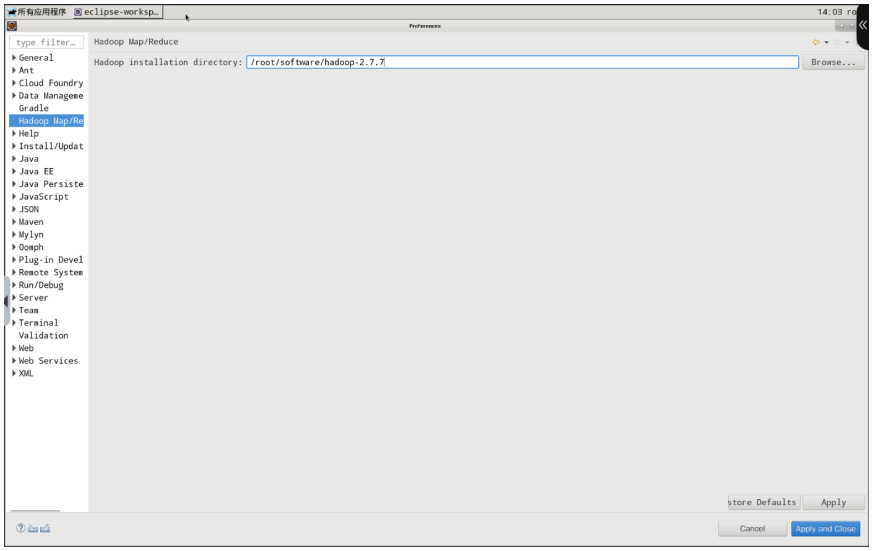


4. 点击 Window->Preferences，左侧对话框选择 Hadoop Map/Reduce 选项。



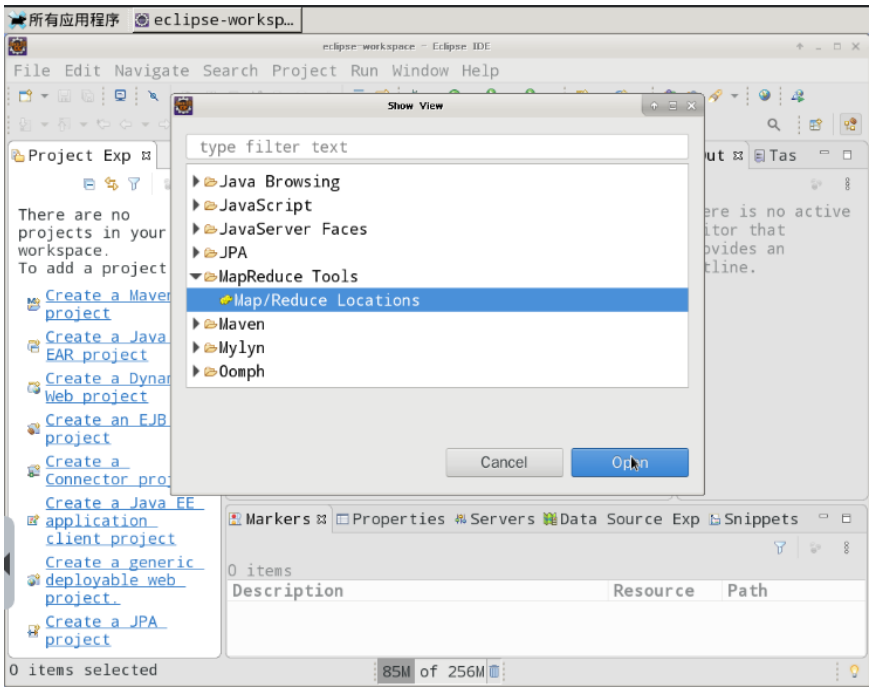
5. 设置 hadoop-2.7.7 的安装路径。（实验环境 hadoop-2.7.7 的安装目录

为：/root/software/hadoop-2.7.7）。

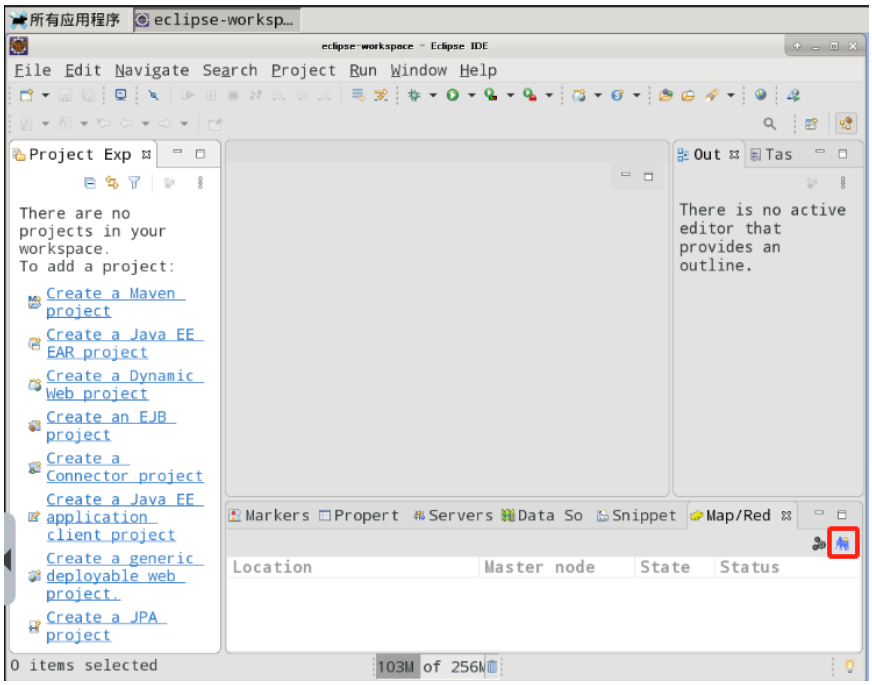


6. 点击 Apply and Close。

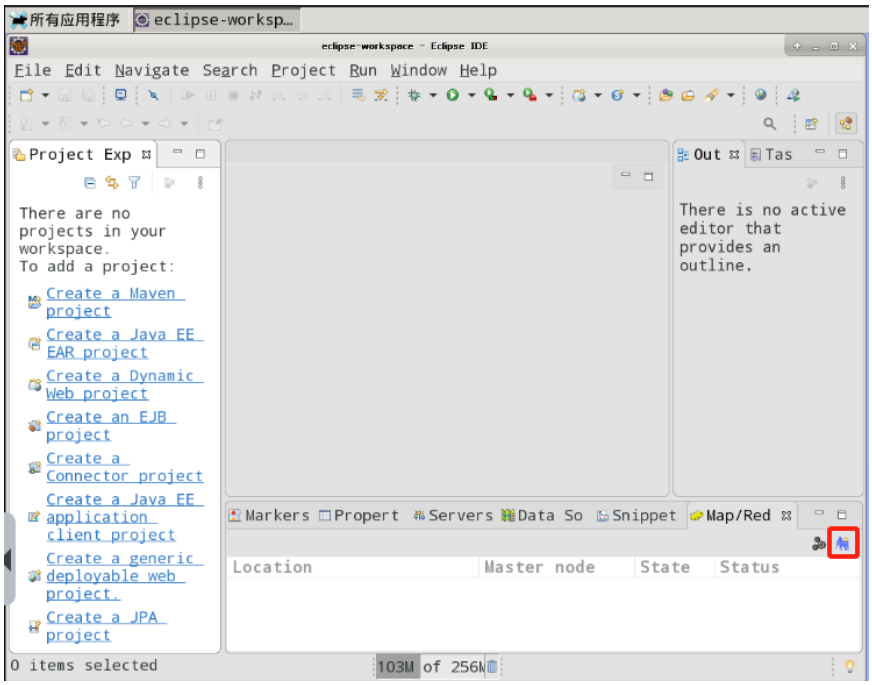
7. 点击 Window->Show View->Other，弹出 Show View 对话框，选中 MapReduce Tools 下的 Map/Reduce Locations，点击 Open。



8. Eclipse 底部出现 Map/Reduce Locations 窗口，点击其右边的蓝色小象图标。

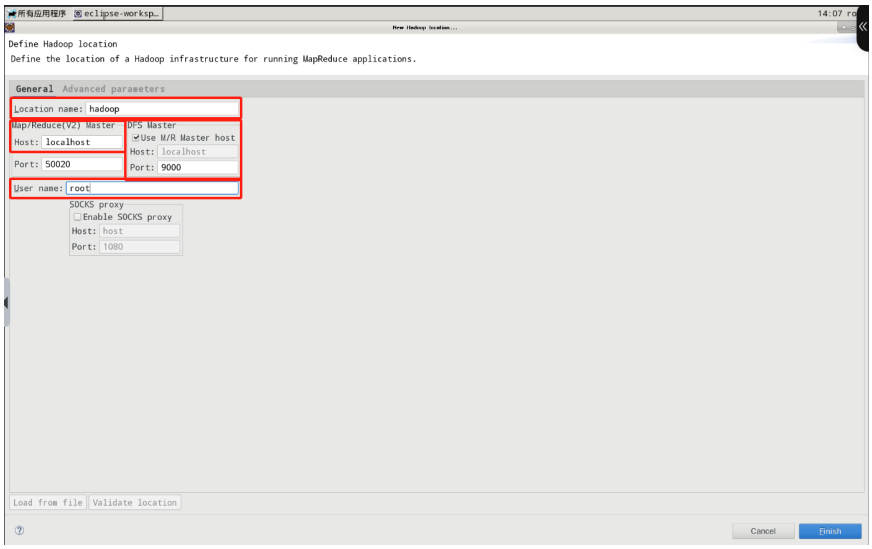


9. 弹出 New Hadoop locaiton... 对话框，将 Location name 命名为 hadoop。



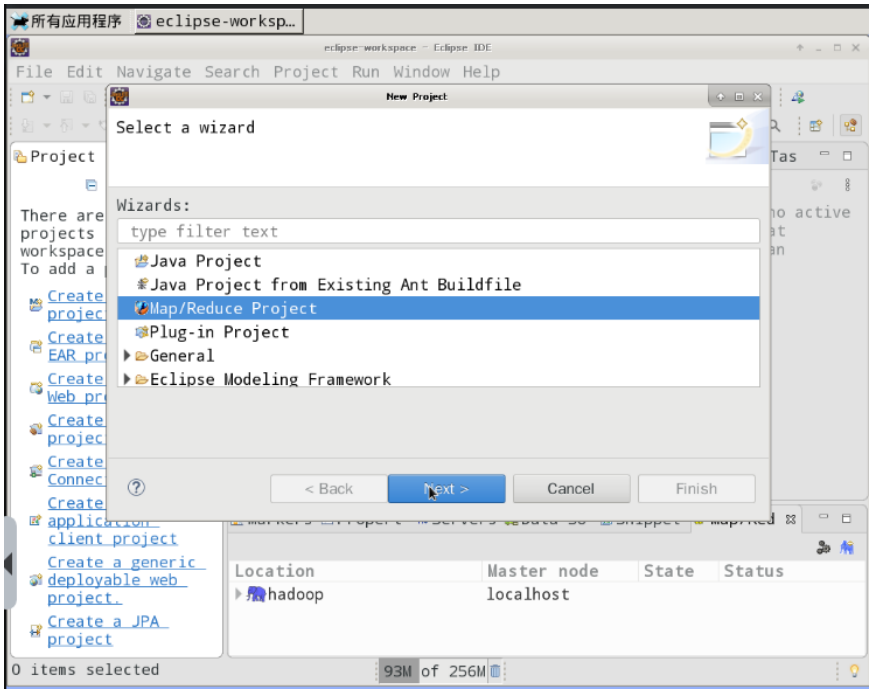
10. Map/Reduce(V2)Master 下将 Host 修改为 YARN 集群主节点的 IP 地址或主机名 localhost。

11. DFS Master 下将 Host 修改为 HDFS 集群主节点的 IP 地址或主机名 localhost，Port 修改为 9000，将 User name 设置为搭建集群用户 root。

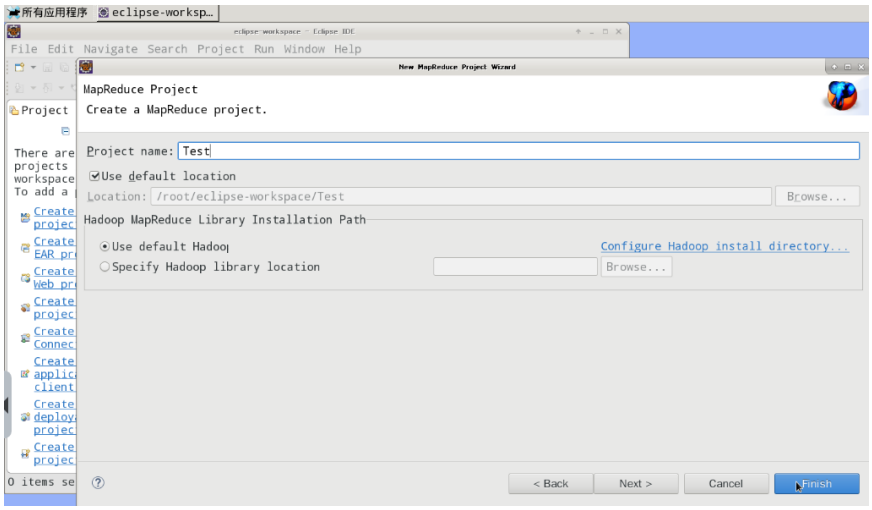


测试连接:

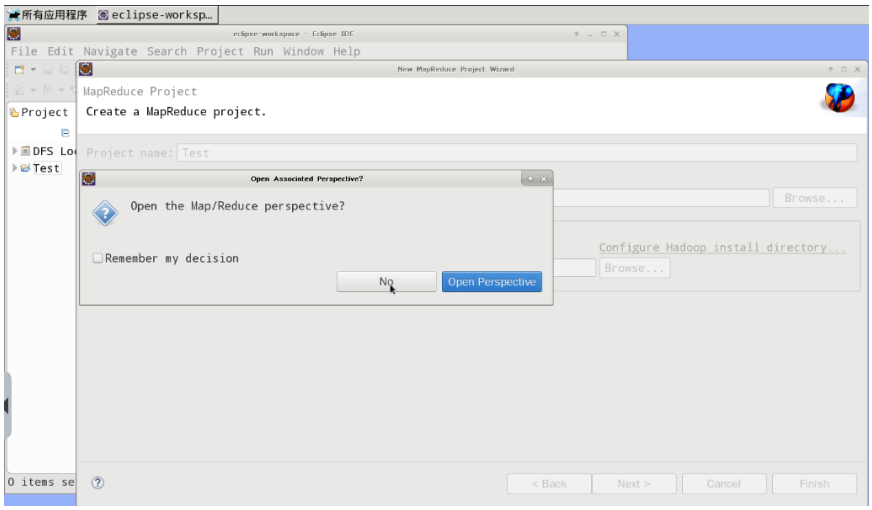
1. 选择 File->New->Project->Map/Reduce Project->Next，弹出 new MapReduce Project Wizard 对话框。



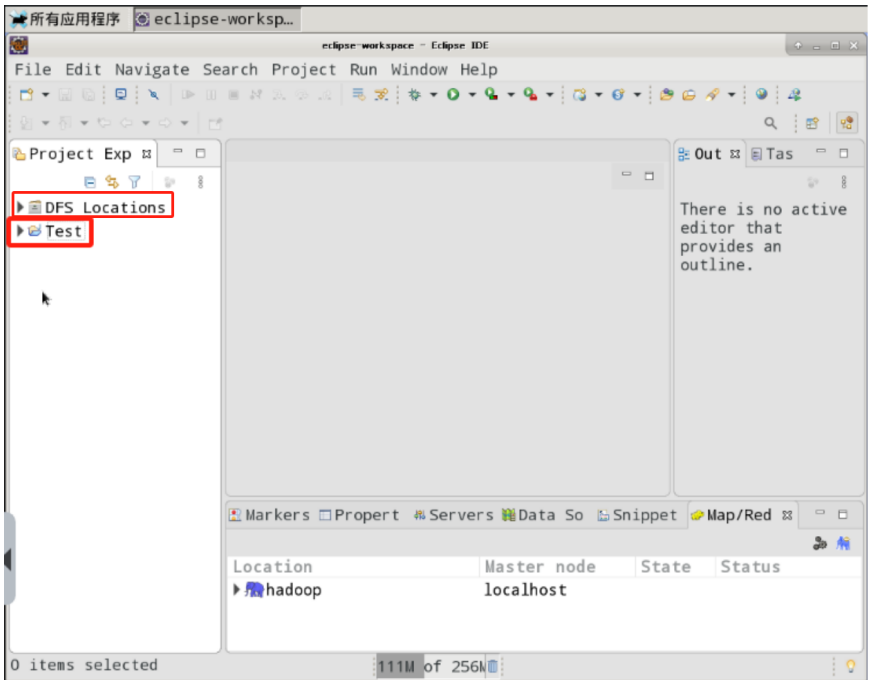
2. 为 Project name 起名为 Test，点击 Finsh。



3. 弹出 Open Associated Perspective 对话框选择 No。



4. 在左侧 Project Explorer 下出现 DFS Locations 列表栏和 Test。

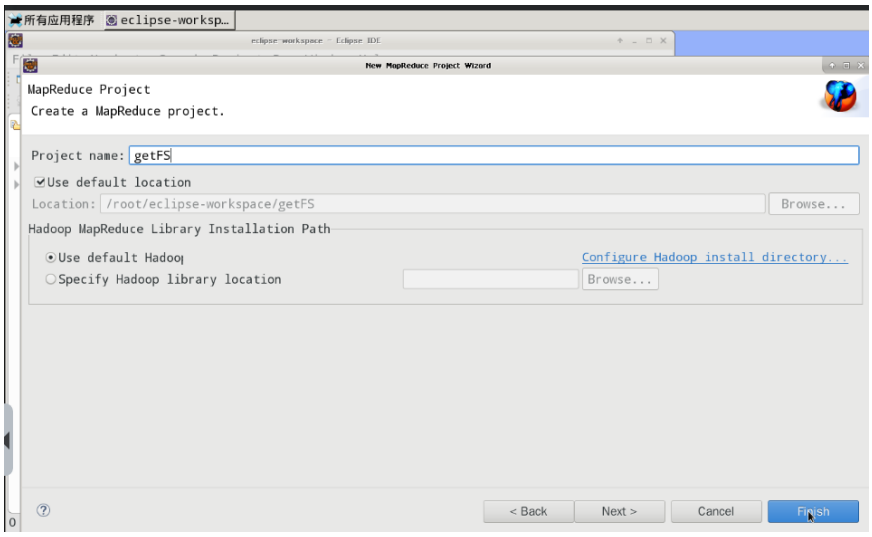


至此，Hadoop 已经可以连接上 Eclipse。后续可进行程序开发。

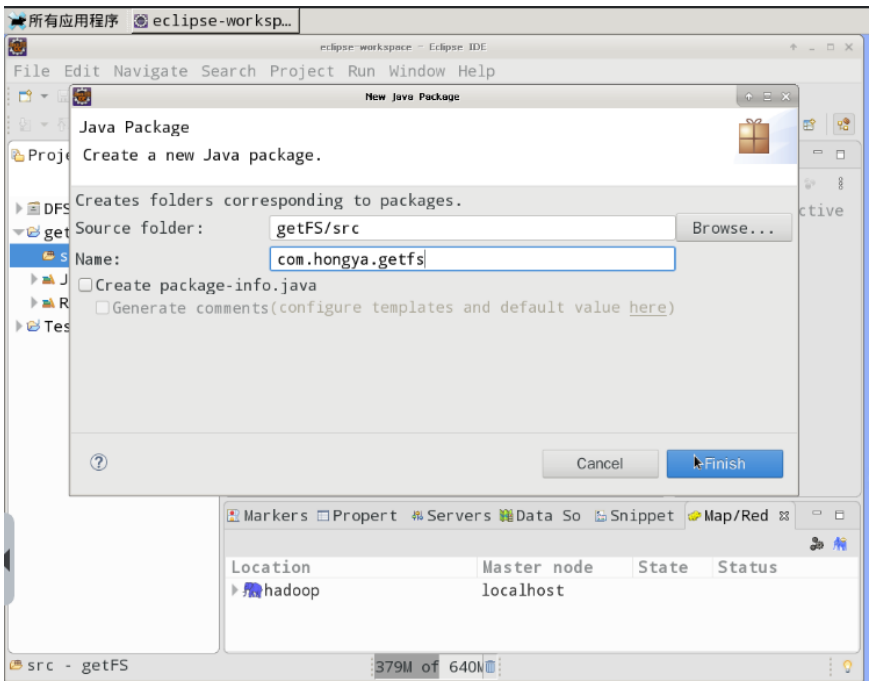
6.1.2 获取 FileSystem 实例演示

创建项目：

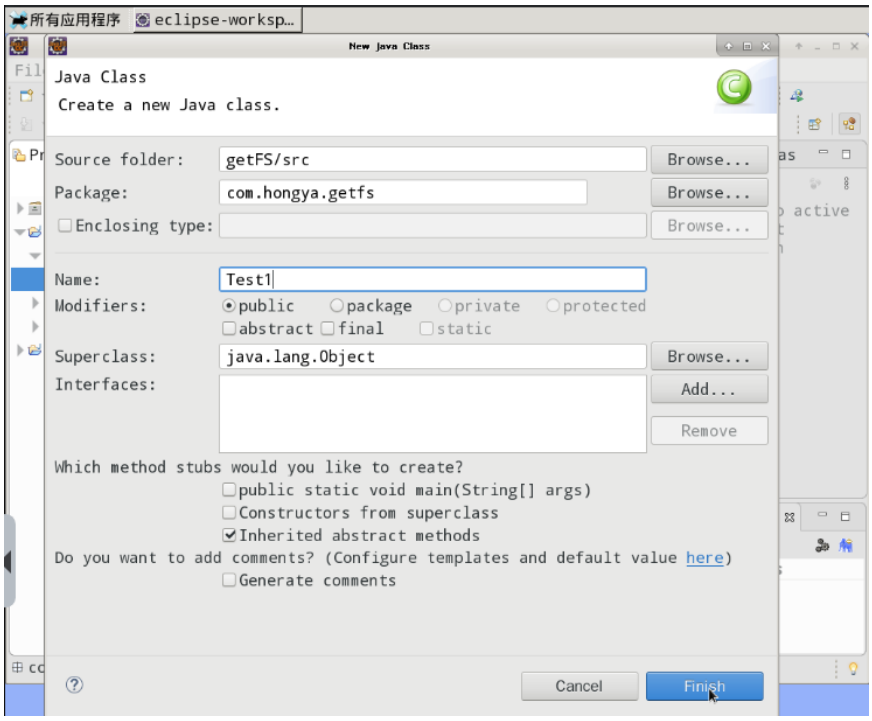
1. 选择 File->New->Project->Map/Reduce Project->Next，弹出 new MapReduce Project Wizard 对话框。Project name 命名为：getFS，点击 Finsh

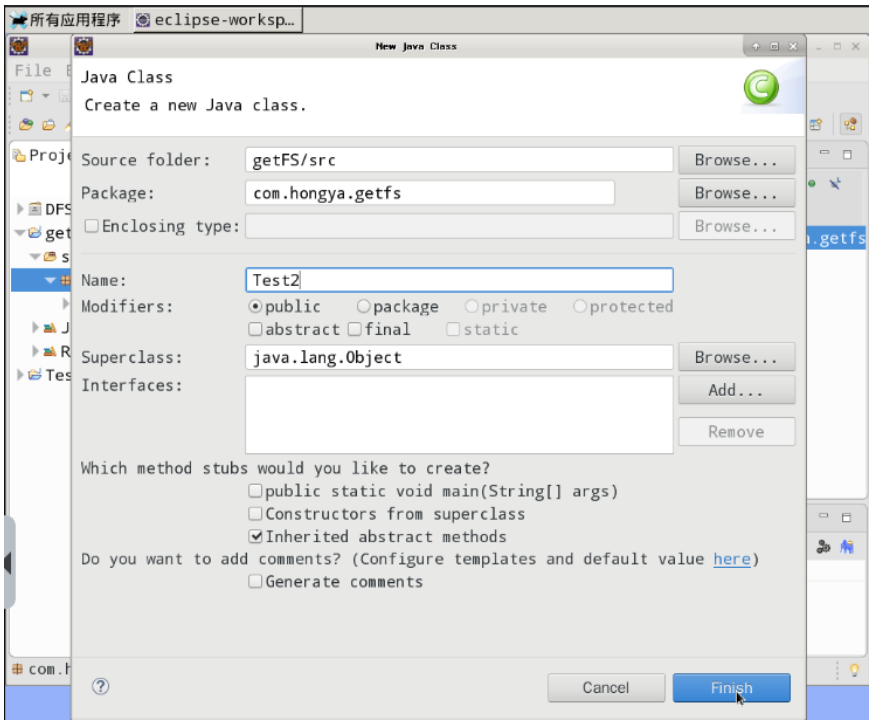


2. 点击左侧 getFS，在 src 下创建包结构：com.hongya.getfs。



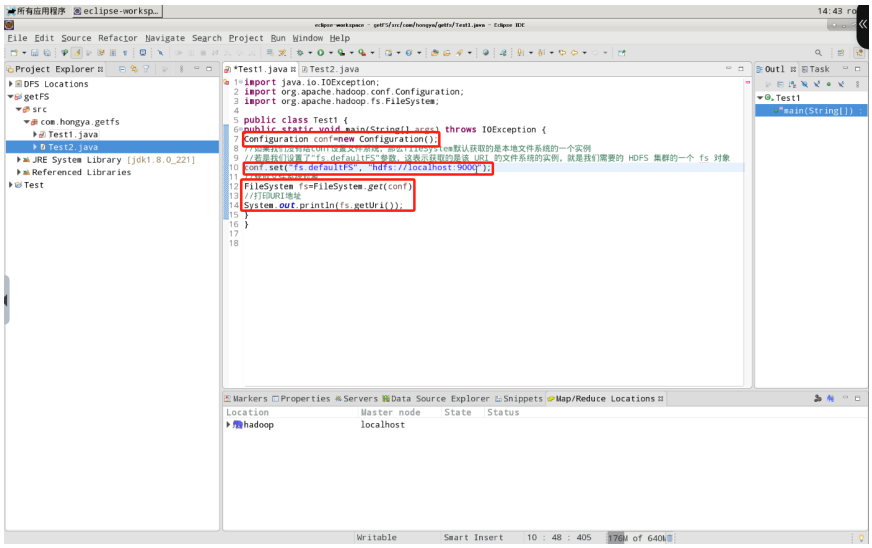
3. 在包下创建测试类 Test1，Test2。


所有应用程序
eclipse-worksp...
New Java Class
Java Class
Create a new Java class.
Source folder:
getFS/src
Browse...
Package:
com.hongya.getfs
Browse...
Enclosing type:
Browse...
Name:
Test1
Modifiers:
public
package
private
protected
abstract
final
static
Superclass:
java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish



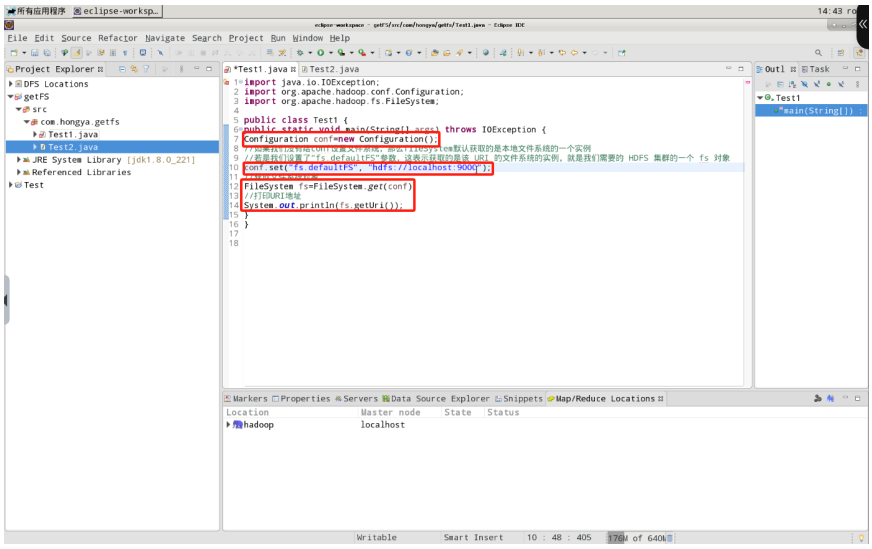
通过配置获取 FileSystem 对象：

1. 在 Test1 下使用 new 构造 Configuration 对象，获取配置信息类。
2. 通过 set 方法设置配置项，指定文件系统为 HDFS，即 fs.defaultFS 参数的值为 hdfs://localhost:9000。
3. 获取文件系统对象并打印 URI 地址。



直接获取 FileSystem 对象：

1. 在 Test2 中用 new 构造 Configuration 对象，获取配置信息类。
2. 通过 get 方法获取 HDFS 文件系统对象，设置 URI 为 hdfs://localhost:9000，安装集群用户名为 root。
3. 打印获取的文件系统 URI。

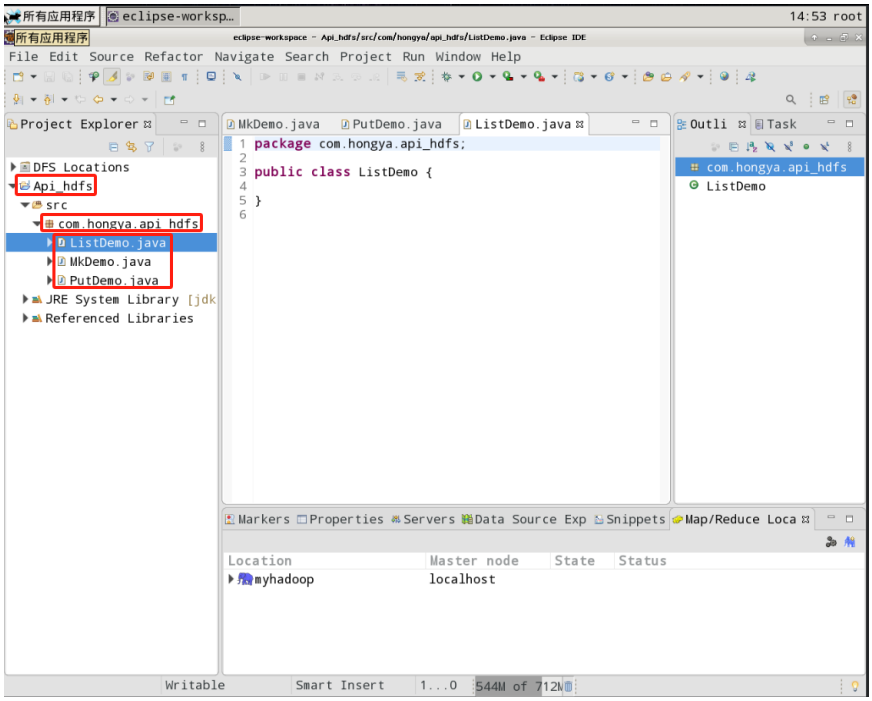


6.2 上传查看文件操作

6.2.1 创建目录

创建项目：

1. 项目名：Api_hdfs。
2. 包名：com.hongya.api_hdfs 。
3. 类：
   - 创建目录类：MkDemo。
   - 上传文件类：PutDemo。
   - 遍历文件类：ListDemo。



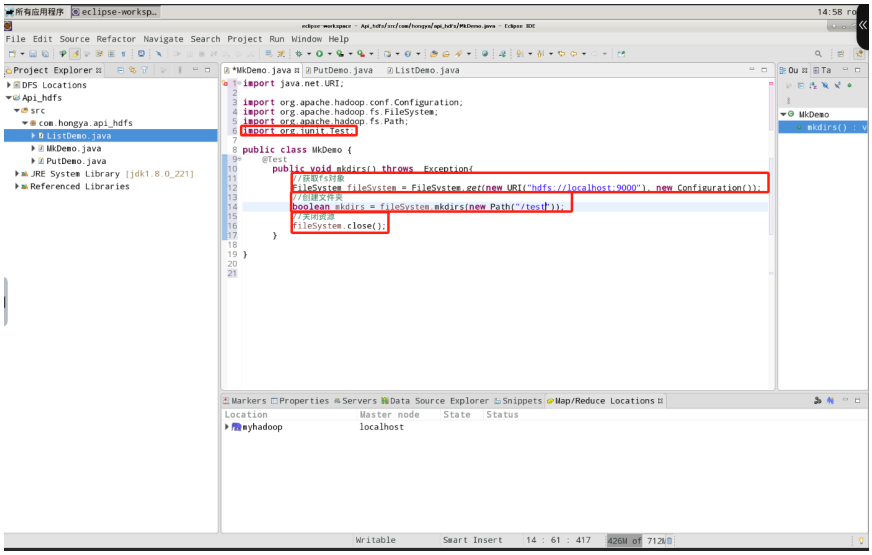
代码逻辑：

1. 在 MkDemo 类中使用单元测试方法。
2. 直接创建 FileSystem 对象。

3. 使用 mkdirs 方法在根目录下创建 test 文件夹。

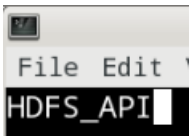
4. 关闭资源。

5. 运行程序并在 Eclipse 中的 DFS Locations 列表栏查看文件夹创建是否成功。



6.2.2 任务 2：上传文件

在本地/root 目录下创建 test.txt 文件。添加如下内容：
HDFS_API



代码逻辑：

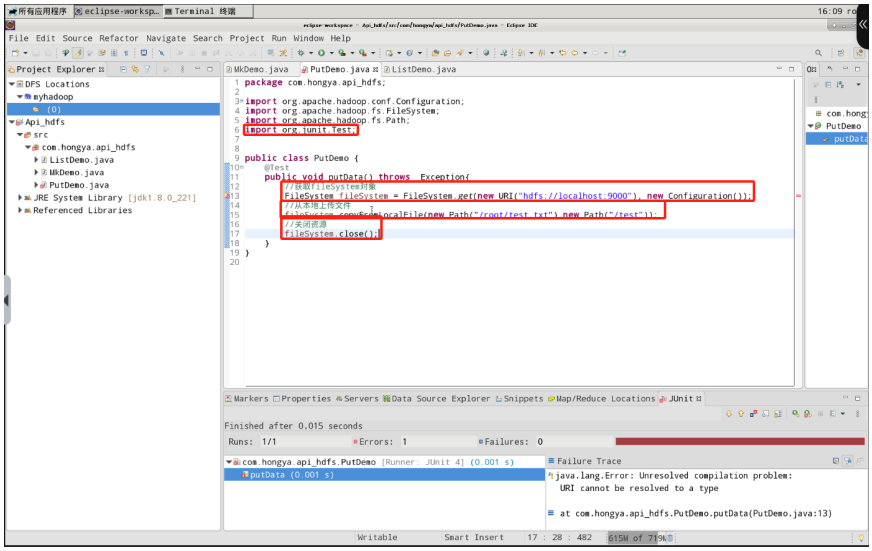
1. 在 PutDemo 类中使用单元测试方法。

2. 创建 FileSystem 对象。

3. 使用 copyFromLocalFile 方法上传文件 test.txt。

4. 关闭资源。

5. 运行程序并在 Eclipse 中的 DFS Locations 列表栏查看文件上传是否成功。



6.2.3 任务 3：遍历文件

遍历文件代码逻辑：

1. 在 ListDemo 类中使用单元测试方法。
2. 获取 FileSystem 对象。
3. 通过 listFiles 方法获取 RemoteIterator 得到所有的文件或者文件夹。
4. 递归遍历并打印文件夹或文件路径。
5. 关闭资源。
6. 关闭资源。

6.3 修改删除文件操作

6.3.1 下载文件

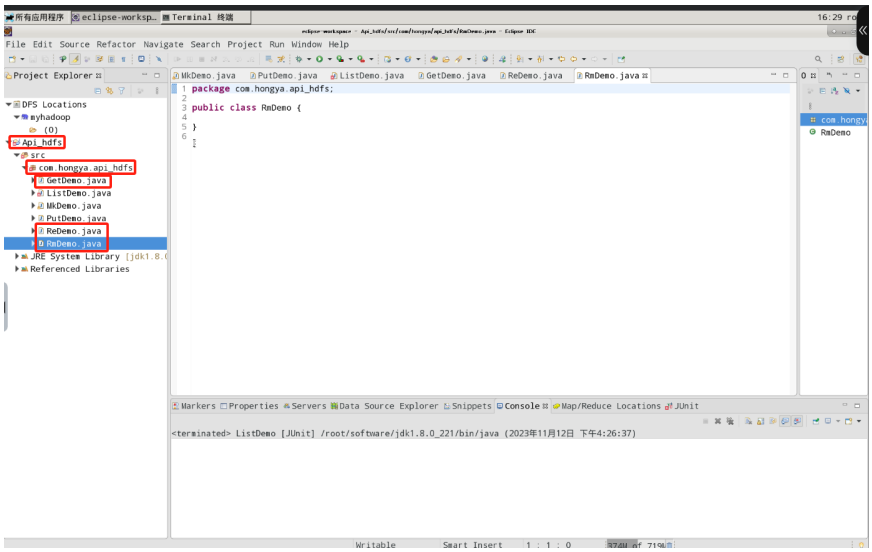
创建项目：

1. 项目名：Api_hdfs。

2. 包名：com.hongya.api_hdfs 。

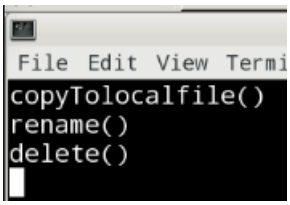
3. 类：

   - 下载文件类：GetDemo。
   - 重命名目录文件类：ReDemo。
   - 重命名目录文件类：ReDemo。



数据准备：

1. 在本地/目录下创建 test.txt 文件，添加内容如下：
   copyTolocalfile()
   rename()
   delete()

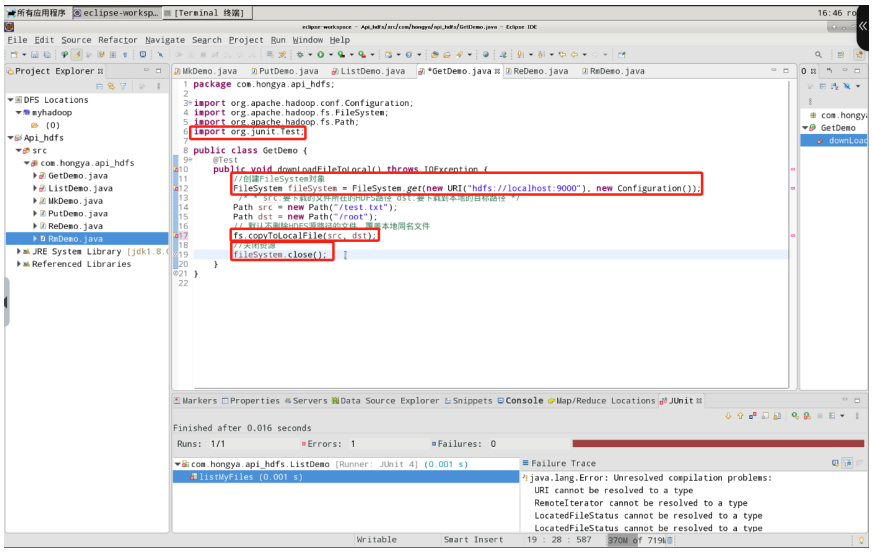


2. 创建并上传 test.txt 文件到 HDFS 目录：/test
   需求：将 HDFS 文件 test.txt 下载到本地/root 目录。

```
→ / hadoop fs -put test.txt /test
→ /
```

下载文件代码逻辑：

1. 在 GetDemo 类中使用单元测试方法。
2. 直接创建 FileSystem 对象。
3. 使用 copyTolocalfile 方法将 test.txt 文件下载到/root。
4. 关闭资源。



查看结果：进入/root 目录通过 ls 或 ll 命令查看 test.txt 文件是否成功下载。

6.3.2 任务 2：重命名目录文件

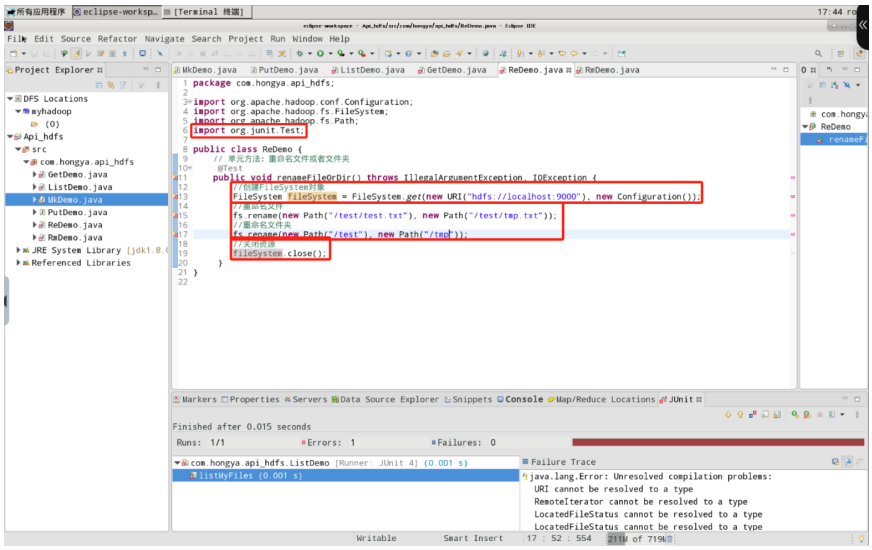
需求 1：将 test.txt 文件重命名为 tmp.txt。
需求 2：将 test 文件夹重命名为 tmp。

代码逻辑：

1. 在 ReDemo 类中使用单元测试方法。
2. 直接创建 FileSystem 对象。

3. 使用 rename 方法重命名目录文件。

4. 关闭资源。



6.3.3 任务 3：删除目录文件

需求 1：删除任务二中修改后文件 tmp.txt。

需求 2：删除任务二中修改后文件夹 tmp。

代码逻辑：

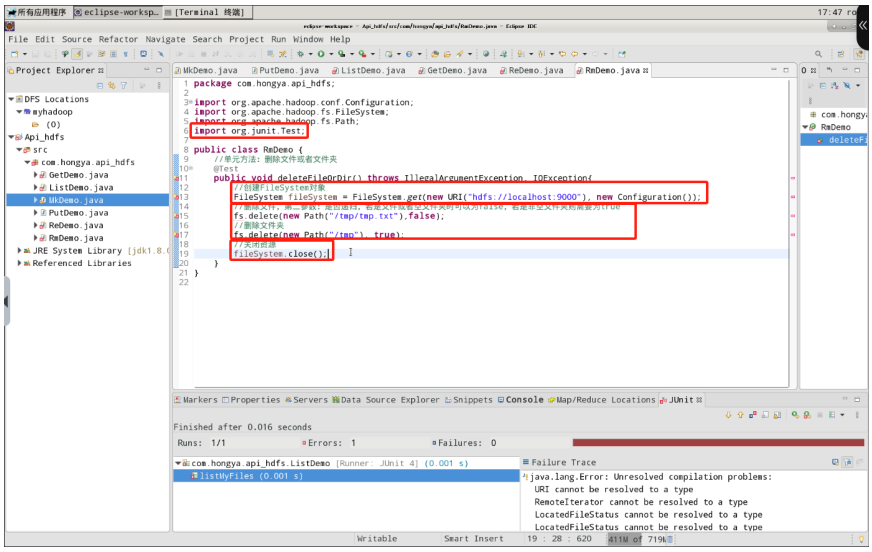
1. 在 RmDemo 类中使用单元测试方法。

2. 直接创建 FileSystem 对象。

3. 使用 delete 方法删除目录文件。

4. 关闭资源。

5. 运行程序并在 Eclipse 中的 DFS Locations 列表栏查看目录文件删除是否成功。

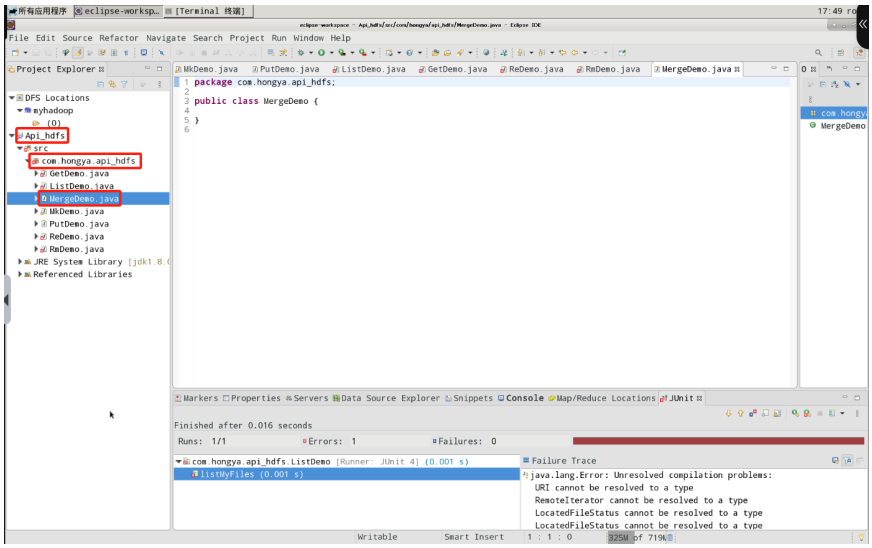


6.3.4 小文件合并

任务：合并小文件操作

创建项目：

1. 项目名：Api_hdfs。
2. 包结构：com.hongya.api_hdfs。
3. 合并文件实现类：MergeDemo。



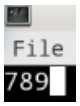
数据准备：

1. 在本地/root 目录下创建目录 input。

2. 在 input 目录下创建 a.txt 文件，添加如下内容：
   123

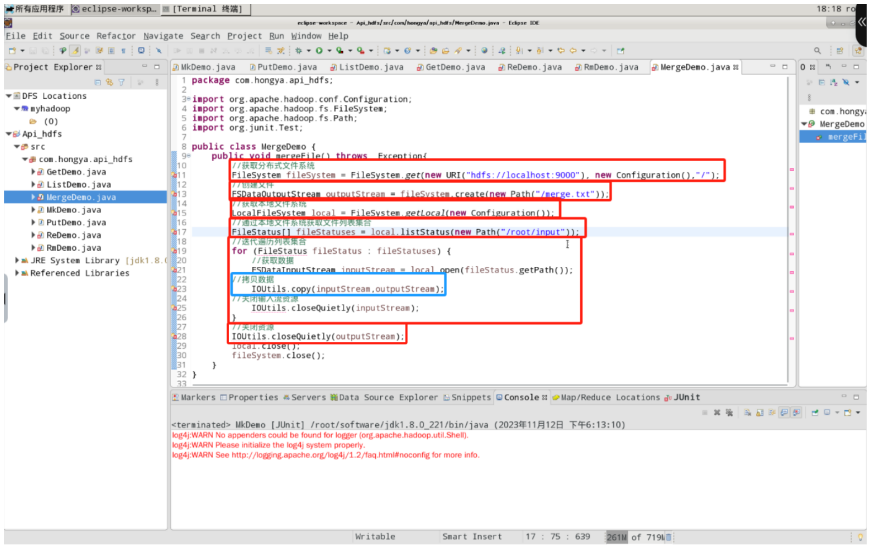
3. 在 input 目录下创建 b.txt 文件，添加如下内容：
   456

4. 在 input 目录下创建 c.txt 文件，添加如下内容：
   789

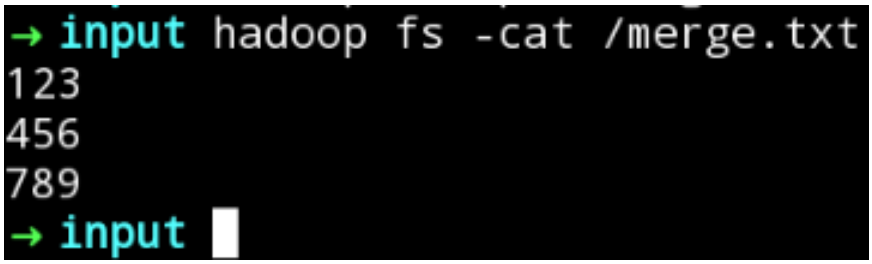


代码逻辑：

1. 获取 HDFS 文件系统对象 FileSystem。
2. 在 HDFS 根目录创建 merge.txt 文件。
3. 获取本地文件系统。
4. 获取本地文件系统文件列表集合。
5. 迭代遍历文件列表获取数据并进行数据拷贝。
6. 关闭资源。



查看结果：通过命令查看 HDFS 根目录下的 merge.txt 文件。



7 实验结果截图

8 困难解决

本次实验较为简单，没有遇到困难。

9 心得体会

做完本次实验，除了掌握了实验目的部分中所有内容的收获之外，我还有以下几点心得体会：

- 实践并掌握了 HDFS 中用 Java API 进行创建目录和上传、遍历、下载合并文件等操作；
- 对比分析了 HDFS 中用 Java API 和 Shell 命令对文件进行上传查看、修改删除等操作的异同点。