

全部课程 (/courses/) / HIVE教程 (/courses/38) / Hive 接口介绍 (Web UI/JDBC)

在线实验，请到PC端体验

Hive 接口介绍

(沒看)

一、实验介绍

1.1 实验内容

>

本次实验学习 Hive 的两种接口：Web UI 以及 JDBC。

1.2 实验知识点

- eclipse开发
- Hive 网络接口 (Web UI)
- jdbc

1.3 实验环境

- Hive V2.0.0
- hadoop2.6.0
- Xfce终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有hadoop基础的用户，熟悉linux基础知识

二、Hive 网络接口 (Web UI)

之前的实验中，我们使用的是 Hadoop V2.4.1版本。我们在测试的时候，发现 Hive V1.1.0 在进行 Session 查询的时候需要 Hadoop V2.6 以上版本，因此本次实验的 Hadoop 版本基于 V2.6.0 版本。由于已经修改 Hadoop 目录、Hive 目录为普通权限，如果提示 hadoop 或 hive 为无效命令，请 `source /etc/profile` 来使配置文件生效。

Hadoop 目录为：`/usr/local/hadoop-2.6.0`

Hive 目录为：`/usr/local/hive-1.1.0`

2.1 Web UI简介

Hive Web UI 提供了图像化的操作界面，通过 Hive Web UI 接口可以更方便、更直观地操作，特别是对刚刚接触 Hive 的用户来说。Hive Web UI 具有一下特性：

分离查询的执行

在命令行 (CLI) 下，要执行多个查询就得打开多个终端，而通过Web UI，就可以同时执行多个查询，还可以在网络服务器上管理会话 Session。

不依赖本地 Hive

用户需要安装本地 Hive，就可以通过网络浏览器访问 Hive 并进行相关操作。如果想通过 Web 与 Hadoop 以及 Hive 交互，那么需要访问多个端口。

动手实践是学习 IT 技术最有效的方式！

开始实验

2.2 配置 hive-site.xml

修改 \$HIVE_HOME/conf 目录下的 hive-site.xml 文件。

我们可以看一下默认配置是什么：

```

1240 <property>
1241   <name>hive.hwi.listen.host</name>
1242   <value>0.0.0.0</value>
1243   <description>This is the host address the Hive Web Interface will l
n on</description>
1244 </property>
1245 <property>
1246   <name>hive.hwi.listen.port</name>
1247   <value>9999</value>
1248   <description>This is the port the Hive Web Interface will listen on
scription>
1249 </property>
1250 <property>
1251   <name>hive.hwi.war.file</name>
1252   <value>${env:HWI_WAR_FILE}</value>
1253   <description>This sets the path to the HWI war file, relative to ${
_HOME}. </description>
1254 </property>

```

我们只需要修改 hive.hwi.war.file，你应该先在 \$HIVE_HOME/lib 目录下检查一下 hwi 的版本：

```

hadoop@0ec129d69802:
hive-contrib-1.1.0.jar
hive-exec-1.1.0.jar
hive-hbase-handler-1.1.0.jar
hive-hwi-1.1.0.jar
hive-jdbc-1.1.0.jar
hive-jdbc-1.1.0-standalone.jar

```

结果令人吃惊的是，居然没有那个 war 文件！

只有一个同名的 jar 包，没有 war 文件。解决办法是自行下载对应 Hive 的源码包再打包成 war 文件。

下载 Hive 源码（注意这里是 src 包，不是 bin 包。上次我们使用的是 1.1.0 版本）

```
$ wget http://labfile.oss.aliyuncs.com/apache-hive-1.1.0-src.tar.gz
```

解压：

```
$ tar zxvf apache-hive-1.1.0-src.tar.gz
```

再进入 hwi 目录，打包 war 文件（注意命令末尾有一个点 .）：

```
$ cd apache-hive-1.1.0-src/hwi
$ jar cvfM0 hive-hwi-1.1.0.war -C web/ .
```

打包完成后，有了我们需要的 war 文件，再复制到 \$HIVE_HOME/lib 目录下：

```
$ cp hive-hwi-1.1.0.war /usr/local/hive-1.1.0/lib
```

动手实践是学习 IT 技术最有效的方式！

开始实验

另外我们还需要拷贝一个 Java 的 tools.jar 到 \$HIVE_HOME/lib 目录下：

```
$ cp /usr/local/jdk1.7.0_67/lib/tools.jar /usr/local/hive-1.1.0/lib
```

否则会出现类似于下面的错误 (因为 JAVA_HOME 指到 \$JAVA_HOME/jre 下了, 而其 lib 下的 tools.jar 跟 \$JAVA_HOME/lib/tools.jar 不一样, 编译的时候需要用到后者) :

```
hadoop@0ec129d69802: /usr/local/hadoop/hive
15/03/27 09:06:27 ERROR mortbay.log: /hwi/
Unable to find a javac compiler;
com.sun.tools.javac.Main is not on the classpath.
Perhaps JAVA_HOME does not point to the JDK.
It is currently set to "/usr/local/jdk1.7.0_67/jre"
```

最后, 我们将 hive-site.xml 文件修改为 :

```
<property>
  <name>hive.hwi.war.file</name>
  <value>/lib/hive-hwi-1.1.0.war</value>
  <description>This sets the path to the HWI war file, relative to ${HIVE_HOME}. </description>
</property>
```

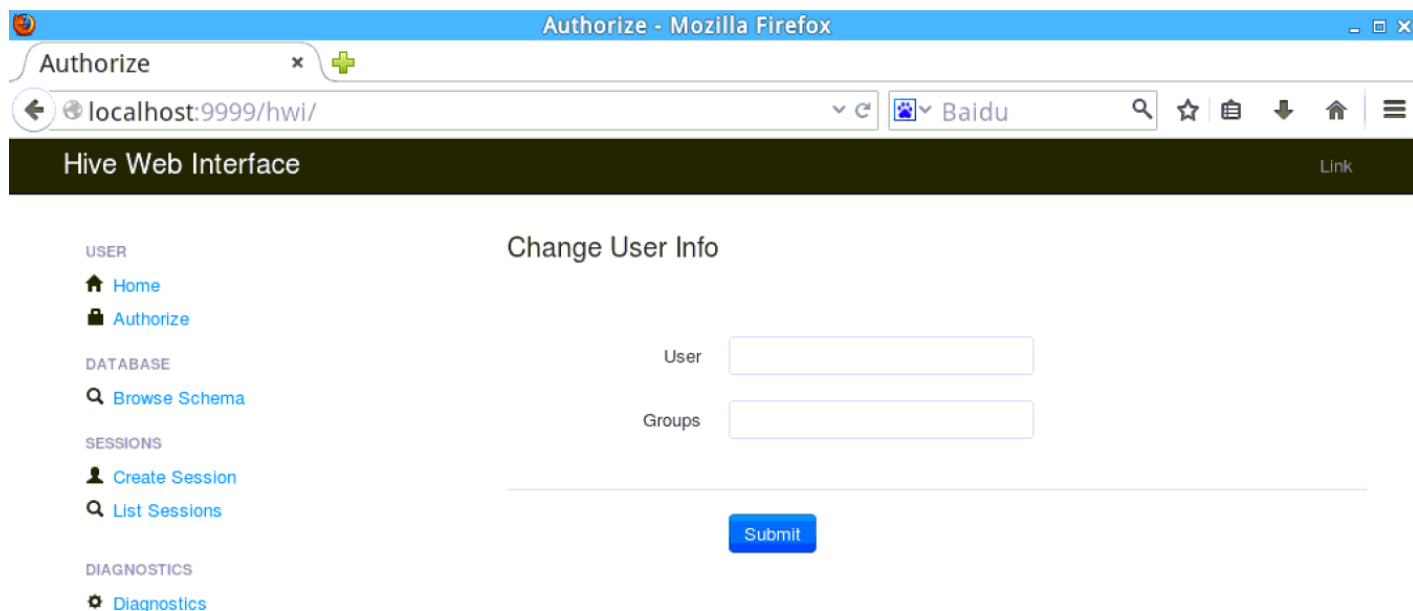
2.3 启动 hwi

在 \$HIVE_HOME/bin 目录下, 启动 hwi (由于我们之前已经修改了 Derby 为 MySQL 数据库, 所以在启动 hwi 之前, **请确保 MySQL 和 Hadoop 已经成功启动**) :

```
$ hive --service hwi
```

```
shiyancelou@0ec129d69802: /usr/local/hive-1.1.0/bin
$ hive --service hwi [9:35:39]
15/04/07 01:35:43 INFO hwi.HWIServer: HWI is starting up
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hive-1.1.0/lib/hive-jdbc-1.1.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
15/04/07 01:35:45 INFO mortbay.log: Logging to org.slf4j.impl.Log4jLoggerAdapter (org.mortbay.log) via org.mortbay.log.Slf4jLog
15/04/07 01:35:45 INFO mortbay.log: jetty-6.1.26
15/04/07 01:35:45 WARN mortbay.log: Can't reuse /tmp/Jetty_0_0_0_0_9999_hive.hwi.1.1.0.war__hwi__.4otk49, using /tmp/Jetty_0_0_0_0_9999_hive.hwi.1.1.0.war__hwi__.4otk49_7954619298025443480
15/04/07 01:35:45 INFO mortbay.log: Extract /usr/local/hive-1.1.0/lib/hive-hwi-1.1.0.war to /tmp/Jetty_0_0_0_0_9999_hive.hwi.1.1.0.war__hwi__.4otk49_7954619298025443480/webapp
15/04/07 01:35:46 INFO mortbay.log: Started SocketConnector@0.0.0.0:9999
```

现在, 我们可以在浏览器中打开网络接口的地址: localhost:9999/hwi, 启动成功:



2.4 Web UI操作实例

数据库及表信息查询

在查询之前，我们先新建一个 person表 以方便后续的查询操作。启动 Hive 后，输入如下命令（这个 table 只是一个示例，你可以根据需要自己创建其他的）：

```
hive> create table person(name STRING,age INT) ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t' STORED AS TEXTFILE;
OK
Time taken: 1.072 seconds
hive>
```

新建一个 txt 文件（路径自定义，比如我是 /home/hadoop/hive/person.txt），写一些示例数据，数据之间是以 \t 划分的；实验中已经写好，直接导入即可：

```
person.txt
1 aaa      25
2 test     30
3 cc       34
```

再把数据导入到 person表 中：

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/hive/person.txt' OVERWRITE INTO TABLE
person;
Loading data to table default.person
Table default.person stats: [numFiles=1, numRows=0, totalSize=33, rawDataSize=0]
OK
Time taken: 1.606 seconds
```

单击 Browse Schema 可以查看当前 Hive 中的数据库，显示的是当前可以使用的数据库信息，只包含一个默认数据库（default）：

Database List

default

再单击 default 就可以看到 default 数据库中包含的所有表的信息了，这里我们刚刚新建的 table：person表。

动手实践是学习IT技术最有效的方式！

开始实验

Hive Web Interface

USER

[Home](#)[Authorize](#)

DATABASE

[Browse Schema](#)

default Table List

Name: default

Description: Default Hive database

[person](#)

点击 person 表即可看到具体信息。

person

ColsSize: 2

Input Format: org.apache.hadoop.mapred.TextInputFormat

Output Format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

Is Compressed?: false

Location: hdfs://localhost:9000/user/hive/warehouse/person

Number Of Buckets: -1

Field Schema

Name	Type	Comment
name	string	null
age	int	null

Hive Session 查询

在进行查询之前我们需要新建一个 Session (注意每次重启hwi后, 之前的 Session 将会失效)。点击 Create Session 来新建 :

USER

[Home](#)[Authorize](#)

DATABASE

[Browse Schema](#)

SESSIONS

[Create Session](#)[List Sessions](#)

DIAGNOSTICS

[Diagnostics](#)

Change User Info

User





Groups

填入以下信息, 打框的必填 :

动手实践是学习 IT 技术最有效的方式 !

开始实验

Manage Session test

 History  Diagnostics  Remove  Result Bucket

Session Details

Result File

Error File

Query

Silent Mode

Start Query

Query Return Codes

Submit

其中：

Result File 为最后的查询结果文件；

Query 为你的查询语句；

Start Query 设为 YES，表示开始查询；

点击 Submit 提交后，即可看到 View File：

Result File [View File](#)

点击 View File 即可看到最后的查询结果：

Hive Web Interface

resultFile

			aaa	25	NULL
test	30	NULL			
cc	34	NULL			

This file contains 0 of 1024 blocks. [Next Block](#)

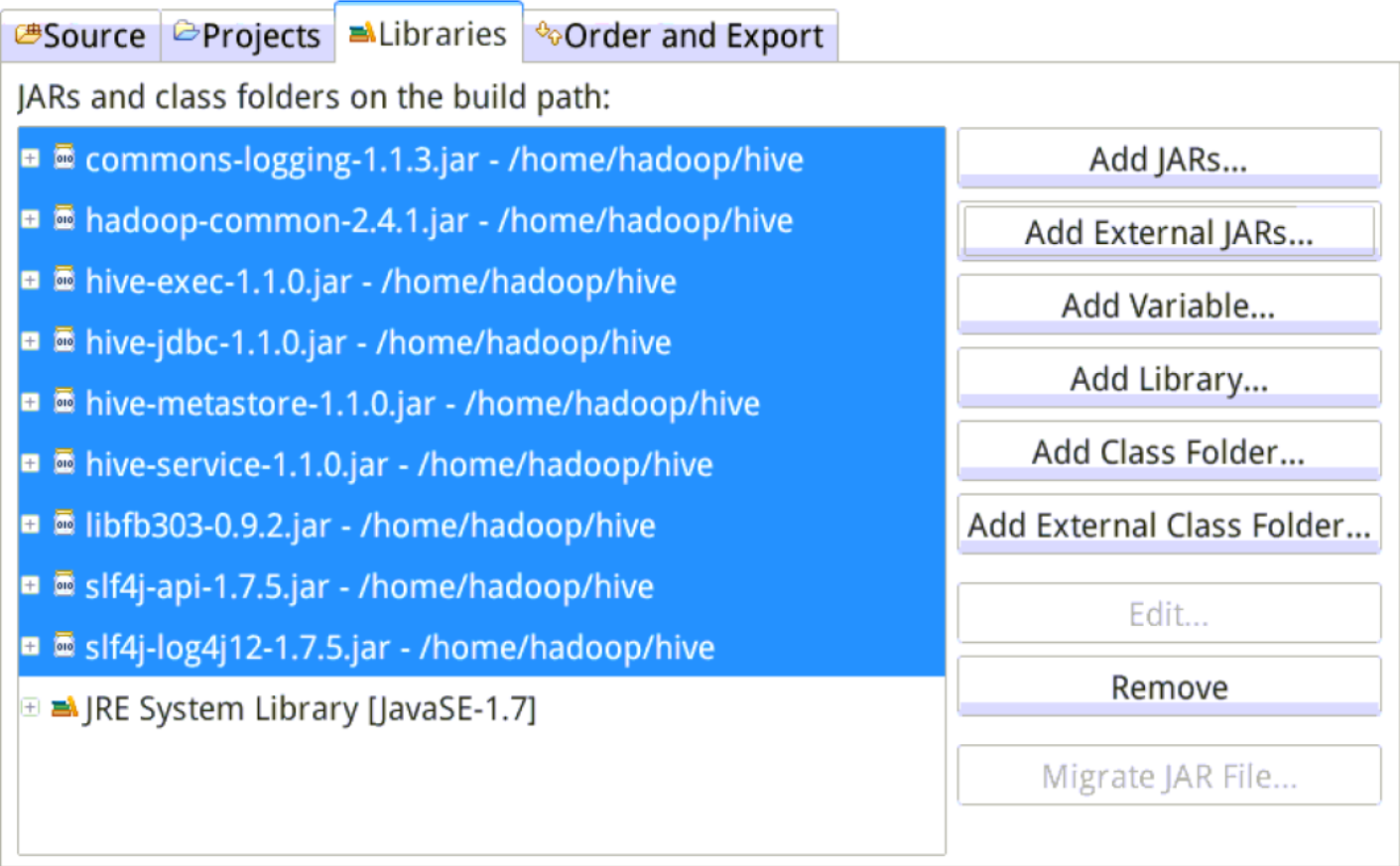
动手实践是学习 IT 技术最有效的方式！

开始实验

通过以上的学习，我们可以了解到 Web UI 虽然提供了可视化的查询操作，但最大的缺点就是用户不能实时了解查询的状态，交互能力较差。

三、JDBC

在 Eclipse 中新建一个 Java 工程，例如 HiveJdbc. 然后添加所需要的 jar 包，右击工程，依次选择 Properties -> Java Build Path -> Libraries -> Add External Jars 来添加。所需 jar 包如下（如不清楚你可以直接添加所有的 jar 包；遇到没有权限添加的，请拷贝 jar 包到其他有权限的目录再添加，实验中已经拷贝到 /home/hadoop/hive ）：



新建包和类，添加如下代码：

```

import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveJdbc {

    private static String driverName =
        "org.apache.hive.jdbc.HiveDriver";

    public static void main(String[] args)
        throws SQLException {
        try {
            Class.forName(driverName);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            System.exit(1);
        }

        Connection con = DriverManager.getConnection(
"jdbc:hive2://localhost:10000/default", "", "");
        Statement stmt = con.createStatement();
        String tableName = "hive_jdbc";
        stmt.execute("drop table if exists " + tableName);
        stmt.execute("create table " + tableName +
            " (key int, value string)");
        System.out.println("Create table success!");
        // show tables
        String sql = "show tables '" + tableName + "'";
        System.out.println("Running: " + sql);
        ResultSet res = stmt.executeQuery(sql);
        if (res.next()) {
            System.out.println(res.getString(1));
        }

        // describe table
        sql = "describe " + tableName;
        System.out.println("Running: " + sql);
        res = stmt.executeQuery(sql);
        while (res.next()) {
            System.out.println(res.getString(1) + "\t" + res.getString(2));
        }

        sql = "select * from " + tableName;
        res = stmt.executeQuery(sql);
        while (res.next()) {
            System.out.println(String.valueOf(res.getInt(1)) + "\t"
                + res.getString(2));
        }

        sql = "select count(1) from " + tableName;
        System.out.println("Running: " + sql);
        res = stmt.executeQuery(sql);
        while (res.next()) {
            System.out.println(res.getString(1));
        }
    }
}

```

从上述代码可以看出，在进行查询之前需要做的工作有：

- 通过 `Class.forName("org.apache.hive.jdbc.HiveDriver")`；来注册 Hive 驱动；
- 通过 `Connection con = DriverManager.getConnection(
"jdbc:hive2://localhost:10000/default", "", "");`；来与 Hive 数据库建立连接；

运行之前需要启动 hiveserver，但目前 Hive 摒弃了 hiveserver 转而改用了 hiveserver2，反映在代码中有两点不同。一是，driverName 由 `org.apache.hadoop.p.hive.jdbc.HiveDriver` 改成了 `org.apache.hive.jdbc.HiveDriver`；二是，Connection 由 `jdbc:hive://localhost:10000/default` 改成了 `jdbc:hive2://localhost:10000/default`。

动手实践是学习 IT 技术最有效的方式！

开始实验

因此，在运行程序前启动 hiveserver 也改成了启动 hiveserver2，在 \$HIVE_HOME/bin 目录下启动。

```
hadoop@0ec129d69802:/usr/local/hive-1.1.0/bin$ ls
beeline      ext      hive-config.sh  metastore_db  schematool
derby.log    hive     hiveserver2     metatool
hadoop@0ec129d69802:/usr/local/hive-1.1.0/bin$
```

最后程序运行结果如下：

```
log4j:WARN No appenders could be found for logger (org.apache.hive.jdbc.Utils).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hive/hive-jdbc-1.1.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Create table success!
Running: show tables 'jdbc_test'
jdbc_test
Running: describe jdbc_test
key int
value string
Running: select count(1) from jdbc_test
```

四、作业

Web UI 和 JDBC 两种接口有什么异同？

五、参考文档

- 《Hadoop实战 第2版》陆嘉恒，机械工业出版社；
- Hive Web接口HWI的操作及使用 (<http://www.ithao123.cn/content-648234.html>)；
- hive jdbc 调用 (<http://www.bubuko.com/infodetail-641646.html>)；

< 上一节 (/courses/38/labs/775/document)

课程教师



牧云Melanie

共发布过2门课程

[查看老师的所有课程 > \(/teacher/225160\)](#)

前置课程

Hadoop部署及管理 (/courses/35)

《Hadoop权威指南》配套实验 (/courses/222)

进阶课程

HBASE教程 (/courses/37)

Mahout教程 (/courses/39)



动手做实验，轻松学动手实践是学习 IT 技术最有效的方式！

[开始实验](#)