**# DistributedKeyValueStore Project - 1**

-----Distributed Key Value storage system based on Apache Thrift-----

**Refrenced Libraries :**

Thrift - 0.9.3

SLF4J - 1.7.21

**To compile** just type:

javac -cp ".:./Referenced\_Libraries/slf4j-simple-1.7.21.jar:./Referenced\_Libraries/slf4j-api-1.7.21.jar:./Referenced\_Libraries/libthrift-0.9.3.jar" kvstore/\*.java

OR

You can also compile by Executing the shell script `./KVCompile\_Server\_Client`.

**To run** **server** just type:

java -cp ".:./Referenced\_Libraries/slf4j-simple-1.7.21.jar:./Referenced\_Libraries/slf4j-api-1.7.21.jar:./Referenced\_Libraries/libthrift-0.9.3.jar" kvstore.kvserver

Server will run by default on localhost, port 9091.

**To run client** just type:

java -cp ".:./Referenced\_Libraries/slf4j-simple-1.7.21.jar:./Referenced\_Libraries/slf4j-api-1.7.21.jar:./Referenced\_Libraries/libthrift-0.9.3.jar" kvstore.kvclient -Arg1 -Arg2 etc.

OR

You can also run the client by Executing the shell script `./kvcient -arg1 -arg2 etc.`.

Client will timeout after 5 seconds if its not able to connect to server.

You can connect the client to a different server by passing command line arguments.

The client has three subcommands:

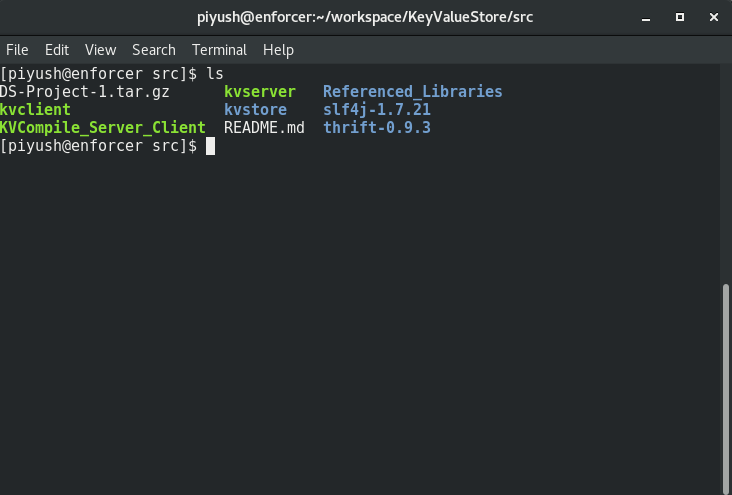
get - Example: ./kvclient -server host:port -get 'my\_key' > my\_value\_file

set - Example: ./kvclient -server host:port -set 'my\_key' 'my\_value'

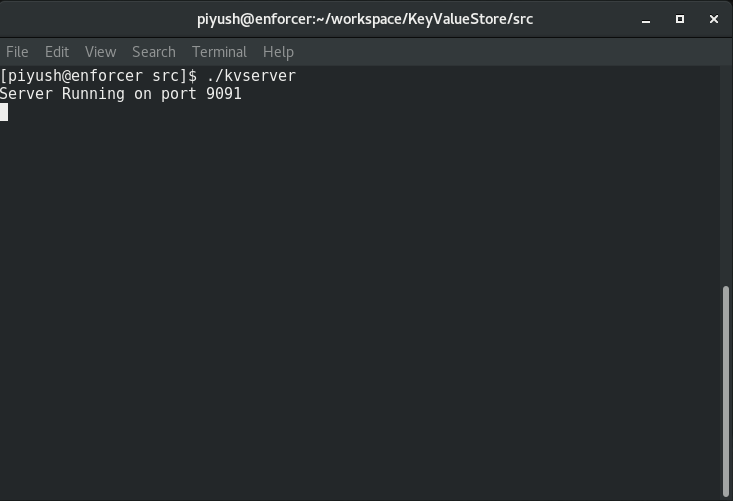
del - Example: ./kvclient -server host:port -del 'my\_key'

The client and server are built using **java** and you just need to run the `kvclient` script to run the client with appropriate arguments.

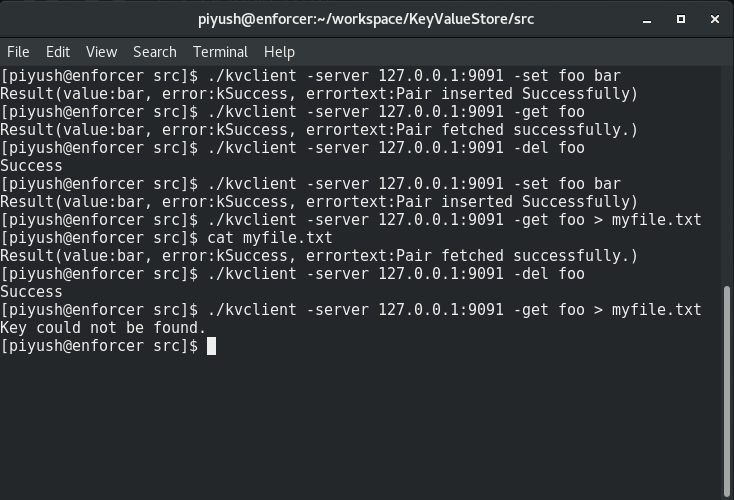
**Screenshots**



**Source Directory**



**Server running**



**Basic operations**