

Distributed Systems 2016 Project 1

Due date: October 17th, 2016. Implement a server and a client for the service defined in kvstore.thrift. The server should be called kvserver. The client should be called kvclient and should run on the command line. The client should have three subcommands:

1. **get** - Example: `./kvclient -server host:port -get 'my_key' > my_value_file`
2. **set** - Example: `./kvclient -server host:port -set 'my_key' 'my_value'`
3. **del** - Example: `./kvclient -server host:port -del 'my_key'`

Note that the get subcommand should write the value to [stdout](#). The kvclient process should have an exit code of zero on success, 1 if a key isn't found, and 2 on other errors. If errortext is returned from the server, kvclient should write it to [stderr](#).

kvstore.thrift

```
namespace cpp kvstore
namespace java kvstore

enum ErrorCode {
    kSuccess = 0,
    kKeyNotFound = 1,
    kError = 2,
}

struct Result {
    // If an RPC returns a value, it is here.
    1: string value,

    2: ErrorCode error,

    // If an RPC returns kError, errortext may contain human-readable text.
    3: string errortext,
}

service KVStore {
    // If a key-value pair already exists, overwrite its value.
    // If a key-value pair does not already exist, create it.
    Result kvset(1:string key, 2:string value),

    // If a key-value pair exists, return its value and kSuccess.
    // If a key-value pair does not exist, return error kKeyNotFound.
    Result kvget(1:string key),

    // If a key-value pair exists, delete it and return kSuccess.
    // If a key-value pair does not exist, return kKeyNotFound.
    Result kvdelete(1:string key),
}
```

