

Issuing RPCs

TABLE OF CONTENTS

- 1 [Example](#)
 - 2 [Calling ECHO](#)
-

You have now fully implemented the `ECHO` RPC. You will now call it from a client.

Example

An example of issuing an RPC is shown in the `example` function in `kv_store_client.c`:

```
int example(int input) {
    CLIENT *clnt = clnt_connect(HOST);

    int *result;

    result = example_1(&input, clnt);
    if (result == (int *)NULL) {
        clnt_perror(clnt, "call failed");
        exit(1);
    }
    int ret = *result;
    xdr_free((xdrproc_t)xdr_int, (char *)result);

    clnt_destroy(clnt);

    return ret;
}
```

The function first connects to the RPC server, then issues the RPC by calling the `example_1` client stub. It then recovers the result by dereferencing the returned pointer, frees the memory allocated by the

RPC, closes the client, and returns the result.

Calling ECHO

Now implement the `echo` function in `kv_store_client.c` so that it calls the RPC you just implemented. You should ensure that you are making the call to the server rather than just returning the input locally (you will not pass the autograder tests otherwise).

We recommend copying the implementation of `example` and changing the necessary types to make sure you are following all of the necessary steps. Keep in mind that you will need to duplicate the returned string using `strdup` before freeing the XDR structure with `xdr_free((xdrproc_t)xdr_string, (char*)result)`.

Compile your code with `make`. You should now be able to test your `ECHO` RPC by starting the server with `./server` and calling the function you just implemented with `./client echo test`. You should see the output `test`.