

# ThriftErlSkel

## Introduction

[ThriftErlSkel](#) is a nice sidekick for creating Thrift services in Erlang. A script created by Todd Lipcon who also created the Erlang bindings for Thrift, [ThriftErlSkel](#) provides a programming environment as using makefiles, and skeleton-code for your service.

## Features

- Generates server stubs -
- Hot-swapping code - no need to restart the service

## Getting started / Tutorial

Download this:

[http://github.com/toddlipcon/thrift\\_eri\\_skel/tree/master](http://github.com/toddlipcon/thrift_eri_skel/tree/master)

First thing you want to do is execute that perl script so do: `./make_new_thrift.pl example Example 9090`

- `example` - This is name of your thrift file without the extension, a thrift file should be in the same directory as the perl script
- `Example` - Probably the class level name
- `9090` - port feel free to change it

Make sure you've taken the `lib/erl/` directory in the thrift source code and placed it in the same directory as the `thrift_skel`. Run `make` on it, we need it compiled for this to work.

Next make an `rpc` directory somewhere, I made mine a directory \*above\* where my `thrift_skel`, `thrift` directories are. Call it whatever you want, I called mine `/rpc/`. Now you want to edit `/example/gen/Makefile`. And change the path of `RPC_DIR` to where you actually have it or you will definitely run into errors.

Now it's fun time, go into your `example/src/` and start editing your `example_service.erl` with any function you want, remember to export it. A simple one would be: `add(A,B) -> A+B`. The export at the top of the file below `handler_function` one would be: `add/2`

Jump back into root directory of your example folder, not the `src` folder. Run `make`. Everytime you change code, you must run `make` but you do not have to restart your service, Todd made it awesome with code reloading, we'll go into that.

Now run: `sudo ./start_example.sh` and the service should start running with a pseudo console for you to reload code and even test your own service. Press enter and type: `example_service:add(1,3)`.

Hopefully it works and you see the output of the result.