

Hi Luigi,

it is not skylark.

It is a very simple single process multi threaded server.

It binds to a port, starts a fixed number of threads to serve the incoming requests and parses headers with `HttpHeader` class.

After that it finds out the endpoint from the URL.

There is a macro generated map with the registered endpoints and based on them, it calls a macro generated function, that parses the HTTP body into the struct, that is the argument of that specific endpoint method.

After that it calls that method within a server class with the deserialized argument.

I use `Jsonize` here.

Then the method does whatever it wants and returns an other struct.

Then the macro generated code serializes it to JSON and sends it back to the client.

So yes, the U++ client communicates with this remote U++ (HTTP + JSON) server.

On client side, the code is just calling the same methods with the `jsonizeable` structs and a macro generated code does the rest.

The invoicing software can also run with a local `sqlite` or `postgresql` database. For this case there are other macros, generating the same methods in a class, that calls the endpoint code directly as methods.

Implementing the server and client side code is very easy this way

You just have to write normal methods on server side and call the methods with the same name on client side. And the new `Assist` is helping you also.

There are some more complex things in it also. For example the exceptions thrown in the server code are propagated to the client side, so the error handling is quite simple as well.

There is a macro, generating the callable javascript functions also.

This "framework" is similar to the database framework of U++.

I run the server side U++ daemon behind `Nginx` on Linux.

---