



# Calling RPC methods in your module or module client

## Receiving RPC method calls

When receiving RPC method calls, the arguments are passed as called with one addition: The first argument is always prepended by a `Struct` with [verified metadata](#). This information can be used to identify the user for example or to get the apartment the user lives in. See the chapter about [verified metadata](#) for more information.

## Calling RPC methods from your UI or other UIs

You can call any RPC method implemented by your module, by sending a JSON-RPC request to the following path of `c1-core` :

```
/modules/<module ID>/rpc
```

If you want to call the RPC method on a client of your module, use:

```
/module_clients/<module client ID>/rpc
```

## Ping

There is a special path provided by `c1-core` :

```
/modules/<moduleI ID>/ping
```

You can send a GET request to this path to check, if a module is reachable. When using `libc1-module` or `c1-module-proxy`, the `ping` request is handled by them. Otherwise the RPC method `ping` needs to be implemented.

## Command line examples

Here's a call to ping on an C1 Proxy

```
curl \
  --cacert /etc/c1-core/coreCaCert.pem \
  -H "PRIVATE-TOKEN: <your private token>" \
```

```
-d '{"id":12,"method":"ping","params":[]}' \
"https://core.sensaru.net/api/v1/modules/c1-proxy/rpc"
```

Here's a call to getDeviceDescription on an Edge Client

```
curl \
--cacert /etc/c1-core/coreCaCert.pem \
-H "PRIVATE-TOKEN: <your private token>" \
-d '{"id":12,"method":"getDeviceDescription","params":[3]}' \
-X POST \
"https://core.sensaru.net/api/v1/module_clients/c1-proxy_d1faa8d0-2db4-11ea-af75-674069e60b74_1005.1.1_1/rpc"
```

## Calling RPC methods from other modules

See [Inter-module RPC usage](#).