

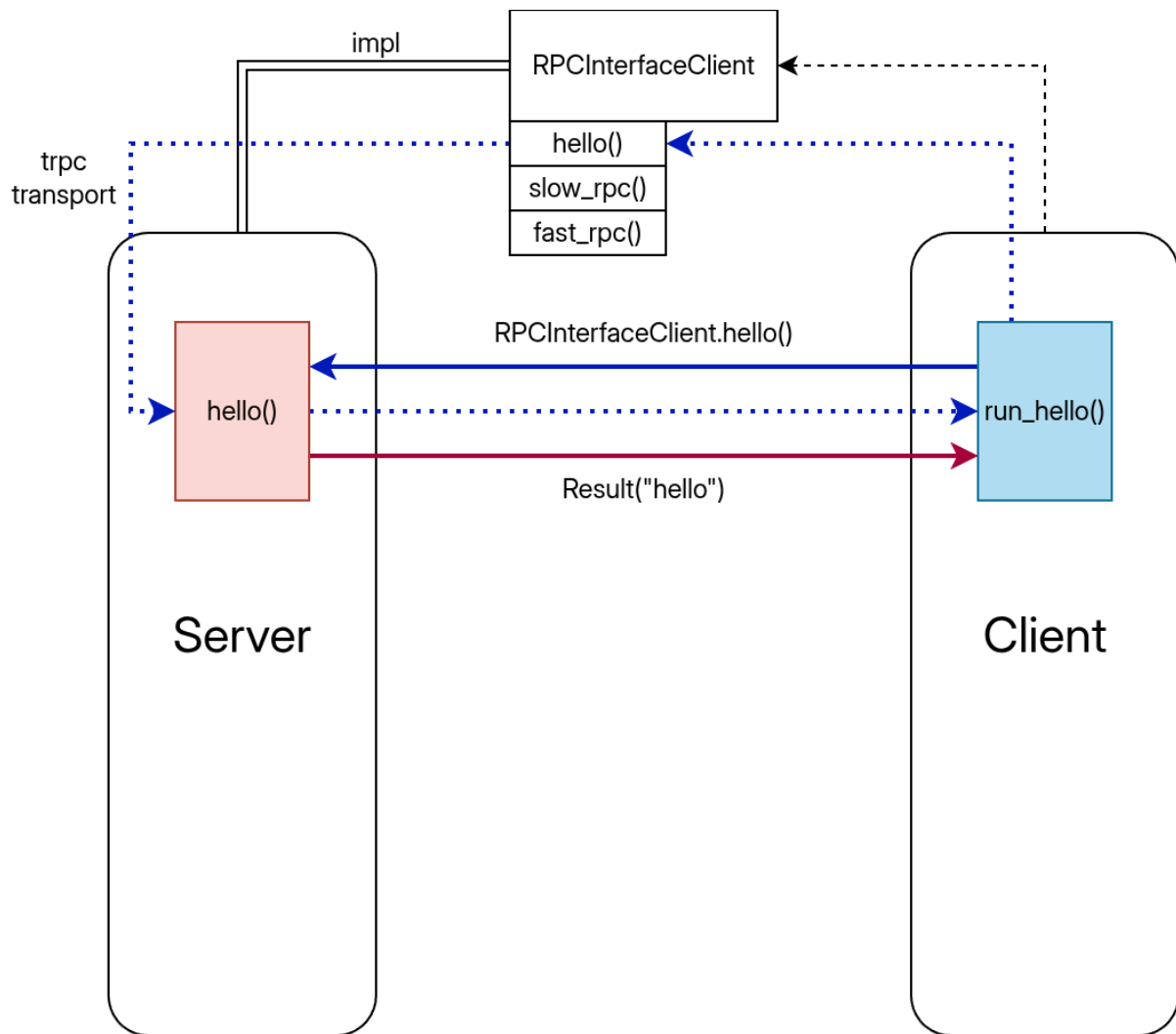
# DS 210 Discussion 8 - Mini Project 3 P2

Instruction: [https://rust4ds.github.io/ds210-sp26-a1/projects/project03\\_analytics.html](https://rust4ds.github.io/ds210-sp26-a1/projects/project03_analytics.html)

## 1. Part 2: Build the Client and Server

### 1.1. Folder Structure and Tasks

1. **analytics\_lib**: You implemented this in part 1.
  - `analytics_lib/src/dataset.rs`: Make `Dataset` and other structs **serializable** and **deserializable**, to enable sending them between client and server.
2. **interface**: The public RPC interface.
  - `interface/src/lib.rs`: Add RPC functions.
  - *What is RPC? - RPC is Remote procedure calls. This is a technology that allows a client to invoke a remote function on a server.*
3. **server**: this contains the server code.
  - `server/src/{lib,solution}.rs`: Implement the server logic for each RPC function.
4. **client**: this contains the client code.
  - `client/src/solutions.rs`: Implement the client logic for calling server RPC functions and handle the parameters and returned data.
  - `client/src/bonus.rs`: Optional bonus credit!



## 1.2. Step 0: Try and Get the Starting Code Run

1. Sync your code with the course GitHub repo. Make sure you **branch off** `analytics_lib` from part 1. Do not modify `analytics_lib` during this part!.
  - This is to say, you should sync a branch **other than** the `analytics_lib` branch you have in part 1. For example, it could be your main branch or things like `part2`.
2. Open the `project_3_client_server_analytics` folder in your VSCode.
  - Make sure you use **open folder**, and not open a file.
  - Make sure you open the **entire** `project_3_client_server_analytics` folder.
    - So you have a view of all submodules for this project in your VSCode.
3. We take a look at the **RPC (interface)** part first. Navigate to `interface/src/lib.rs`:



```

pub async fn run_hello(rpc_client: &RPCInterfaceClient) {
    let result: String = rpc_client.hello(ctx: Context::current()).await.unwrap();
    println!("The server says: `{}`", result);
}

pub async fn run_slow_rpc(rpc_client: &RPCInterfaceClient, query: Query) → Dataset {
    println!("using slow_rpc");

    // let dataset = rpc_client.slow_rpc().await.unwrap();
    todo!("Implement this");

    // What should you do to the dataset?
    // Hint: you have not used `query`, maybe you need to use it somehow?
}

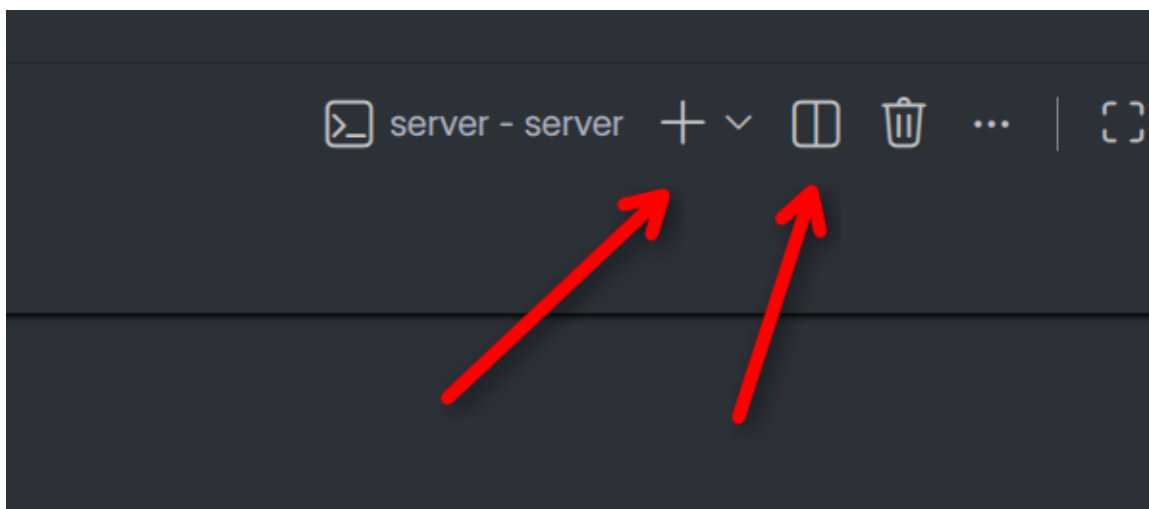
pub async fn run_fast_rpc(rpc_client: &RPCInterfaceClient, query: Query) → Dataset {
    println!("using fast_rpc");

    // You should call fast_rpc here and not slow_rpc.
    todo!("Implement this");
}

```

### 1.3. Give it a try!

1. Open a terminal, navigate to [path to repo]/project\_3\_client\_server\_analytics/server,
  - You can also do that in the VSCode terminal by `cd ./server`, assuming you have the whole project folder opened in your VSCode.
2. Then run `cargo run -- grades`, wait for the program to compile and start running.
  - which starts up your **server**.
3. Open a **second terminal**, navigate to [path to repo]/project\_3\_client\_server\_analytics/client.
  - Do not close/stop the previous terminal! You need both **server** and **client** running for the experiment.
  - In VSCode, you can do that by either



4. Then run `cargo run --bin main -- hello`
5. You should see things like

```
alytics/server on git:main o [11:45:50] C:130
$ cargo run -- grades
14 | pub struct AnalyticsServer {
    | ----- field in this struct
15 |     dataset: &'static Dataset,
    |     ^^^^^^^
    |
= note: 'AnalyticsServer' has a derived impl for the trait 'Clone', but this
is intentionally ignored during dead code analysis
= note: '#[warn(dead_code)]' (part of '#[warn(unused)]') on by default

warning: 'server' (Lib) generated 4 warnings (run 'cargo fix --lib -p server' t
o apply 3 suggestions)
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.06s
Running `~/home/maro/Documents/ds210-sp26-a1-code/project_3_client_server_a
alytics/target/debug/server grades`
hello called

alytics/client on git:main o [11:45:47]
$ cargo run --bin main -- hello
28 | pub async fn run_fast_rpc(rpc_client: &RPCInterfaceClient, query: Query) -
    | -----
    | Dataset {
    |     ^^^^^ help: if
    |     this is intentional, prefix it with an underscore: `'_query'`
    |
warning: 'client' (Lib) generated 5 warnings (run 'cargo fix --lib -p client' t
o apply 5 suggestions)
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.05s
Running `~/home/maro/Documents/ds210-sp26-a1-code/project_3_client_server_a
alytics/target/debug/main hello`
The server says: 'hello'

# maro @ maro-82v2 in ~/Documents/ds210-sp26-a1-code/project_3_client_server_an
alytics/client on git:main o [11:45:55]
$
```