The TrialChain Blockchain API mock.

On this task, you'd **mock** a new, made-up, blockchain, called the TrialChain. For that you'd create a server that **pretends** to be connected to the hypothetical TrialChain network, offering some Rest API Endpoints.

Notice, the idea is only to build the mock, focusing only on the API and simulating anything else.

Requirements.

1. Design how a TrialChain transaction should look like. (Notice, in this context, a transaction is the signed message the peers of the said blockchain network send in order to change the state of the blockchain; A transaction could be something conceptually equivalent to a message such as "I'm Joe and wants to send 1 unit of money to Foo"). Some suggestions for the fields a transaction might have are:

    a. Source(s) of the funds used on the transaction.
    b. Funds destiny(y).
    c. Amount sent.
    d. Signature(s) of the sender (it could be mocked).

2. Create an API method to send TrialChain transactions to the TrialChain blockchain network. Remember that TrialChain is something that you need to mock, and you would assume that is an external service.
    a. Design the Request and Response messages
    b. Desing the possible error messages

3. Create an API method to retrieve a previously broadcasted transaction given its transaction id (the transaction id, what we'll call the txid, will be the hash of the **unsigned** content of the transaction, expressed as a hexadecimal string).

Expected

- Implement this solution in Haskell code using stack to manage its dependencies.
- Implement an HTTP server with the framework of libraries you want (No need to use a streamed TCP interface to the server, just the rest HTTP equivalent would be ok)
- You can use any Haskell lib and any framework.
- Document how to run and build your solution.
- Document how to run your solution and test it with some example data
- Do Testing
- Document how to run tests
- Don’t forget the point of this task is to implement a mock, so everything should be as simple as possible.
- Document how to you solve the problem

You can create a public repo with your solution in your Github account and send us the link after finished.

Good luck!!!