Copy this doc

New Test (Sablier)

Context

In achieving our mission to **redefine business banking** we build Multis with the following strategies in mind:

1. Provide a great UX
2. Build on top of existing protocols
3. Avoid custodial risks

Tech stack

Multis is part of the trend around *static websites* and *serverless computing*. Almost all the logic is happening on the frontend, making the case for the use of a more powerful programming language: *ClojureScript*.
More info can be found in this blog post: https://medium.com/multis/imagining-a-leaner-way-how-to-ship-a-highly-dynamic-webapp-as-a-static-website-5088f83c3813

Test

Given that Multis is a non-standard webapp running in this ever-evolving blockchain space, we want candidates to **1/** prove knowledge of the web3 stack and **2/** understand Multis mission and strategies.

Goals

The goals of this test are:

- Deploy a **static** website
- Connected to the Ethereum blockchain **Rinkeby** via MetaMask
- Allowing a user to **stream** `n` **ETH to an address** `a` **over** `h` **hours**

Tips

Copy this doc

- In true Multis fashion, it is recommended to use an existing protocol for streaming value: The Sablier protocol (documentation is here)
- A user is only represented by the Ethereum address of their account provided by MetaMask
- Etherscan is your friend
- No fancy UI is needed, just a good enough UX
- GitHub pages is perfect for simple static websites
- It has to be a re-frame ClojureScript project with Shadow-cljs as the build tool (and integration with npm)
- Usage of a web3 js library (like this one) can be done directly — no cljs wrappers are needed
- VSCode with Calva is helpful
- Have fun!

🛰 Next

1. When you're done send us an email with a link to your GitHub repo.
2. I'll review your code and decide to setup a follow-up call or in-person meeting.
3. We'll chat about the test and do some peer coding on your machine.

💪*Onwards*