**Description**

When a new token is deployed via the token deployer, users will choose a representative from a dropdown. This representative will be available to answer questions and assist with a smooth launch for the user.

The rep will have an account page accessible on the ftp website which will allow him to view all tokens he is responsible for assisting with.

Page requirements:

* List tokens associated with reps
  + includes token name, symbol, telegram, twitter, site, whitepaper, etherscan, owner address
* Display weekly and total earnings
* Display unpaid earnings
* Display chart for earnings over time (week by week)
* Sort tokens based on owner
* Provide input field for notes on owner
* Provide input field for notes on token
* Notes will ultimately be pushed to central repository for reference by team

Assumptions:

* All data manipulation permissions will be enforced on the backend API when provided a connected wallet hash from the frontend
* Not all required API endpoints will be ready when building the front end. Mock data can be created for testing

Constraints:

* We don’t use React or Angular
* Any library that can assist you via npm is good to use.

Always keep user privacy and data protection at the forefront of your mind.

Soo Kim | FTP | I'll Never Message You, [16.08.21 14:41]

[In reply to Moonco]

this is good. kinda the layout we are leaning toward. although it will be dark themed

Jim|FairTokenProject(will never ask for money), [16.08.21 15:14]

This compact option is pretty much spot on to what we are envisioning

granted, this is more site wide, not so much for rep page

the rep page itself is not going to be in anyway related to the rest of the site, So i wouldnt worry about this stuff too much

essentially theres no need for a sidebar or navigation at all, aside from Home

Soo Kim | FTP | I'll Never Message You, [16.08.21 15:17]

the rep page itself probably wont need navigation

not at first at least

want to keep it as simple as possible

Moonco, [16.08.21 15:22]

I was thinking there's really 2 different comment-exchange features you're headed toward with the rep page - correct me if I'm wrong.

The epic description appears to mention:

~~1) communication with users - a rep's admin interface would display incoming queries from end-users which they can answer (SMPT capability, eg. mailgun, built in to site?)~~

2) internal note taking mechanism where our various reps can save notes on A) tokens and B) token owners separately. These notes are collected db side.

— assumed that a rep will always be able to view their own previously saved notes

— perhaps a rep might also be given permissions to also see other rep's notes for a given token or token owner?

— there might be an interface where elevated admins (like Jim) can read \*all\* notes

Jim|FairTokenProject(will never ask for money), [16.08.21 15:26]

So the notes, absolutely. The communication side theres a lot to design here. Starting with how random leads are even filtered/sorted. Something internal and automatic in this regard is definitely not a consideration at this point. We arent even sure if thats the best route.

As our dominance ramps up, and it gets to a point where people dont even know anything besides FTP(like people dont even know what etherscan is). Then creating this internal structure would be more of a priority

Soo Kim | FTP | I'll Never Message You, [16.08.21 15:26]

the reps will only be responsible for dealing with ppl on telegram, not on our site

we dont want to manage comms on the site

Jim|FairTokenProject(will never ask for money), [16.08.21 15:27]

What we need in the rep page right now is a front end that can

\* return earnings values to the reps,

\* catalog their tokens (upcoming and launched),

\* and a generic note interface.

Dont go too crazy with it

Moonco, [16.08.21 15:31]

[In reply to Jim|FairTokenProject(will never ask for money)]

Got it. So I'm thinking of a dashboard for viewing all tokens' metrics at once, and then a global dropdown (like at the top left somewhere) which switches a rep's global context for that specific token.

An alternative UX flow might be that there is a notes page, and all the input boxes appear for all tokens on that page at once. (similar to how the dashboard shows an overview of everything at once.)

Jim|FairTokenProject(will never ask for money), [16.08.21 15:32]

well for now there wont be a database for historical data on tokens

everything will be pulled from the blockchain

so you can create a way to show multiple tokens at once, just know that its a bit more lethargic. And trying to pull 100 different tokens each time the page loads would be cumbersome

In the future there will be a database that is updated from our nodes. Just isnt there now

So, what you have access to on the blockchain, is total earnings ever; current unpaid balance; weekly total earnings; and total earnings per contract

Also something super important to know is that reps will pretty much have no involvement with these tokens after about an hour of launch

Moonco, [16.08.21 15:39]

oh I see, so the "rep" is like what basedwinner was when we launched dogereloaded

Jim|FairTokenProject(will never ask for money), [16.08.21 15:39]

The main purpose of the data that exists currently is just because we manually have to code it. The only reason for the rep to have context of historical data is if they are working with repeat customers

There are going to be an absolute assload of cases where reps never even talk to the person they are assigned to

because the deployer will have such an intuitive interface

I know trying to create something without knowing all this is rough :P. Just start simple. If we could get a dashboard that reps could log into (connect their wallet), and see their earnings data from the web3 calls, that would be a great start for us

the notes section would be the other useful point. After that you can really go crazy with improving efficiency for the reps

Moonco, [16.08.21 15:45]

hmm... if I am a bad guy, can't I use fiddler to intercept my server request after connecting my wallet, and use gotten sensitive info to supplant a wallet address of someone whom I believe might have admin rights in FTP backend?

For example, if I know basedwinner's wallet hash, I could pretend to be his wallet hash, and FTP server might think I'm him, and send the response meant for him to my browser. would an attack like that be possible?

Jim|FairTokenProject(will never ask for money), [16.08.21 15:46]

you cant

but also, all this will be front end

once the database stuff is setup, our backend will update a database with the exact same calls your frontend will be making

then the page will switch to using the database in a traditional setup

so essentially, when you call the payroll contract via web3 (uses your provider eg: Metamask) it checks what the requesting wallet is. If that wallet isn't approved to call the view function, it will return 0. So they couldnt just say ask from your wallet, they would have to have your wallet.

but, in this case. these calls are just view functions

Moonco, [16.08.21 15:50]

oh I see, there is a direct communication occurring with the wallet that can't be tampered with

Jim|FairTokenProject(will never ask for money), [16.08.21 15:50]

so even if they did find the byte data and replicate it in their own web3 call. it would only return a number.

and that number is the same thing as what someone could add up by watching our payroll contract

in regards to this specifically

For example, if I know basedwinner's wallet hash, I could pretend to be his wallet hash, and FTP server might think I'm him,

There is no communication to the backend based on this kind of verification

right, just imagine the blockchain as another database

So for a security point, leaving ABI on the front end is a security risk, if your stuff solidity side isnt permission locked. And even then, you may not want people knowing the details of your contract

We will just have bytecode data for 3 functions, so someone could technically grab this data, but they can also get it from etherscan

Jth Studios, [16.08.21 16:49]

I guess my personal hesitation with introducing new technologies, tech stacks, and spreading things across various repos / access points / logins / etc is the additional complexity that is introduced without solid reasoning. Not that you haven't provided solid reasoning above, but I wouldn't pull the trigger without looking through it first. It's going to be Soo and Jims decision to make in the long run. I know that we are trying to complete these updates very quickly, and a framework may be the way to go with that, but I will also say that with frameworks comes maintanence, regardless of the framework.

In the future a framework could absolutely be the way to go, but is it the way to go when we are trying to release proofs of concept as quickly as possible?

Soo Kim | FTP | I'll Never Message You, [16.08.21 17:18]

I appreciate your input here, and I value your knowledge but right now I have no interest in modifying our backend stack. its working well right now and I've not heard complaints on load times

i want everything to conform

we are beginning to deploy our flagship products starting the 20th and continuing over the next month, so lets keep it as it is.

once we all agree on your rep page mockups ill get you started with git

all of our subdomains exist in one repo

id prefer it to stay that way as we move forward

feel free to use anything you need to

im also completely ok with you using react for the subdomain

but it needs to get packed up with webpack and conform to our current pipeline which is cloudfront