BloqFin

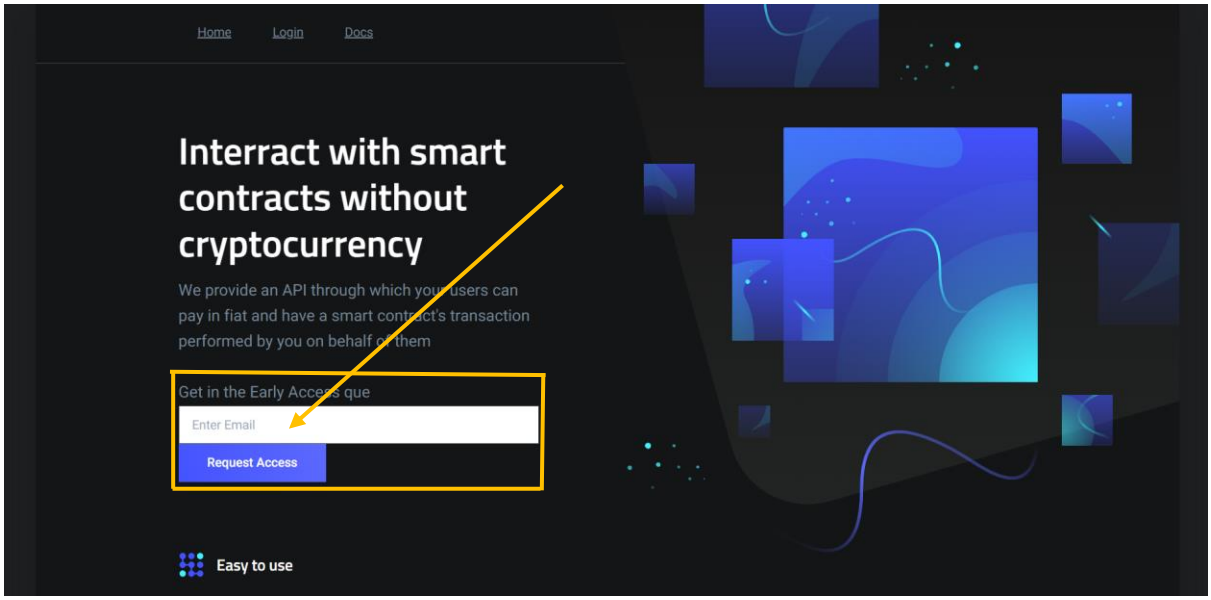
Connecting the Centralized and Decentralized world

API Docs

Index:

1. Request Access:

Enter your email, preferably with a company email, and press the ‘Request Access’ button.

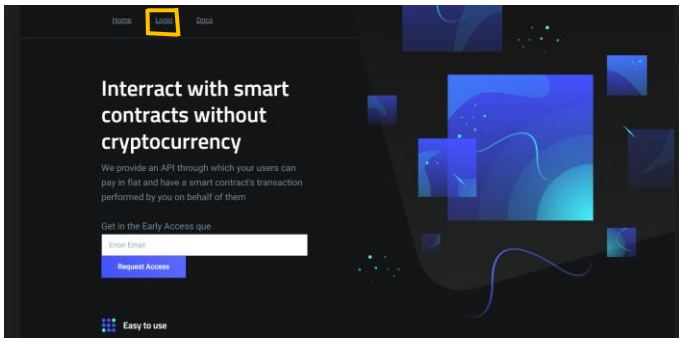


We’ll get back to you once we launch our beta publicly to know more about the project and how to integrate it with our API

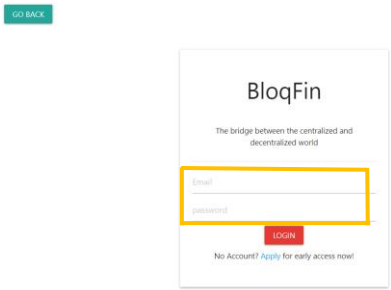
2. How to use BloqFin

1. Go to Login Page
2. Login with credentials provided
3. Click on the add button
4. Add your contract address, ABI, code (code explained in next part)
   Add a title to it. Chain IDs include ‘ROPSTEN’, ‘MATIC’ and ‘BSC’ for Ethereum’s test chain Ropsten, Polygon Matic, and Binance Smart Chain respectively.
   ‘Enter On Success Webhook Link’ and ‘Enter your Input Parameters’ will record your input but they aren’t active yet. Will explain about them in future versions
5. Click on save
6. You’ll be redirected to dashboard, click on the project with that title
7. Please wait until you get the link, it will be displayed on the top
8. Once you get the link, use it in the next part ‘How to code with BloqFin’

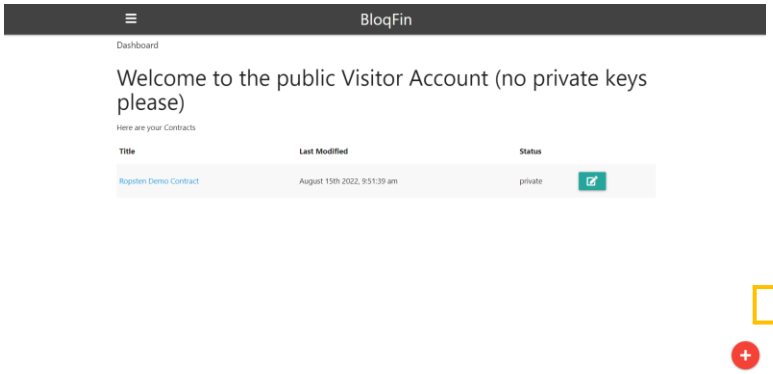
1.



2.



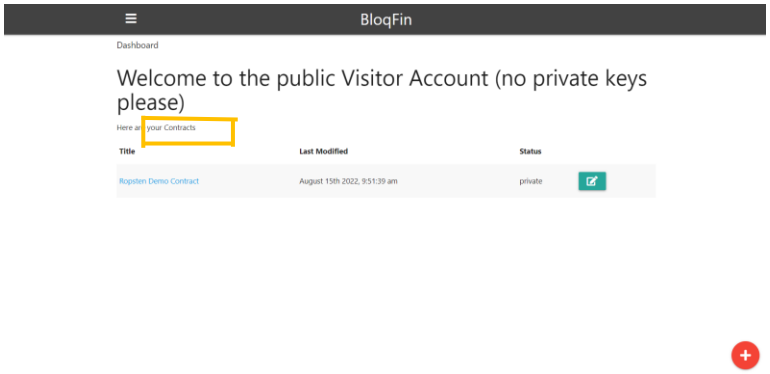
3.



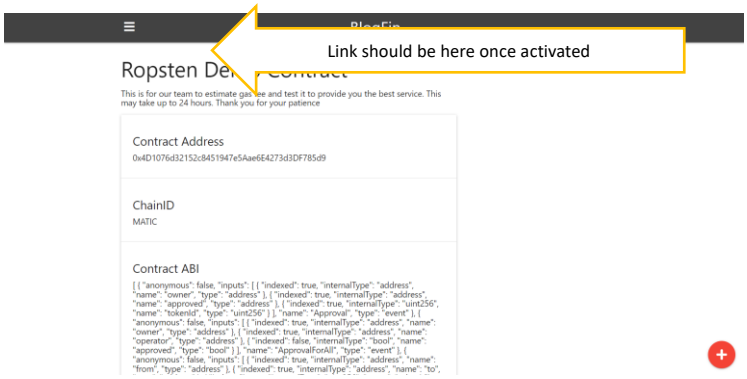
4.

5.

6.



7.



Note: in this example we used a demo account. The link in this account will never activate

3. How to code with BloqFin

Please upload the code with BloqFin in mind. We can take JSON request with upto 10 inputs
POST the request to api.bloqfin.com
The JSON is as follows:

```
{
      contractID: "api.bloqfin.com/request/62f38037f4ccf45380c5e2da",
      successURL: "http://localhost:5500/client/success.html",
      cancelURL: "http://localhost:5500/client/cancel.html",
      priceInEth: 0.0001,
      gasLimit: 0.001,
      slippageLimit: 5,
      input0: " ",
      input1: " ",
      input2: " ",
      input3: " ",
      input4: " ",
      input5: " ",
      input6: " ",
      input7: " ",
      input8: " ",
      input9: " ",
};
```

contractID: the link provided mentioned in step 8 previously. This will take you to a stripe payment session.
successURL: Redirected to this URL on a successful payment
cancelURL: Redirected to this URL on a failed payment
priceInEth: Price involved in the transaction. This can be 0
gasLimit: Gas limit to execute the smart contract. Must be greater than 0
slippageLimit: Limit to slippage that you would like to allow
input0 – input9: inputs to use in the code

Note: gas fees in extra steps are added be BloqFin seperately
Note: payment refers to fiat payments and transaction refers to blockchain transactions
Note: if the price changes past slippage, the payment can succeed but no transaction occurs on the blockchain. The payment can be refunded

In the code, please assign inputs to your variable. Eg: const yourVariable = input0;
Your smart contract address and abi can be reassigned to variables too. Eg:

```
const MyContractAddress = contractAdd;
const my_contract_abi = contractABI;
const MyContract = new web3.eth.Contract(my_contract_abi, MyContractAddress);
```

This is to provide as much freedom as possible to you, as a developer, on your preferred