

The screenshot illustrates the development process of a web application component. On the left, a code editor displays the `Transactions.js` file, which contains JSX code for rendering transaction data based on the user's account status. On the right, a browser window shows the resulting UI on `localhost:3000`. A large red arrow points from the word "Transactions" in the code editor to the corresponding heading in the browser, indicating the flow from code to interface.

`Transactions.js` content:

```
8     return (
9       <div className="flex w-full justify-center
10      items-center 2xl:px-20
11      gradient-bg-transactions">
12       <div className="flex flex-col md:p-12
13      py-12 px-4">
14         {currentAccount ? (
15           <h3 className="text-white text-3xl
16           text-center my-2">
17             Latest Transactions
18           </h3>
19         ) : (
20           <h3 className="text-white text-3xl
21           text-center my-2">
22             Connect your account to see the
23             latest transactions
24           </h3>
25         )
26       </div>
27     </div>
28   );
29 }
```

Browser UI content (localhost:3000):

Services that we continue to improve

Latest Transactions

The screenshot shows a developer's environment with a code editor and a browser preview side-by-side.

Code Editor (VS Code):

- File:** Transactions.jsx - web3....
- File Path:** client > src > components > Transactions.jsx
- Content:** JSX code for a transaction component. It includes a conditional check for a current account, displaying "Latest Transactions" if true or a connection message if false.
- Terminal:** Local host setup and a "ready in 1466ms" message.

Browser Preview:

- Title:** Vite App
- URL:** localhost:3000
- Content:** A dark-themed landing page with a large white text area containing the message "Services that we continue to improve".
- Right Panel:** A sidebar with a red arrow pointing from the code editor to the "Connect your account to see the latest transactions" message in the browser.
- Text in Right Panel:**
 - Security is guaranteed. We always maintain privacy and maintain the quality of our products
 - Fastest transactions
 - Security is guaranteed. We always maintain privacy and maintain the quality of our products

EXPLORER ... TransactionContext.jsx Transactions.jsx Welcome.jsx ●

WEB3.0

client > src > components > Welcome.jsx > [ε] Welcome

```
97          </div>
98          <BsInfoCircle fontSize={17} color="#fff" />
99      </div>
100
101
102      | | | <div>
103      | | |   <p className="text-white font-light text-sm">
104      | | |     {currentAccount}
105      | | |   </p>
106      | | |   <p className="text-white font-semibold text-lg mt-1">
107      | | |     Ethereum
108      | | |   </p>
109      | | |   </div>
110      | | </div>
111      | </div>
112
113      <div className="p-5 sm:w-96 w-full flex flex-col justify-start items-center blue-glassmorphism">
```

The screenshot shows a developer's environment with a code editor and a browser preview side-by-side.

Code Editor (Left):

- File: Welcome.jsx - web3.0 - ...
- Content: JSX code for a component named 'Welcome'.
- Line 44: `<BsInfoCircle fontSize={17} color="#fff" />`
- Line 45: `<BsInfoCircle fontSize={17} color="#fff" />`
- Line 46: `</div>`
- Line 47: `<div>`
- Line 48: `<p className="text-white font-light text-sm">`
- Line 49: `{currentAccount}` (highlighted with a red arrow pointing from the browser preview)
- Line 50: `</p>`
- Line 51: `<p className="text-white font-semibold text-lg mt-1">`
- Line 52: `Ethereum`
- Line 53: `</p>`
- Line 54: `</div>`
- Line 55: `</div>`

Terminal:

- Output: `0xe5ded5c8b695c7c0892340bb694e1280935599f6`

Browser Preview (Right):

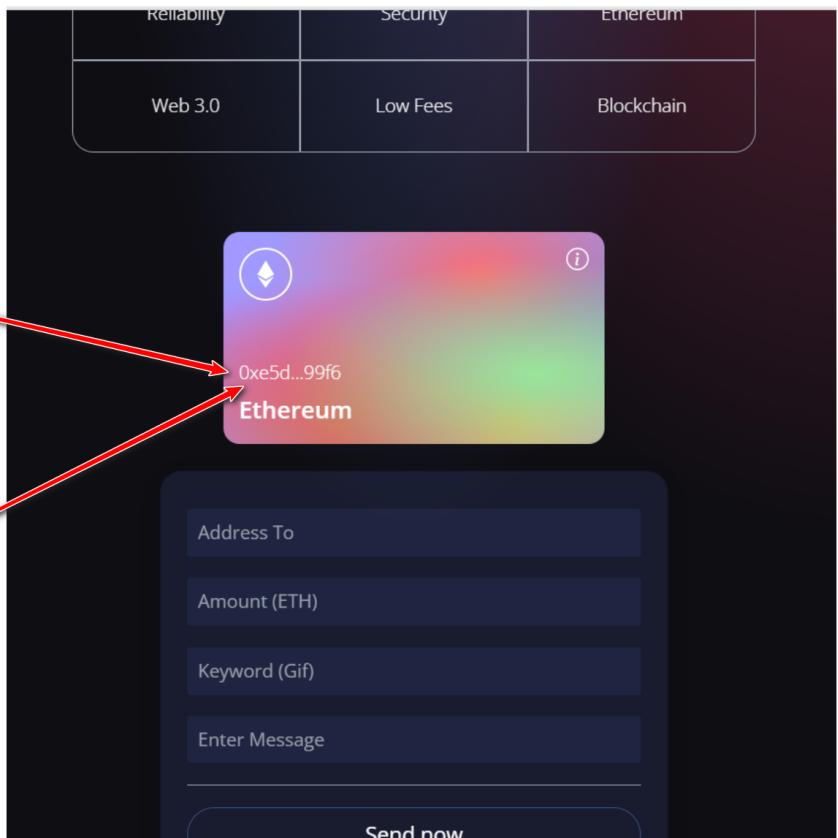
- Address: `localhost:3000`
- Title: Vite App
- Content:
 - Header: **CROSS THE WORLD**
 - Text: Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto.
 - Table:

Reliability	Security	Ethereum
Web 3.0	Low Fees	Blockchain
 - Footer: `0xe5ded5c8b695c7c0892340bb694e1280935599f6` Ethereum

```

1 import React, { useContext } from "react";
2
3 import { AiFillPlayCircle } from "react-icons/ai";
4 import { SiEthereum } from "react-icons/si";
5 import { BsInfoCircle } from "react-icons/bs";
6
7 import { TransactionContext } from "../context/TransactionContext";
8 import { Loader } from "./";
9 import { shortenAddress } from "../utils/shortenAddress"
10
11 const CommonStyles = "min-h-[70px] sm:px-0 px-2 sm:min-w-[120px] flex justify-center items-center border-[0.5px solid #e0e0e0] border-gray-400 text-sm font-light text-white";
12
13 > const Input = ({ placeholder, name, type, value, handleChange }) => (
14   <input
15     type={type}
16     placeholder={placeholder}
17     name={name}
18     value={value}
19     onChange={handleChange}
20   />
21 );
22
23 const Welcome = () => {
24
25   const { currentAccount, connectWallet, handleChange, sendTransaction, formData, isLoading } = useContext(
26     TransactionContext
27   );
28
29   const handleSubmit = (e) => {
30     const { addressTo, amount, keyword, message } = formData;
31
32     e.preventDefault();
33     if (!addressTo || !amount || !keyword || !message) return;
34
35     <div>
36       <p>Address To</p>
37       <p>Amount (ETH)</p>
38       <p>Keyword (Gif)</p>
39       <p>Enter Message</p>
40     </div>
41
42     <div>
43       <p>Send now</p>
44     </div>
45   };
46
47   <div>
48     <p>Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto.</p>
49   </div>
50 }

```



File Edit Selection View Go Run Terminal Help

Welcome.jsx - web3.0

TransactionContext.jsx Transactions.jsx Welcome.jsx

client > src > components > Welcome.jsx > ...

```
127           rounded-full cursor-pointer"
128         >
129         |   Send now
130         |   </button>
131       )
132     </div>
133
134   </div>
135
136   </div>
137 </div>
138 );
139 }
140
141 export default Welcome;
```

PROBLEMS OUTPUT TERMINAL

> < TERMINAL

0 0 0

Windows Taskbar icons: File Explorer, Google Chrome, VS Code, Microsoft Edge, Microsoft Word.

Vite App (893) किती (893) Build

localhost:3000

Reliability Security Ethereum

Web 3.0 Low Fees Blockchain

Ethereum icon

0xe5d...99f6

Ethereum

Address To

Amount (ETH)

Keyword (Gif)

Enter Message

Send now

01:52 AM 18-06-2022

```
Transactions.js X Welcome.jsx
client > src > components > Transactions.js > Transactions
1 import React, { useContext } from "react";
2 import { TransactionContext } from "../context/TransactionContext";
3 import dummyData from "../utils/dummyData";
4 import { shortenAddress } from "../utils/shortenAddress"
5
6 const TransactionsCard = ({ addressTo, addressFrom, timestamp, message, keyword, amount, url }) => {return 'transaction'}
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
<div className="flex flex-wrap justify-center items-center mt-10">
26 {dummyData.reverse().map((transaction, i) =>
27 (
28     <TransactionsCard key={i} {...transaction} />
29 ))}
30
31 <gradient-bg-transactions>
32     <div className="flex flex-col md:p-12 py-12 px-4">
```

Services that we continue to improve



Best exchange rates

Security is guaranteed. We always maintain privacy and maintain the quality of our products



Fastest transactions

Security is guaranteed. We always maintain privacy and maintain the quality of our products

Latest Transactions

transactiontransactiontransactiontransactiontransactiontransactiontransaction

Services that we continue to improve



Best exchange rates

Security is guaranteed. We always maintain privacy and maintain the quality of our products



Fastest transactions

Security is guaranteed. We always maintain privacy and maintain the quality of our products

Latest Transactions

transaction transaction transaction transaction transaction transaction

A screenshot of a developer's workspace showing a code editor and a browser. The code editor on the left displays the file `TransactionsCard.js` with the following content:

```
1 import React, { useContext } from "react";
2 import { TransactionContext } from "../context/TransactionContext";
3 import dummyData from "../utils/dummyData";
4 import { shortenAddress } from "../utils/shortenAddress"
5
6 const TransactionsCard = ({ addressTo,
7   addressFrom, timestamp, message, keyword,
8   amount, url }) => {
9   return (
10     <div className="bg-[#181918] m-4 flex flex-1
11       2xl:min-w-[450px]
12       2xl:max-w-[500px]
13       sm:min-w-[270px]
14       sm:max-w-[300px]
15       min-w-full
16       flex-col p-3 rounded-md hover:shadow-2xl"
17     >
18       <div className="flex flex-col items-center
19         w-full mt-3">
20         </div>
21       </div>
22     )
23   }
24 }
```

The browser window on the right shows a dark-themed web application titled "Latest Transactions". It displays a list of transaction cards, each consisting of a small image, some text, and a "View Details" button. A red arrow points from the highlighted CSS class definition in the code editor to one of the transaction cards in the browser, indicating the visual representation of the styled element.

```
Edit Selection View ... Transactions.jsx - web3.... X
Transactions.jsx X Welcome.jsx
client > src > components > Transactions.jsx > TransactionsCard
6  const TransactionsCard = ({ addressTo, addressFrom,
7    timestamp, message, keyword, amount, url }) => {
8    return [
9      <div className="bg-[#181918] m-4 flex flex-1
10     2xl:min-w-[450px]
11     2xl:max-w-[500px]
12     sm:min-w-[270px]
13     sm:max-w-[300px]
14     min-w-full
15     flex-col p-3 rounded-md hover:shadow-2xl"
16   >
17     <div className="flex flex-col items-center
18     w-full mt-3">
19       <div className="display-flex justify-start
20     w-full mb-6 p-2">
21         <a href={`https://ropsten.etherscan.io/
22           address/${addressFrom}`} target="_blank"
23           rel="noopener noreferrer">
24           <p className="text-white text-base">From:
25             {shortenAddress(addressFrom)}</p>
26         </a>
27       </div>
28     </div>
29   </div>
30 >]
```

Latest Transactions

From: 0xCF8...6A90

From: 0xCF8...6A90

From: 0xCF8...6A90

From: 0xCF8...6A90

From: 0xCF8...6A90

From: 0xCF8...6A90

[All Filters](#)


[ROPSTEN Testnet Network](#)
[Home](#)
[Blockchain](#)
[Tokens](#)
[Misc](#)
[Ropsten](#)
Address 0xCF8e569A97c423952DdFf902375C7C76549A6A90


Overview

Balance: 0.0000000000000063 Ether

Token:

\$0.00 2

More Info

My Name Tag: Not Available

[Transactions](#) [Internal Txns](#) [Erc20 Token Txns](#)

Latest 25 from a total of 156 transactions



Txn Hash	Method <small>i</small>	Block	Age	From <small>▼</small>	To <small>▼</small>	Value	Txn Fee
0x5e652f58fa32e04ec3b...	Transfer	12331301	12 days 21 hrs ago	0xcf8e569a97c423952d...	OUT	0x0dec3cbcd520272979...	1.261641498847641 Ether
0xf49b8dd284660d6cc6...	Transfer	12331299	12 days 21 hrs ago	0xba334c9cdad292bc92...	IN	0xcf8e569a97c423952d...	1.261692423847872 Ether
0x72a1760b9e2a21fd7...	Transfer	12296140	24 days 5 hrs ago	0xcf8e569a97c423952d...	OUT	0xe1c5b0f5b85f0fad577...	2.902393943936929 Ether
0x518ae571c2b1b4b544...	Transfer	12295048	24 days 13 hrs ago	0x81b7e08f65bdf564860...	IN	0xcf8e569a97c423952d...	1 Ether

The screenshot illustrates a development environment where a developer is working on a React application to display recent Ethereum transactions.

Code Editor (VS Code):

- The current file is `TransactionsCard`.
- The code defines a card component with a list of transactions.
- Each transaction item contains two hyperlinks to Etherscan, one for the "From" address and one for the "To" address.
- A red box highlights the second "To" address hyperlink in the code, which corresponds to the transaction shown in the browser.

Browser:

- The browser title is "Vite App".
- The address bar shows "localhost:3000".
- The page title is "Latest Transactions".
- There are five transaction cards displayed, each showing a transfer from address `0xCF8...6A90` to address `0x8aa...fdbE`.

```
Edit Selection View ... Transactions.jsx - web3.... □ □ □ | 08 - □ X
Transactions.jsx × Welcome.jsx
client > src > components > Transactions.jsx > [TransactionsCard
  ↗
  13   min-w-full
  14   flex-col p-3 rounded-md hover:shadow-2xl"
  15   >
  16     <div className="flex flex-col items-center
  17       w-full mt-3">
  18       <div className="display-flex justify-start
  19         w-full mb-6 p-2">
  20         <a href={`https://ropsten.etherscan.io/
  21           address/${addressFrom}`} target="_blank"
  22             rel="noopener noreferrer">
  23               <p className="text-white text-base">From:
  24                 {shortenAddress(addressFrom)}</p>
  25             </a>
  26             <a href={`https://ropsten.etherscan.io/
  27               address/${addressTo}`} target="_blank"
  28                 rel="noopener noreferrer">
  29                   <p className="text-white text-base">To:
  30                     {shortenAddress(addressTo)}</p>
  31               </a>
  32               <p className="text-white text-base">Amount:
  33                 {amount} ETH</p>
  34             </div>
  35           </div>
  36         </div>
  37       </div>
  38     </div>
  39   </div>
  40 </div>
```

Latest Transactions

From: 0xCF8...6A90

To: 0x8aa...fdbE

Amount: 0.01 ETH

The screenshot illustrates a development environment where a React component is being developed and tested. On the left, a code editor displays the `TransactionsCard` component from the `Transactions.js` file. A red box highlights the timestamp logic within the component's template.

```
        <p className="text-white text-base">To:<br/>{shortenAddress(addressTo)}</p>
        <p className="text-white text-base">Amount:<br/>{amount} ETH</p>

        {message && (
            <>
                <br />
                <p className="text-white text-base">Message: {message}</p>
            </>
        )}
    </div>

    <div className="bg-black p-3 px-5 w-max rounded-3xl -mt-5 shadow-2xl">
        <p className="text-[#37c7da] font-bold">{timestamp}</p>
    </div>
</div>
</div>
)
```

The browser window on the right shows five identical transaction cards, each representing a transfer of 0.01 ETH from address `0xCF8...6A90` to address `0x8aa...fdbE`. All transactions occurred at the same timestamp: `12/21/2021, 4:33:21 PM`.

Index	From	To	Amount	Timestamp
1	0xCF8...6A90	0x8aa...fdbE	0.01 ETH	12/21/2021, 4:33:21 PM
2	0xCF8...6A90	0x8aa...fdbE	0.01 ETH	12/21/2021, 4:33:21 PM
3	0xCF8...6A90	0x8aa...fdbE	0.01 ETH	12/21/2021, 4:33:21 PM
4	0xCF8...6A90	0x8aa...fdbE	0.01 ETH	12/21/2021, 4:33:21 PM
5	0xCF8...6A90	0x8aa...fdbE	0.01 ETH	12/21/2021, 4:33:21 PM

Welcome to your GIPHY Developer Dashboard. Get started by creating an app, where you will be assigned a beta API key. All newly created beta keys subject to rate limits and are best used in a development environment.

Once you are ready to use your app in production, please verify your GIPHY integration, if needed, and upgrade your key by clicking on 'Upgrade to Production'.

Your Apps

Krypt API[Edit](#)

API Key

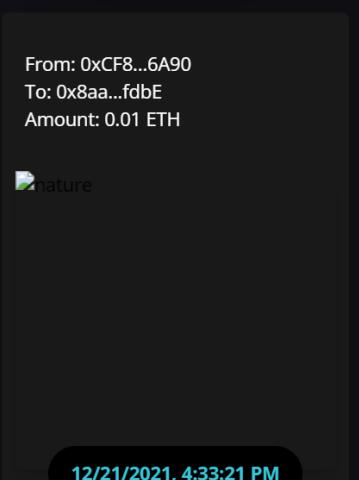
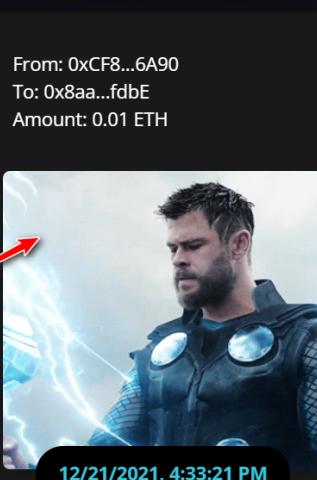
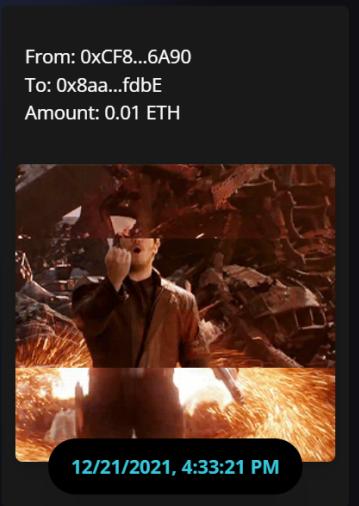
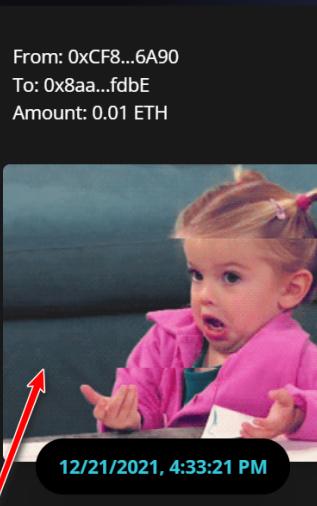
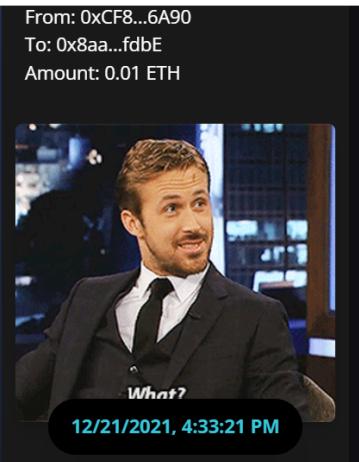
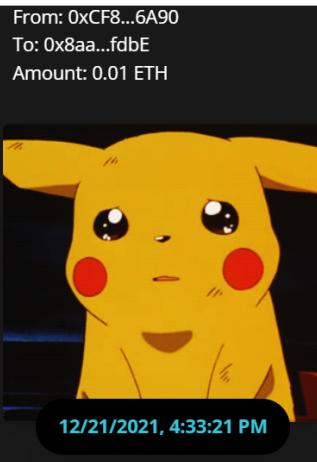
IWlqEESYL8O4at2nnV53VXPUDUYdgyTU



Ready to build another app already?
Click anywhere to get started!

```

Transactions.jsxs X .env useFetch.jsxs Welcome.jsxs ...
client > src > components > Transactions.jsxs > TransactionsCard
  1 import React, { useContext } from "react";
  2 import { TransactionContext } from "../context/TransactionContext";
  3 import dummyData from "../utils/dummyData";
  4 import { shortenAddress } from "../utils/shortenAddress";
  5 import useFetch from "../hooks/useFetch";
  6
  7 const TransactionsCard = ({ addressTo, addressFrom, timestamp, message, keyword, amount, url }) => {
  8   const gifUrl = useFetch({ keyword });
  9
 10   return (
 11     <div className="bg-[#181918] m-4 flex flex-1
 12       2xl:min-w-[450px]
 13       2xl:max-w-[500px]
 14       sm:min-w-[270px]
 15       sm:max-w-[300px]
 16       min-w-full
 17       flex-col p-3 rounded-md hover:shadow-2xl"
 18     >
 19       <div className="flex flex-col items-center w-full mt-3">
 20         <div className="display-flex justify-start w-full mb-6 p-2">
 21           <a href={`https://ropsten.etherscan.io/address/${addressFrom}`} target="_blank"
 22             rel="noopener noreferrer">
 23             <p className="text-white text-base">From:<br/>
 24               ${shortenAddress(addressFrom)}</p>
 25           </a>
 26           <a href={`https://ropsten.etherscan.io/address/${addressTo}`} target="_blank"
 27             rel="noopener noreferrer">
 28             <p className="text-white text-base">To:<br/>
 29               ${shortenAddress(addressTo)}</p>
 30           </a>
 31           <p className="text-white text-base">Amount:<br/>
 32             ${amount} ETH</p>
 33
 34           {message && (
 35             <>
 36               <br />
 37               <p className="text-white text-base">Message:<br/>
 38                 ${message}</p>
 39             </>
 40           )}
 41         </div>
 42
 43         <img
 44           src={gifUrl || url}
 45           alt="nature"
 46           className="w-full h-64 2xl:h-96 rounded-md shadow-lg object-cover"
 47         />
 48       </div>
 49     </div>
 50   </div>
 51 
```



From: 0xCF8...6A90

To: 0x8aa...fdbE

Amount: 0.01 ETH



12/21/2021, 4:33:21 PM

From: 0xCF8...6A90

To: 0x8aa...fdbE

Amount: 0.01 ETH



12/21/2021, 4:33:21 PM

From: 0xCF8...6A90

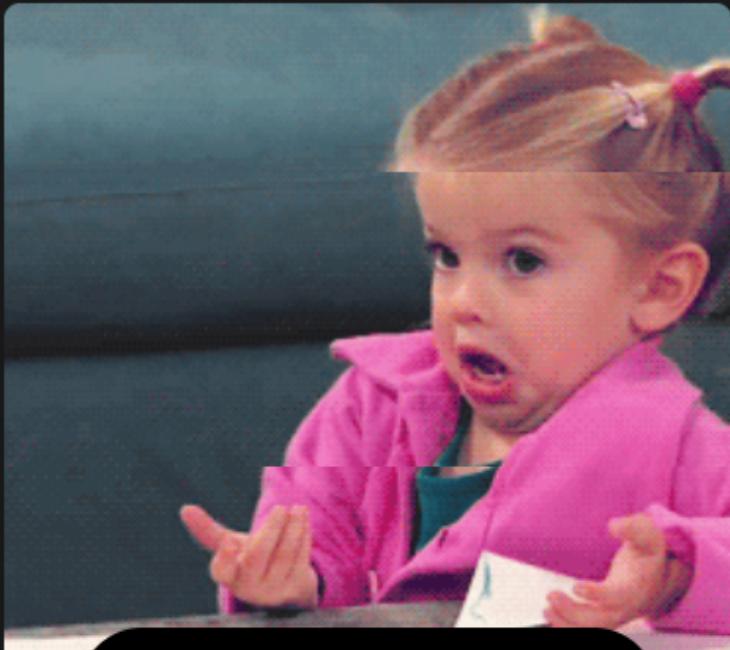
To: 0x8aa...fdbE

Amount: 0.01 ETH

From: 0xCF8...6A90

To: 0x8aa...fdbE

Amount: 0.01 ETH

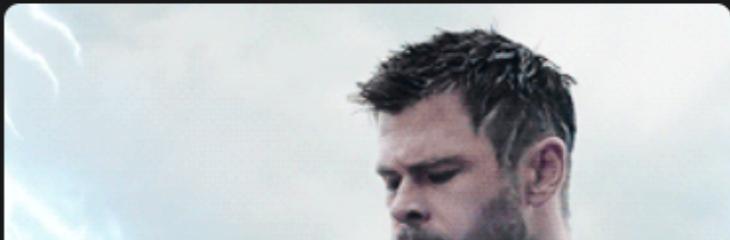


12/21/2021, 4:33:21 PM



12/21/2021, 4:33:21 PM

From: 0xCF8...6A90
To: 0x8aa...fdbE
Amount: 0.01 ETH



From: 0xCF8...6A90
To: 0x8aa...fdbE
Amount: 0.01 ETH





12/21/2021, 4:33:21 PM

12/21/2021, 4:33:21 PM

Vite App (926) Build Choose H Shopping The Com (970) YouTube (970) YouTube Applican Applican

localhost:3000

Google Mails YouTube Udemy HL Network... Trade UpGrad AQUA fight Again ! GitHub Repo मराठी संकेतस्थळ NETR Passion Other bookmarks

TO: 0x00aa...10DE Amount: 0.01 ETH

TO: 0x00aa...10DE Amount: 0.01 ETH

12/21/2021, 4:33:21 PM

12/21/2021, 4:33:21 PM

 KRYPT

Market Exchange Tutorials Wallets

Come join us and hear for the unexpected miracle
info@kryptomastery.com

@kryptomastery2022 All rights reserved

03:59 PM 19-06-2022

```
myData.js ●  ❁ Footer.jsx  ❁ TransactionContext.jsx X  ❁ Welcome.jsx  ⓘ ...  
client > src > context > ❁ TransactionContext.jsx > [x] TransactionsProvider  
43         }  
44     } catch (error) {  
45         console.log(error);  
46     }  
47 };  
48  
49  
50  
51     const checkIfTransactionsExists = async () => {  
52         try {  
53             if (ethereum) {  
54                 const transactionsContract = getEthereumContract()  
55                 const currentTransactionCount = await  
56                 transactionsContract.getTransactionCount();  
57  
58                 window.localStorage.setItem("transactionCount",  
59                 currentTransactionCount);  
60             }  
61         } catch (error) {  
62             console.log(error);  
63             throw new Error("No ethereum object");  
64         };  
65  
66  
67  
121         useEffect (() => {  
122             checkIfWalletIsConnect();  
123             checkIfTransactionsExists();  
124         }  
125         const accounts = await ethereum.request({ method:  
126             "eth_requestAccounts" });  
127  
128     }
```

```
29 |
30 const getAllTransactions =async () => {
31 try {
32   if (!ethereum) return alert("Please install MetaMask.");
33   const transactionsContract = getEthereumContract();
34   const availableTransactions = await transactionsContract.getAllTransactions ();
35   console.log(availableTransactions);
36
37 } catch (error) {
38   console.log(error)
39 }
40 
41 }
42 |
```

```
43
44
45     const checkIfWalletIsConnect = async () => {
46         try {
47             if (!ethereum) return alert("Please install MetaMask.");
48
49             const accounts = await ethereum.request({ method: "eth_accounts" });
50
51             if (accounts.length) {
52                 setCurrentAccount(accounts[0]);
53
54                 getAllTransactions();
55             } else {
56                 console.log("No accounts found");
57             }
58         } catch (error) {
59             console.log(error);
60         }
61     };
62 }
```

The screenshot shows a developer's environment with several windows open:

- Code Editor:** An IDE window displaying `TransactionContext.jsx`. The code handles Ethereum transactions using MetaMask. It includes functions to get all transactions and check if the wallet is connected.
- Browser:** A browser window showing the application at `localhost:3000`. The page has fields for "Amount", "Keyword", and "Enter Message", with a "Send now" button and a shield icon.
- Developer Tools:** The browser's developer tools are open, specifically the "Console" tab. The console output shows Vite logs: "[vite] connecting..." and "[vite] connected.". Below these, there is a detailed log entry for a transaction object:

```
[vite] connecting... client.ts:16
[vite] connected. client.ts:53
TransactionContext.jsx:35
[{"id": 1, "amount": "0x01c6bf52634000", "keyword": "test", "message": "test", "receiver": "0x09B03ceE57a812Eb9A981E2503B995db3e5e5840", "sender": "0xe5DEd5C8B695c7C0892340bb694e1280935599f6", "timestamp": "0x62ac42af", "length": 6}, {"length": 1}]]
```

10
11
12
13
14
15
16
17 export const TransactionsProvider = ({
18 children }) => {
19 const [formData, setformData] = useState({
20 addressTo: "", amount: "", keyword: "",
21 message: ""});
22 const [currentAccount, setCurrentAccount] =
23 useState("");
24 const [isLoading, setIsLoading] = useState
25 (false);
26 const [transactionCount,
27 setTransactionCount] = useState(localStorage.
28 getItem('transactionCount'));
29 const [transactions, setTransactions] =
30 useState([]);
31
32 const handleChange = (e, name) => {
33 setformData((prevState) => ({ ...
34 prevState, [name]: e.target.value }));
35 };
36
37 const getAllTransactions =async () => {
38 try {
39 if (!ethereum) return alert("Please install
40 MetaMask.");
41 const transactionsContract =
42 getEthereumContract();
43 const availableTransactions = await
44 transactionsContract.getAllTransactions ();
45 console.log(availableTransactions);
46
47 const structuredTransactions =
48 availableTransactions.map((transaction) => ({
49 addressTo: transaction.receiver,
50 addressFrom: transaction.sender,
51 timestamp: new Date(transaction.timestamp.
52 toNumber() * 1000).toLocaleString(),
53 message: transaction.message,
54 keyword: transaction.keyword,
55 amount: parseInt(transaction.amount._hex) /
56 (10 ** 18)
57 }));
58
59 console.log(structuredTransactions);
60 setTransactions(structuredTransactions);
61 }
62 catch (error) {
63 console.error(error);
64 }

1 Issue: 1

► 4 messages
► 4 user mess...
✖ No errors
⚠ No warnings
► ⓘ 4 info
⚙ No verbose

[vite] connecting... client.ts:16
[vite] connected. client.ts:53
► [Array(6)] TransactionContext.jsx:35
TransactionContext.jsx:46

▼ [{...}] ⓘ
▼ 0:
addressFrom: "0xe5DEd5C8B695c7C0892340bb6..."
addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
amount: 0.0005
keyword: "test"
message: "test"
timestamp: "6/17/2022, 2:30:31 PM"
► [[Prototype]]: Object
length: 1
► [[Prototype]]: Array(0)



top ▼



Filter

Default levels ▼



1 Issue: 1

- ▶ 4 messages
- ▶ 4 user mess...
- ✖ No errors
- ⚠ No warnings
- ▶ 4 info
- ⚙ No verbose

[vite] connecting... [client.ts:16](#)

[vite] connected. [client.ts:53](#)

▶ [Array(6)] [TransactionContext.jsx:35](#)

▶ [{}...] [TransactionContext.jsx:46](#)

▼ 0:
addressFrom: "0xe5DEd5C8B695c7C0892340bb6..."
addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
amount: 0.0005
keyword: "test"
message: "test"
timestamp: "6/17/2022, 2:30:31 PM"
▶ [[Prototype]]: Object
length: 1
▶ [[Prototype]]: Array(0)



Transactions.jsx

TransactionContext.jsx

Welcome.jsx

```
ient > src > components > Transactions.jsx > [?] Transactions
49
50
51
52 const Transactions = () => {
53   const { currentAccount, transactions } = useContext(TransactionContext);
54
55   return (
56     <div className="flex w-full justify-center items-center 2xl:px-20 gradient-bg-transactions">
57       <div className="flex flex-col md:p-12 py-12 px-4">
58         {currentAccount ? (
59           <h3 className="text-white text-3xl text-center my-2">
60             Latest Transactions
61           </h3>
62         ) : (
63           <h3 className="text-white text-3xl text-center my-2">
64             Connect your account to see the latest transactions
65           </h3>
66         )}
67
68       <div className="flex flex-wrap justify-center items-center mt-10">
69         {transactions.reverse().map((transaction, i) => (
70           <TransactionsCard key={i} {...transaction} />
71         )));
72       </div>
73     </div>
74   )
75 }
```

EXPLORER ... Transactions.jsx TransactionContext.jsx Welcome.jsx

client > src > context > TransactionContext.jsx > [e] TransactionsProvider

```
148
149     useEffect (() => {
150         checkIfWalletIsConnect();
151         checkIfTransactionsExists();
152     },
153     []);
154
155
156
157
158     return [
159         <TransactionContext.Provider value={connectWallet, currentAccount, formData,
160             setformData, handleChange, sendTransaction, transactions, isLoading}>
161             }
162             { children }
163         </TransactionContext.Provider>
164     ];
165 }
```

File Edit Selection View Go Run Terminal Help Welcome.jsx - web3.0 - Visual Studio Code

EXPLORER Transactions.jsx • TransactionContext.jsx Welcome.jsx

```
client > src > components > Welcome.jsx > Welcome > handleSubmit
  4 import { SiEthereum } from "react-icons/si";
  5 import { BsInfoCircle } from "react-icons/bs";
  6
  7 import { TransactionContext } from "../context/TransactionContext";
  8 import { Loader } from "./";
  9 import { shortenAddress } from "../utils/shortenAddress"
10
11 const CommonStyles = "min-h-[70px] sm:px-0 px-2 sm:min-w-[120px] flex justify-center items-center border-[0.5px] border-gray-400 text-sm font-light text-white";
12
13 > const Input = ({ placeholder, name, type, value, handleChange }) => (
14   );
15
16 const Welcome = () => {
17
18   const { currentAccount, connectWallet, handleChange, sendTransaction, formData,
19         isLoading } = useContext(TransactionContext);
20
21   const handleSubmit = (e) => {
22     const { addressTo, amount, keyword, message } = formData;
23
24     e.preventDefault();
25     if (!addressTo || !amount || !keyword || !message) return;
26
27     sendTransaction();
28   };
29
30   <div className="h-[1px] w-full bg-gray-400 my-2" />
31
32   isLoading
33     ? <Loader />
34     : (
35       <button
36         type="button"
37         onClick={handleSubmit}
38         className="text-white w-full mt-2 border-[1px] p-2 border-[#3d4f7c]
39           hover:bg-[#3d4f7c] rounded-full cursor-pointer"
40       >
41         Send now
42       </button>
43     )
44
45   <div>
46     <Text>Address</Text>
47     <Input type="text" name="addressTo" value={formData.addressTo} onChange={handleChange} />
48   </div>
49
50   <Text>Amount</Text>
51   <Input type="text" name="amount" value={formData.amount} onChange={handleChange} />
52
53   <Text>Keyword</Text>
54   <Input type="text" name="keyword" value={formData.keyword} onChange={handleChange} />
55
56   <Text>Message</Text>
57   <Input type="text" name="message" value={formData.message} onChange={handleChange} />
58
59   <Text>Send now</Text>
60   <button>Send now</button>
61
62 </div>
63
```

File Edit Selection View Go Run Terminal Help • Welcome.jsx - web3.0 - Visual Studio Code

EXPLORER ... Transactions.jsx • TransactionContext.jsx • Welcome.jsx •

client > src > components > Welcome.jsx > [e] Welcome

```
114 <Input placeholder="Amount (ETH)" name="amount" type="number" handleChange={handleChange} />
115 <Input placeholder="Keyword (Gif)" name="keyword" type="text" handleChange={handleChange} />
116 <Input placeholder="Enter Message" name="message" type="text" handleChange={handleChange} />
117 <div className="h-[1px] w-full bg-gray-400 my-2" />
118
119
120 {isLoading
121 ? <Loader />
122 : [
123   <button
124     type="button"
125     onClick={handleSubmit}
126     className="text-white w-full mt-2 border-[1px] p-2 border-[#3d4f7c]
127       hover:bg-[#3d4f7c] rounded-full cursor-pointer"
128   >
129     Send now
130   </button>
131 ]
132 </div>
133
134 </div>
135
```

OUTLINE

TIMELINE Welcome.jsx

- File Saved 1 day
- File Saved
- File Saved
- File Saved 2 days
- File Saved
- File Saved

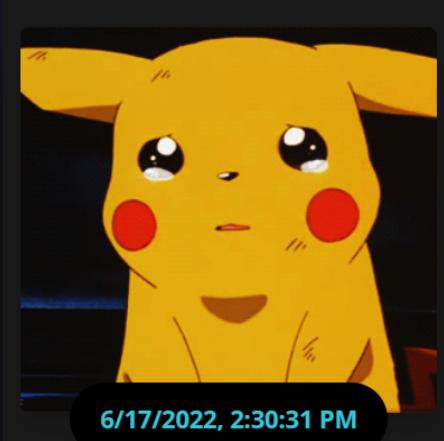
Ln 129, Col 26 Spaces: 2 UTF-8 CRLF {} JavaScript React ⚡ Go Live 🔍 05:41 PM 19-06-2022

Windows Search File Explorer Chrome VS Code Taskbar Settings

C H i localhost:3000

Mails YouTube Udemy HL Network... Trade UpGrad AQUA fight Again ! GitHub Repo मराठी संकेतस्थळ NETR Passion » Other bookmarks

Message: test



6/17/2022, 2:30:31 PM

Elements Console Filter Default levels ▾ 2 hidden

1 Issue: 1

- ▶ 6 messages
- ▶ 4 user mess...
- ▶ 2 errors
- ⚠ No warnings
- ▶ 4 info
- ✖ No verbose

[vite] connecting... client.ts:16
[vite] connected. client.ts:53
▶ [Array(6)] TransactionContext.jsx:35
TransactionContext.jsx:46

▼ [...] ⓘ
▼ 0:
addressFrom: "0xe5DEd5C8B695c7C0892340bb6..."
addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
amount: 0.0005
keyword: "test"
message: "test"
timestamp: "6/17/2022, 2:30:31 PM"
▶ [[Prototype]]: Object
length: 1
▶ [[Prototype]]: Array(0)

KRYPT Market Exchange Tutorials Wallets

Come join us and hear for the unexpected miracle
info@kryptomastery.com

A screenshot of a Windows desktop showing a browser window and the Windows taskbar.

The browser window displays a Vite App at localhost:3000. The app interface includes:

- A purple header bar with a circular icon containing a downward arrow and the text "0xe5d...99f6 Ethereum".
- A dark blue card with the address `0x09B03ceE57a812Eb9A981E2503B995db3e5e584`.
- The amount `0.001`.
- The message `Hello`.
- The message `Hello from account 1`.
- A large button labeled `Send now`.

The browser's address bar shows `localhost:3000`. The taskbar icons include:

- Windows Start button
- Search icon
- File Explorer icon
- Google Chrome icon
- Visual Studio Code icon
- File Manager icon
- Settings icon
- MetaMask icon

The system tray shows the date and time as `19-06-2022 06:45 PM`.

To the right of the browser, a MetaMask extension notification is open, titled "MetaMask Notification". It shows:

- "Edit" button
- "Goerli Test Network" button
- "Account 1" icon
- "Account 2" icon
- The URL `http://localhost:3000`
- The text "SENDING ETH"
- The amount `0.001`
- Gas information: `0.0000315`, `0.0000315 ETH`, `Likely in < 30 seconds`, `Max fee: 0.0000315 ETH`
- Total gas: `0.0010315`, `0.0010315 ETH`, `Amount + gas fee`, `Max amount: 0.0010315 ETH`
- Buttons: `Reject` and `Confirm`

Vite App (localhost:3000)

Ethereum

0xe5d...99f6

0x09B03ceE57a812Eb9A981E2503B995db3e5e584

0.001

Hello

Hello from account 1

Console

11 Issues: 10 Errors, 1 Warning

- [vite] connecting... client.ts:16
- [vite] connected. client.ts:53
- [Array(6)] TransactionContext.jsx:35
- [...] TransactionContext.jsx:46
- [...]:
 - addressFrom: "0xe5DEd5C88695c7C0892340bb6..."
 - addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
 - amount: 0.0005
 - keyword: "test"
 - message: "test"
 - timestamp: "6/17/2022, 2:30:31 PM"
- [[Prototype]]: Object
- length: 1
- [[Prototype]]: Array(0)
- TransactionContext.jsx:142
- {code: -32602, message: 'Invalid parameters: must provide an Ethereum address.', stack: '{}'}
 - code: -32602, message: 'Invalid parameters: must provide an Ethereum address.'
 - stack: '{}'
- Loading - TransactionContext.jsx:131
- 0x0c3bf551116893a5e92783cb14cdb1e25ab0025e261714 e87b360beb32cd2b72



06:46 PM
19-06-2022

Vite App | YouTube (926) Build and | Choose Hostin... | Shopping Cart | The Complete | YouTube (977) National | How to speed | + | - | X

localhost:3000

Google Mails YouTube Udemy HL Network... Trade UpGrad AQUA fight Again ! GitHub Repo मराठी संकेतस्थळ NETR Passion » Other bookmarks

Ethereum

0xe5d...99f6

0x09B03ceE57a812Eb9A981E2503B995db3e5e584

0.001

Hello

Hello from account 1

Send now

Security

Console

11 Issues: 10 1

- 15 messages
- 8 user mess...
- 4 errors
- No warnings
- 7 info
- 4 verbose

[vite] connecting... client.ts:16

[vite] connected. client.ts:53

[Array(6)] TransactionContext.jsx:35

[...] TransactionContext.jsx:46

0:
addressFrom: "0xe5DEd5C88695c7C0892340bb6..."
addressTo: "0x09B03ceE57a812Eb9A981E2503B995db3e5e584"
amount: 0.0005
keyword: "test"
message: "test"
timestamp: "6/17/2022, 2:30:31 PM"

[[Prototype]]: Object
length: 1

[[Prototype]]: Array(0)

TransactionContext.jsx:142

code: -32602, message: 'Invalid parameters: must provide an Ethereum address.', stack: '{}'

code: -32602, message: 'Invalid parameter: hfbeogaeaoehlefknkodbefgpgknn/common-5.js:19: 40881'

Loading - TransactionContext.jsx:131
0x0c3bf551116893a5e92783cb14cdb1e25ab0025e261714
e87b360beb32cd2b72

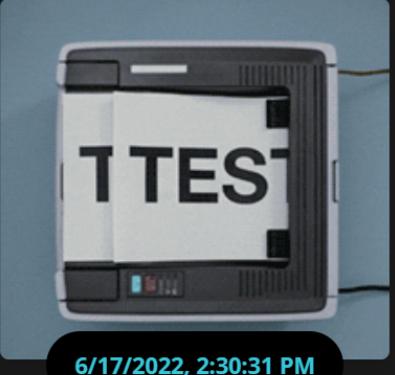
Success - TransactionContext.jsx:133
0x0c3bf551116893a5e92783cb14cdb1e25ab0025e261714
e87b360beb32cd2b72

06:46 PM 19-06-2022

localhost:3000

To: 0x09B...5840
Amount: 0.0005 ETH

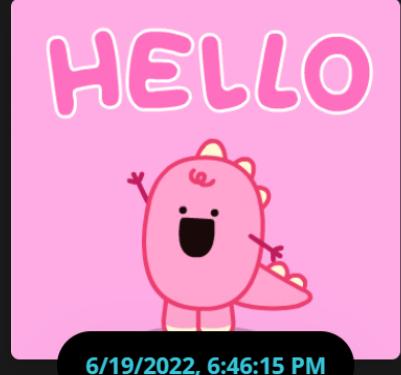
Message: test



6/17/2022, 2:30:31 PM

To: 0x09B...5840
Amount: 0.001 ETH

Message: Hello from account 1



6/19/2022, 6:46:15 PM

From: 0xe5D...99f6
To: 0x09B...5840
Amount: 0.015 ETH

Message: Hers's a gift for you.

From: 0x09B...5840
To: 0xe5D...99f6
Amount: 0.00598988 ETH

Message: Sending Ethers back to account 1

Elements Console Recorder Filter Default levels

17 Issues: 16 1

- [vite] connecting... client.ts:16
- [vite] connected. client.ts:53
- (4) [Array(6), Array(6), Array(6), Array(6)] TransactionContext.jsx:35
- (4) [{} , {} , {} , {}] t TransactionContext.jsx:46
- 0:
 - addressFrom: "0xe5DEd5C8B695c7C0892340bb6..."
 - addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
 - amount: 0.005
 - keyword: "test"
 - message: "test"
 - timestamp: "6/17/2022, 2:30:31 PM"
- 1:
 - addressFrom: "0xe5DEd5C8B695c7C0892340bb6..."
 - addressTo: "0x09B03ceE57a812Eb9A981E2503B..."
 - amount: 0.001
 - keyword: "Hello "
 - message: "Hello from account 1"
 - timestamp: "6/19/2022, 6:46:15 PM"
- 2: {addressTo: '0x09B03ceE57a812Eb9A981E2503B...'}
- 3: {addressTo: '0xe5DEd5C8B695c7C0892340bb6...'}
- length: 4