

Summary

CoinEX Unity SDK is a simple to interact with CoinEX smart chain. By using CoinEX Unity SDK we can interact with CoinEX smart chain inside unity. By using this we can make Gaming DAPP development easier and in a most efficient way. We can also use this SDK for interacting with smart contract inside the CoinEX smart chain. It also comes with prebuild function for interacting with CoinEX smart chain. For now it comes with pre build function for ERC20, ERC721 smart contract's to interact with them.

Difficulties

P2E game development industries is one of the booming game development industries and every year many normal game developers are trying to move into P2E game development but the real problem the face is how to interact with blockchain and they don't know how to interact with them. And our project will solve that problem.

These are some problems faced by game developers who are switching to P2E game development: 1) Interacting with blockchain 2) How to perform those functions 3) Debug those data from blockchain 4) Wallet connection. 5) Open-source Code.

- 1) Interacting with blockchain is difficult for new blockchain based game developers because most of them don't have full knowledge about blockchain dapps.
- 2) Even if they are having blockchain knowledge most of them don't know how to perform those actions inside unity.
- 3) And the main problem is how to debug those data from blockchain. Because blockchain carries many amounts of

data but new P2E game devs don't know how to debug those data for using inside the unity.

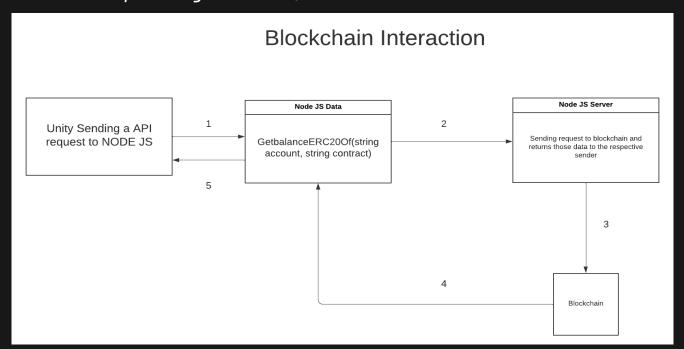
- 4) Wallet Connection is one of the most and import one inside the P2E game development. Because many P2E games were just crashed due to improper wallet maintenance. Many users have lost their assets due to this.
- 5) Open-Source is a key point for developer who needs more improvement in their project. So, we have provided the node js API for open-source so that other developers can host the node js server on their colud and use those for unity bockchain interaction's.

Solutions

Interreacting with blockchain can be achieved by making a backend with node API server. Basically, CoinEX Unity SDK uses a Node Js API server for interacting blockchain. And it also uses NETHEREUM .NET framework for interacting with blockchain. So, that users can achieve these things just by single line of code.

Example Code will be like:

String hash = await CoinEX.ERC20.GetBalanceOf(string
account, string contract)



This single line of code is good enough for interacting with CoinEX smart chain inside unity. The only thing we is just those data's like contract and account string.

Like I said before by using node js at backend for getting those data from blokchain is become easier. And it is very easy to use those function inside the SDK. Making it more flexible for multipe data for interacting with blockchain. Some of the function for interacting with blockchain are:

- CoinEX.ERC20.GetERC20BalanceOf()
- 2) CoinEX.ERC721.GetERC721URI()

ETC...! Making a developer friendly environment for developers to use our CoinEX smart chain inside unity.

And By using those function and methods it will automatically returns all the value's debugged like HexDeciaml to string, hash to string etc...! CoinEX Unity SDK will make everything debugged for the developers.

Wallet Connect is one of the most important function in the P2E game development. Wallet connect function is want to be secured and we want to make sure the wallet interaction is secured. CoinEX Unity SDK uses Metamask Wallet. So, that we can make sure wallets are secured and Easy to interact.

We have also made all of this code open source with MIT license in github. And our main key point is Node JS

```
public async void SendCET() // Single line of code needed for sending CET

( string hash = amait CoinEX.Wallet.SendCET(cet_to.text, CoinEX.Wallet.ToWei(cet_amount.text)); //this one

to_cet text = cet_to to text;
amount_cet_text = cet_amount.text;
t_hash_cet_text = cet_amount.text;
t_hash_cet_text = shash;

Debug_Log(hash);
}

ordererors

public async void CreateERC28() // Single line of code needed for creating erc20
{
    string hash = amait CoinEX.ERC20.CreateERC20(erc20_name.text, erc20_amountc.text, erc20_symbol.text, erc20_decimals.text); //this one
    erc20_contract_text = hash;
    Debug_Log(hash);
}

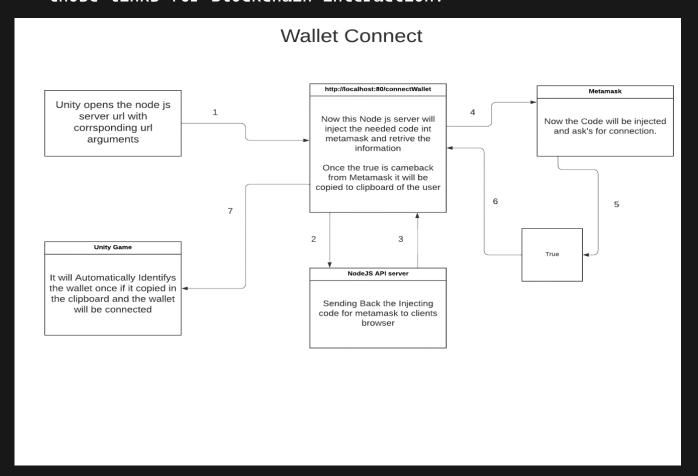
public async void SendERC28() // Single line of code needed for sending ERC20

{
    string hash = amait CoinEX.ERC20.SendERC20fo(erc20_to.text, PlayerPrefs.GetString(*erc20*), CoinEX.Wallet.ToWei(erc20_amount.text)); //this one
    to_erc20.text = erc20_to.text;
amount_erc20.text = erc20_sount.text;
tx_hash_erc20.text = hash;
    Debug_Log(hash);
}

ordererors

public async void GetErc200alance()
{
    string hash = amait CoinEX.ERC20.SendERC200.ERC20BalanceOf(erc20_bal_account.text, PlayerPrefs.GetString(*erc20*));
    erc20_bal_account text = erc20_bal_account.text;
    erc20_bal_amount.text = erc20_bal_account.text;
    erc20_bal_amount.text = bal.ToString();
    bebug_Log(hash);
}
```

server files are aso open source so that developers can add or remove or alter those easy function as per their need. Making it more flexible for development. Developer's can host those node js server files in their cloud and use those links for Blockchain interaction.



How We Build It:

Basically we have first created a unity SDK inside unity editor using nethereum .net framework and UnitywebRequest. And we developed the Node js server for backend in which we can get Blockchain data. Manily the Nodejs server runs with the help for express js. It carries multiple function's such as geterc20balance etc..!

So, How this work is CoinEX Unity SDK consist of multiple functions inside the CoinEX namespace. To retrive the data from the blockchain it will send the data to node js server and retrive

the data from it. And for transaction it will send the needed arguments to the node js server and open the browser from the unity game. The browser will inject the data into the wallet and the transaction will be performed.

To connect the wallet the same method will be done as mentioned above. Once the transaction is done the transaction hash for the transaction will be copied to the clipboard. Once the value of transaction hash is updated to the clipboard it will automatically detected inside using a while loop check the data was starting of 0x and has a bit of 132. Then the function will return the value to the function call.

Node JS API server uses Ethers js for interacting with the blockchain to get the values from the blockchain. And we are using commands of metamask to inject the arguments inside the metamask Wallet.

Conclusion:

At conclusion we are trying to make the Gaming DAPP development easier and in a most efficient way. So, that more number of Game developers can use CoinEX smart chain for using the blockchain for their game inside unity. And we do some new improvements to make sure the Value consistency in game and SDK.

Further Plan of Improvements

- 1) We need to publish the node js server for public usage.
- 2) More amount of work is needed on Debugging and code optimization is needed.

3) New token standards are needed. For now, there are only two token standards have been deployed in game. Soon we will add other token standards.

Links:

1) Github: https://github.com/Pr0fe5s0r/CoinEX-Unity-SDK

2) Youtube: https://youtu.be/F0jTTql1PQ0

Contact:

1) Gmail: santhsri75@gmail.com

2) telegram: https://t.me/IAMTHEprofessor

3) Discord username: Professor#0212