Ethereum Studio Docs

Obsidian Labs

Feb 14, 2022

GETTING STARTED

1	Overview	1
2	Installation 2.1 Download	3 3 3 6
3	3.1 Login web client 3.2 Create an ERC20 project 3.3 Connect to MetaMask Account 3.4 Request Ropsten Faucet 3.5 Build Contracts 3.6 Deploy Contracts	11 11 11 13 15 17 18 20
4	4.1 Create Project 4.2 Open Project 4.3 Local Projects and Cloud Projects	23 23 31 32 33
5	5.1 Build 5.2 Deploy 5.3 Project Settings	35 35 39 47 52
6	6.1 Local Development	57 57 65
7	7.1 Account	73 73 74 74 75 76
8		79 80

	8.2	View Functions	83
	8.3	Events	84
9	Кеур	pair Manager	85
	9.1	Create Keypair	86
	9.2	Check Keypair	89
	9.3	Delete Keypair	89
	9.4	Import Keypair	90
10	RPC	Client	93
11	Supp	ported Faucets	95
	11.1	Ropsten Faucet	95
	11.2		95
	11.3	Goerli Faucet	95
	11.4	Kovan Faucet	96
12	Supp	oorted Frameworks	97
	12.1	Open Zeppelin	97
		Hardhat	97
	12.3	Truffle	97
	12.4	Waffle	98
13	Supp	ported Testnets	99
	13.1	Ropsten	99
	13.2	Rinkeby	99
	13.3	Goerli	99
	13.4	Kovan	99
14	Tran	saction History	101
	14.1	•	101
15	Truff	fle Migration Script	103
16	ABI	Storage	105

CHAPTER

ONE

OVERVIEW

The Ethereum Studio is a world-class Ethereum smart contract and DApp integrated development environment (IDE), aiming to make Ethereum development faster and easier. Ethereum Studio currently offers a standalone desktop application running on macOS, Windows and Linux, and Ethereum Studio Web that runs in modern web browsers. With Ethereum Studio, you can:

- Set and save a project in the cloud or local quickly
- Manage keypair information with MetaMask easily
- · Build and test smart contracts with different framework
- Deploy smart contracts to Ethereum mainnet and testnet
- · Check and call deployed contract functions through the address
- Query address information with Explorer on the selected network
- Setting local RPC node, ABI Storage and other advanced features

display

When deploying contracts, you should use the latest released Docker image version. Apart from exceptional cases, only the latest version receives security fixes. Furthermore, breaking changes as well as new features are introduced regularly. We currently use a 0.y.z version number to indicate this fast pace of change.

Ideas for improving Ethereum Studio or this documentation are always welcome. Read our **contributors guide** for more details.

CHAPTER

TWO

INSTALLATION

2.1 Download

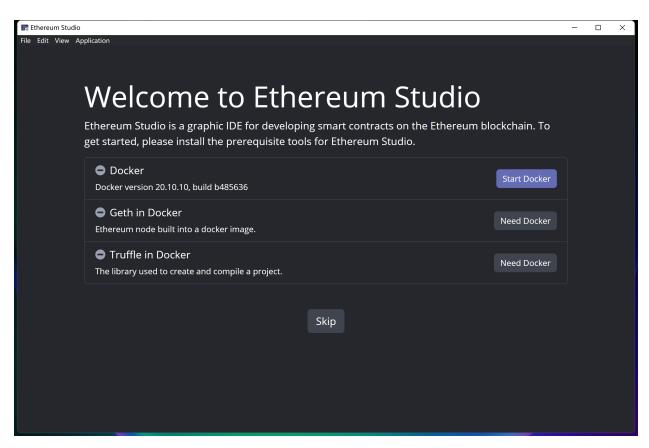
Download Ethereum Studio installation package in Github Ethereum Studio Latest Release according to the computer system type (.dmg for macOS, .AppImage for Linux, .exe for Windows).

2.2 Install Dependencies

Starting Ethereum Studio requires several image dependencies, including Docker, Ethereum Node and Ethereum Truffle. Running Ethereum Studio desktop client requires all three image files. The modules are installed in Docker client like this:

				Upgrade 🔅 🐐	hanyouyan)	_ □ ×
😂 Containers / Apps	Images on disk		3 images	Total size: 590.17 MB	IN USE UN	IUSED Clean up
Images						
📾 Volumes	LOCAL REMOTE REPOSITORIES					
Dev Environments PREVIEW		In	Use only			
	NAME 1	TAG	IMAGE ID	CREATED	SIZE	
	docker/getting-started					
	ethereum/client-go					
	obsidians/truffle					
<u>ه</u>						

Install image files through the "Welcome" page, if there is any missing dependency in Docker. Select the available versions in the "Geth in Docker" list. It is the same for "Truffle in Docker", too.



Click the "Get Started" button, and "Project" interface will pop up, after all the required dependencies are installed.

File Edit View Application	
Welcome to Ethereum Studio Ethereum Studio is a graphic IDE for developing smart contracts on the Ethereum blockchain. To get started, please install the prerequisite tools for Ethereum Studio.	
OckerStartedDocker version 20.10.11, build dea9396Started	
Geth in Docker Installed	
Truffle in Docker Installed Installed	
Get Started	

2.2.1 Docker

Docker is used to start the Ethereum Node and build projects in Ethereum Studio. If Docker is not installed yet, users can click the "Install Docker" button to visit the official Docker website and download and install it.

Docker can not wake up automatically through the Ethereum Studio desktop client now. Open Docker desktop client before starting the the Ethereum Studio desktop client. Otherwise, an error report would remind "Docker has not been installed".

2.2.2 Geth

Geth in Docker is the Ethereum node image. Ethereum Studio uses the image to run the Ethereum node and build projects. Install the Geth image through the "Install" button and select the required version. The latest version is always recommended as the beginner's default version.

2.2.3 Truffle

Truffle in Docker is an Ethereum version of Truffle used to create and build projects. Install the Truffle image through the "Install" button and select the required version. The latest version is always recommended as the beginner's default version.

The grey "Skip" button will change into a violet "Get Started" button after all dependencies are correctly installed. Click it to enter the main interface of Ethereum Studio.

2.3 Contrast with Web Client

F	eatures	Web	Desktop
Drojecto	Cloud	Yes	Yes
Projects	Local	No	Yes
_	Truffle	No	Yes
Framework Choices	Hardhat	No	Yes
Choices	Waffle	No	Yes
Build	by Framework	Only Truffle	Truffle, Hardhat, Waffle and Dockerized Truffle
	by Command Line	No	Yes
Deploy	by Framework	Only Truffle	Truffle, Hardhat, Waffle and Dockerized Truffle
	by Command Line	No	Yes
Login Ac	count by Github	Yes	Yes
Auto-Log Ad	dress by MetaMask	Yes	No
Bloc	kc Explorer	Yes	Yes
c	ontract	Yes	Yes
	Developemt	No	Yes
Network	Testnets and Mainnet	Yes	Yes
	Custom	No	Yes

2.3.1 Create a New Project

Creating an ERC20 project in the Ethereum Studio Desktop client is a bit different, compared with creating it in web client. The "Project name" is "testTransfer" and the "Project location" is automatically settled as "C:\Users\Administrator\Ethereum Studio\testTransfer" in the Windows system. The "Project location" must be an empty documentation. The web client will save projects in the cloud automatically.

Ethereum Studio							-	×
File Edit View Application								
Project (None)	-			O Explorer				
	Create a New Project				×			
	☐ Local							
	Project location							
				Choc	ose			
Local (Project name					v BO		
	testTransfer							
	Template		Open Zep	pelin Versior	ı			
	Basics - ERC20, ERC721 & ERC1155 (v3 Open Zeppe	in 👻	v4.2.0		•			
	Framework	Truffle	version					
	Truffle Hardhat Waffle Dockerized Truffle	v5.4.6						
	Npm client							
	npm yarn cnpm							
			Cancel	Create Pro	oject			
🤌 😐 Development 📲 Transact	tions 🗄 ABI Storage							

In the Ethereum Studio desktop client, set "Template" as "Basics - ERC20, ERC721 & ERC 1155" or Truffle framework "Metacoin" to save time importing basic projects in the later development process. While in the web client, there are only three types to choose from – "Empty Project", "Coin", and "ERC20 Token". The "Open Zeppelin Version" is the latest version automatically.

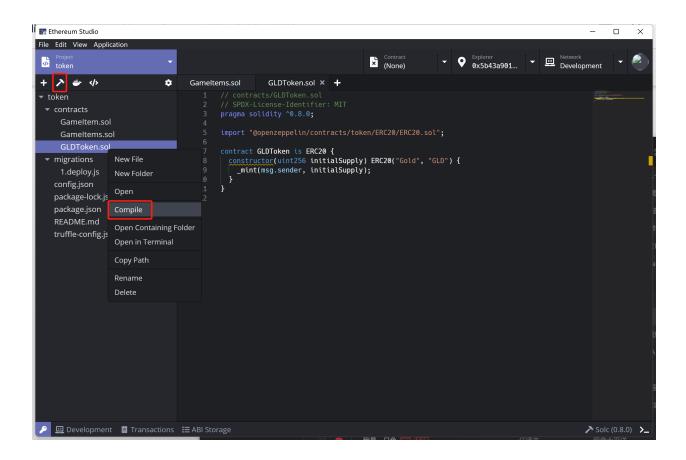
Project X (None)		Contract	Explorer	
	Create a New Project		×	
	Empty Project Coin ERC20 Token OPEN ZEPPELIN Basics - ERC20, ERC721 & ERC1155 (v3.1+)		Choose	
	TRUFFLE			
	Metacoin	C	Open Zeppelin Version	
	Basics - ERC20, ERC721 & ERC1155 (v3	pen Zeppelin 🔻	v4.2.0 •	
	Framework	Truffle ve	ersion	
	Truffle Hardhat Waffle Dockerized Truffle	v5.4.6		
	Npm client			
			Cancel Create Project	
🔎 🖳 Development 📲 Transac	tions ᠄ 🗄 ABI Storage			

Select one of the three different frameworks as the development environment in the Ethereum Studio desktop client. The frameworks are "Truffle", "Hardhat", "Waffle", and "Dockerized Truffle". Besides, there are also three types of "Npm client", which are "npm", "yarn", and "cnpm" that will not show in the web client.

Those frameworks and tools will be used automatically in the command line when building or deploying projects. The default framework version is settled during the install process of Docker, and it will be introduced later.

2.3.2 Build contracts

In the Ethereum Studio desktop client, build a Solidity file through "Right Click", the file name. In contrast, in the web client, clicking the "hammer" icon is the only way to build.



2.3.3 Keypair Manager with MetaMask

The Ethereum Studio web client will wake up the extension of MetaMask automatically in browser. After logging in to the MetaMask account, the Block Explorer will link to the account. Check the detailed information about it. Remember to import the mnemonic in the Keypair Manager to use it in the later paying gas fee in deploying.

Import a MetaMask account through mnemonic in Keypair Manager in the Ethereum Studio desktop client since the browser extension is invalid in the desktop client.

2.3.4 Development and Custom Network

Select "Network" as "Development" to set a local instance in the Ethereum Studio desktop client while the web client has no local network.

F Ethereum Studio					- 0	×
File Edit View Application						
token		×	Contract (None)	 ▼ ♀ Explorer 0x5b43a901 	Development	\bigcirc
Ethereum Instances fo	r Development			🚍 Geth Version Manager	😐 Development	:e
NAME		NODE VERSION	CHAIN	HEIGHT	ETHEREUM	
		(No Geth instance)			🔗 Ropsten	
					🔗 Rinkeby	
					🔗 Gorli	
					🔗 Kovan	
					🌐 Mainnet	
					OTHERS	
					🗹 Custom	
📰 node						匬
🔎 😐 Development 🛛 🗮 Transactions	i≡ ABI Storage					

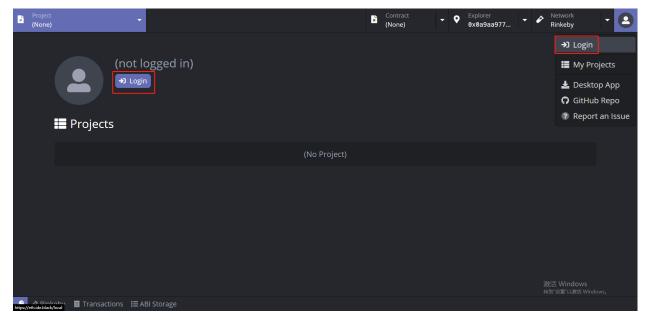
CHAPTER

THREE

QUICKSTART

3.1 Login web client

Click the "Login" icon on the bottom of "not logged in". Otherwise, click the upper right corner and click the "Login" on the panel.



Users login through their Github accounts, in the Ethereum Studio. Once logged, users log again through Github information automatically. The Ethereum Studio will not keep users' accounts information.

3.2 Create an ERC20 project

After login, create a new project by clicking the green button with "New".

Proj (Nor		-		Contract (None)	- 9	Explorer 0x0a9aa977	- 6	Network Rinkeby	- 🥘
		HanYouya (No description							
	📕 Projec	ts						+ New	
	v2 A HanYouya	ang/v2							
	test 🌰 HanYouya	ang/test							
	testERC20) ang/testERC20							
								激活 Windows 转到"设置"以激活 Windo	
🤌 🖉 R	inkeby 📕 Transa	ictions 🛛 🔚 ABI Stora							

Create an ERC20 project in Ethereum Studio. Set "Project" as "tokenTransfer" and "Template" as "ERC20 Token". The project will be automatically saved in Ethereum Studio's cloud under one's account. Then click the purple button at the bottom left corner to create.

(None)								
(None)			Create a New Project		×			
		HanYo	,					
			tokenTransfer					
			Template					
	Project	ts	ERC20 Token					
				Cancel	Create Project			
	testERC20							
							数活 Windows ^{表到"设置"} 以激活 Windo	
🤌 🔗 Rink						🌶 英 ; 🍨 📟 👪		

After the project is created, the ERC20 contracts have been generated successfully.

Project HanYouyang/tokenTransfer	•				× Contract (None)	-	Explore 0x0a9	er aa977	- 6	Network Rinkeby	- 🙆
+ 🎢 🖆 🌾	٠	C README.md	🗅 ERC20.sol 🗙	C READMD.md	+						
✓ tokenTransfer											A STRATEGICAL STRATE.
			1:1:1: 00 0 0								No.121 COM 447 ENGLASSINGLASSING EXCHANGE SALASSING ROMAN SALASSING
Context.sol		3 pragma sc 4									
ERC20.sol		5 import".	/IERC20.sol";								1018 1036
IERC20.sol			/IERC20Metadata.so								THE CONVERSION AND ADDRESS
IFRC20Metadata.sol		7 import". 8	/Context.sol";								Participation Concerning
 migrations 											
0											Beneral and an
1.deploy.js											None and N
✓ scripts											Service and and
deploy.js				<pre>has to be added in see {ERC20PresetMi</pre>		ract using	g {_mint}.				JAMANOT
waffle-deploy.js		15 *									And and and a second and a seco
READMD.md											Hard Control of the second
README.md											A second
config.json											Write and the second second second
truffle-config.is			e followed general	l OpenZeppelin quid		ns revert					
0,											
											Automotive and a second and as second and a
				ents. Other impleme							
										激活 Windows	10055:princi
										病/占 Windows 转到"设置"以激活 W	
🤌 🖉 Rinkeby 🗒 Transactions 🗄	E ABI S	Storage								≯ s	olc (0.8.0) >_

3.3 Connect to MetaMask Account

A MetaMask wallet pop up in the web client to login through users' passwords. Suppose a user does not have a MetaMask account. Please refer to the MetaMask register introduction.









密码



After linking to the MetaMask, copy the mnemonic into the Keypair Manager for later deploying contracts. Click the purple key icon on the bottom left corner and then click the "Import" button and paste the mnemonic words, 12 English Words representing the private key. Finally, give this imported private key a name, and the Etereum Studio will save it under this account in the cloud.

3.4 Request Ropsten Faucet

After choosing the "Network" and "Explorer", click the upper right "Faucet" icon and get the request page of testETH on the testnet. Choose "Ropsten" testnet and click the icon to turn to the request page.

Project HanYouyang	'tokenTransfer				×	Contract (None)	- •	Explorer 0x0a9aa97	7 🔽 🖌	Network Ropsten	-	2
New Tab	0x0a9ab7b6	< +										
C 🔍 0x0a9a	aa9771e347d616978	6ecbe19db71119c9b7b	6								☆ ¢	ی ا
Account			Informati	ion								
🖸 Balance		2.875 ETH									(Non	e)
# Nonce		24										
Transacti	ons											
тіме												
				No Transactions	s Found							
🤌 🔗 Ropsten	🗒 Transactions ᠄ 🗄	ABI Storage										

Click the upper right "Faucet" icon, and the page will jump to the Ropsten Faucet. One can copy the wallet address in MetaMask and paste it into the box. Then clicks the button "Send me test Ether" and wait for a few minutes.

Ropsten Ethereum Faucet								
Enter your testnet account address 0x0a9AA9771E347D6169786EcBE19db71119C9b7bd Send me test Ether								
Description Last deposits 0x02tf559b382447f491235d7fd8523984aaca68a 0x32tfc4cc2653d92296c892ftb21baa8c7cc3f9 0x517a489c27a99b338414260b30453a9c205297c 0x68cc22tca3e1fbae1ee499b685aaf9c3d90b6eec 0x4947aef6a3b0cd988ad87c293775s65c8e0a637ie								
This faucet drips 0.3 Ether every 10 seconds. You can register your account in our queue. Max queue size is currently 50. Serving from account OxcDA0D6adCD0fICCeA6795F9b1F23a27ae643FE7C (balance 0 ETH).								
For inquiries, support or just to say thanks please reach out to us on Twitter								

With all the requests finished, there will be 5 testETH in the wallet address on Ropsten test network.

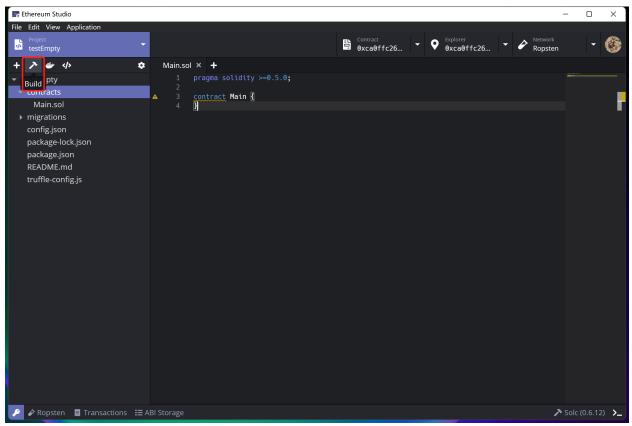
×	● Ropsten 测试网络 ~							
0 未连接	Account 1 0x0a9b7b6							
:								
	5 ETH							
・	发送 発換 Swap							
资产	活动							
s eth	>							
D	Don't see your token? Import tokens							
Need help?	Contact MetaMask Support							

Back to the web client, check the balance if there is 5 testETH on the test network, Ropsten.

	Project (None)						×	Contract (None)	- 9	Explorer 0x0a9aa97	, -	 Network Ropsten 	• 🕘
Ne	w Tab	0x0a9ab7b6	× +										
С (Q 0x0a9a	a9771e347d61697	86ecbe19db	71119c9b7b6	5								🔄 🗘 🚓
Ac	count				Informat	tion							
C	Balance			5 ETH									(None)
#	Nonce			0									
Tra	ansactio	ons											
тім			тх назн										
						No Transactio	ons Found						
۶.	🖍 Ropsten	Transactions	■ ABI Storage										

3.5 Build Contracts

There is three Solidity files in the "ERC20/contracts/" file in the left panel after successfully creating the project. Select one of the contract and click the "Hammer" icon to build the contract.



Now this IDE only supports building all contracts together. Single-file compilation will come soon in the later version.

 ★ ★ ★ ↓ CREADME.md JERC20.sol KERC20.sol J // SPDX-License-Identifier: MIT pragma solidity ^0.8.0; contracts contracts contracts import "./IERC20.sol"; import "./IERC20.sol import *./contracts/REC20.sol import	Project HanYouyang/tokenTransfer	Contract ← Custorer →	 Network Ropsten Ropsten
<pre>> build > build > contracts Contracts Context.sol ERC20.sol II * Odev Implementation of the {IERC20} interface. II * This implementation is agnostic to the way tokens are created. This means * that a supply mechanism has to be added in a derived contract using {_mint}. * TIP: For a detailed writeun see our quide * TIP: For a detailed writeun s</pre>	+ <i>> 👄 </i> 🔶	□ README.md □ IERC20.sol □ ERC20.sol × +	
config.json Config.json truffle-config.js [soljson-v0.8.0+commit.c7dfd78e.js] solcjsbin ./contracts/ERC20.sol Warning: This declaration shadows an existing declaration. > ./contracts/ERC20.sol:55:5: 55 uint256 totalSupply Note: The shadowed declaration is here: ->> /contracts/ERC20.sol:97:3: 97 function totalSupply() public view virtual override returns (uint256) {	 build contracts contracts context.sol ERC20.sol IERC20.sol IERC20Metadata.sol migrations scripts READMD.md 	<pre>2 3 pragma solidity ^0.8.0; 4 5 import "./IERC20.sol"; 7 import "./IERC20Metadata.sol"; 7 import "./Context.sol"; 8 9</pre>	
P ≥ Ropsten ☐ Transactions ABI Storage	config.json truffle-config.js	Compiler [soljson-v0.8.0+commit.c7dfd78e.js] solcjsbin ./contracts/ERC20.sol Warning: This declaration shadows an existing declaration. > ./contracts/ERC20.sol55:5: 55 uint256 totalSupply Note: The shadowed declaration is here: > ./contracts/ERC20.sol:97:3: 97 function totalSupply() public view virtual override returns (uint256) {	

3.6 Deploy Contracts

After a successful building, deploy the ERC20 contract on the Ropsten testnet. Select the JSON file "ERC20.json".

Project HanYouyang/tokenTransfer				
	Deploy Contract ERC20	×		_
 build contracts 	Compiled contract (compiler output JSON)			
▼ contracts	ERC20.json	•		
Context.sol ERC20.sol	Context.json			
IERC20.sol IERC20Metadata.sol	ERC20.json IERC20.json			
 migrations scripts 	IERC20Js01			
READMD.md		UTF8		
README.md config.json	symbol_			ŵ
truffle-config.js	TEST			
		UTF8		
	totalSupply			
		Cancel Estimate & Deploy]	
🤌 🖉 Ropsten 🛛 🗒 Transactions 🗄	E ABI Storage		,	4 💷 👪

Then set "name_" as "test", "symbol_" as "TEST" and totalSupply as "3000000000".

Project		Contract	🚽 🥥 Explorer	✓ ♦ Network	- 🙉
HanYouyang/tokenTransfer ↔	Deploy Contract ERC20				
✓ tokenTransfer ✓ build	Compiled contract (compiler output JSON)				
▶ contracts	ERC20.json		-		Market (Market) Market
 ⊂ contracts Context.sol ERC20.sol 	Constructor Parameters				Barton Series Constraints and a series of the ser
IERC20.sol IERC20Metadata.sol ▶ migrations	test				The second secon
▶ scripts			UTF8		
READMD.md README.md	symbol_				
config.json truffle-config.js	TEST				Û
trume-config.js			UTF8		
	totalSupply				
	123 300000000				
		Cancel	Estimate & Deploy		
🔎 🖉 Ropsten 🗒 Transactions 🖽 A					Solc (0.8.0) 🔪

Select the signer as the keypair saved in Keypair Manager. If there is no choice, please check the chapter **Keypair** Manager.

Project HanYouyang/tokenTransfer					
+ > + + + + + + + + + + + + + + + + + +	Deploy Contract ERC20				
 ▼ tokenTransfer ▼ build 			UTF8		
 ▶ contracts ▼ contracts 	symbol_				
Contracts Context.sol ERC20.sol	TEST				
IERC20.sol IERC20Metadata.sol			UTF8		
 migrations 	totalSupply				
 scripts READMD.md 	123 300000000				
README.md	Signer				
config.json truffle-config.js	0x0a9aa9771e347d6169786ecbe19db71119c9b	o7b6	han 👻		
	Gas L 0x0a9aa9771e347d6169786ecbe19db71119c9b	o7b6	han		
	Default: 1,000,000	e 💰 m			
		Cancel	Estimate & Deploy		
🔎 🖉 Ropsten 🗒 Transactions 🖽 AB	ll Storage			P	Solc (0.8.0) >_

Set the "Tip" as desired amount, or generate it with "Gas Limit" and "Max Fee" together by clicking the bottom right button to "Estimate". There will be the estimated fee in real-time. If the fee is not reasonable, click the "Re-estimate" or wait for a non-congestion period.

Project HanYouyang/tokenTransfer		s	Contract	🚽 🔮 Explorer	- P Network Ropsten	- 🔊
+ > + + + + + + + + + + + + + + + + + +	Deploy Contract ERC20					
✓ tokenTransfer ✓ build				UTF8		
▶ contracts ▼ contracts	symbol_					
Context.sol	TEST					
ERC20.sol IERC20.sol IERC20Metadata.sol				UTF8		
 migrations 	totalSupply					
 scripts READMD.md 	123 300000000					
README.md	Signer					
config.json truffle-config.js	₽ 0x0a9aa9771e347d61	L69786ecbe19db71119c9b7b	06	han 🔻		
	Gas Limit	Тір	Max Fee			
	o Default: 1,000,000	2 max priority fee pe	š max			
			Cancel Es	stimate & Deploy		
🔎 🖉 Ropsten 🗒 Transactions 🖽 A						

After deploying successfully, a window with detailed transaction information will pop up. Now the contract has been deployed at the address: 0xF30438E789b361Eca03B3C7AB8cB176e436C7259. Click the address to turn to the "Contract" interface.

Project HanYouyang/tokenTransfer					- 🔎
+ > => +>	Deploy a Contract				
 ✓ tokenTransfer ✓ build ▶ contracts 	Basic Parameters Tx	Receipt ABI			A more restriction of Total and a second second second Second Second Second Second Second Second Second Second Second Second Second
	# Hash	0x5c40ff3255c9a60ca8f4	8296af48f88a1108be393e8d4232	3a983806a892ee16	
ERC20.sol IERC20.sol	O Status			CONFIRMED	The second secon
IERC20Metadata.sol ▶ migrations	🖺 Contract		0xF30438E789b361Eca03B3C7A	B8cB176e436C7259	
 scripts READMD.md 	🖺 Contract Name			ERC20	e a construir de la construir
README.md config.json	🔮 ETH Sent			0.0 ETH	<u>ل</u>
truffle-config.js	👂 Signer		0x0a9aa9771e347d6169786ecb	e19db71119c9b7b6	
				Close	
🔎 🔗 Ropsten 🛛 🗒 Transactions					S # ', 🎍 🖩 🖁

3.7 Check Balance and Transfer

Since 5 test ETH is far enough for paying fees, choose the "transfer" function from the purple inverted triangle icon. Select the "recipient" as the target address and set the token amount to transfer as 1000. Click the purple "Estimate" button to set "Parameters" and "Gas" automatically.

Project HanYouyang/tokenTransfer			B Contract 0xF30438E7 ▼ ◆ Explorer han	▼ 🖋 Network 🗸 🖉
0xf3047259 × +				
C • 0xF30438E789b361Eca03B3C7AB8	3cB176e436C7259			☆
f⊗ transfer 🝷 🕨	👁 allowance 🔻 🕨	🛗 Approval 👻 🕨		
Parameters	Parameters	Parameters		
recipient	owner	Range		
• 0x0a9aa9771e347d6169786e han	Select or type an address			Clear
amount	spender	Event Logs		
123 1000	Select or type an address	BLOCK	SPENDER	
Gas	Result			
Gas Limit	(None)		(no data)	
o 30592				
Тір				
250000000				
Max Fee				
<u>a</u> 250000016				
Authorization				
Signer				
0x0a9aa9771e347d616978 han				
Result				
(None)				
🥟 🖉 Ropsten 🗒 Transactions 🖽 ABI Sta				S 🗸 🖓 🍨 📟 👪

After all the parameters are settled, click the triangle icon beside "transfer" to execute a function.

Project HanYouyang/tokenTransfer			Contract 0xF30438E7 Contract han	 Network Ropsten 	- 🅘
0xf3047259 × +					
C 🛛 0xF30438E789b361Eca03B3C7AB8c	B176e436C7259				
fω transfer 🔹 🕨	👁 allowance 👻 🕨	🛗 Approval 👻 🕨			
Parameters Execute	Parameters	Parameters			
recipient	owner	Range			
♥ 0x0a9aa9771e347d6169786e han	Select or type an address				Clear
amount	spender	Event Logs			
123 1000	Select or type an address	BLOCK	SPENDER		
Gas Estimate	Result				
Gas Limit 30592 Tip 2500000000 Max Fee 2500000016 Authorization Signer Øx0a9aa9771e347d616978 ₪mm Result (None)	(None)		(no data)		
🥟 🖉 Ropsten 🗒 Transactions 🖽 ABI Stor					

After the compeletion of "PUSHING" state, the contract is deployed successfully with the "CONFIRMED" state. Select "Explorer" on the upper right panel to see the past transaction detail.

Project HanYouyang/tokenTransfer			B	Contract	- P Explorer han	- P Network Ropsten	- 🙆
0xf3047259 × +		Call a Contract					
C							
f‰ transfer 👻 🕨		Basic Parameters Tx Rece	ipt Result				
recipient • 0x0a9aa9771e347d6169781		# Hash 0xt	524bce739042c9df02ed81c92664f710bdd039a6466e824c50a	0a8cf5a1df1baf			
amount		O Status		CONFIRMED			
123 1000		🖺 Contract	0xF30438E789b361Eca03B3C7AB8c1	:B176e436C7259			
Gas Estimate		_ contract					
Gas Limit		f ⇔ Function		transfer			
3 0592		💣 ETH Sent		0.0 ETH			
Tip							
250000000		🔎 Signer	0x0a9aa9771e347d6169786ecbe190	db71119c9b7b6			
Max Fee 500000016							
250000016				Close			
Authorization							
Signer							
Øx0a9aa9771e347d61697 han							
Result Pretty Raw							
(1 return							
true bool)							
🤌 🖉 Ropsten 🗒 Transactions 🖽 ABI Store	age						

After transferring the 1000 "TEST" token, click the upper right "Explorer" to see the left balance of 4.997 test ETH now. The balance consumed by the "Gas Fee" and "Tip" will not be 5 testETH for deploying and calling a contract.

Project HanYouyang/tokenTransfer				Contract 0xF30438E7	👻 🔮 Exp har	n volorer	Network Ropsten	- 🍳	
♦han × +									
C (9 0x0a9aa9771e347d6169786ec	be19db71119c9b7b6						습)¢ #	•
Account		Information							
🗇 Balance	4.997 ETH						6	(None)	
# Nonce									
Transactions									
TIME BLOCK			то ∴ Loading						
🎤 🖉 Ropsten 🗒 Transactions 🖽 AB	3I Storage								

Here is the most simple quickstart example of the Ethereum Studio. Please feel free to ask us any questions through Github issue link.

CHAPTER

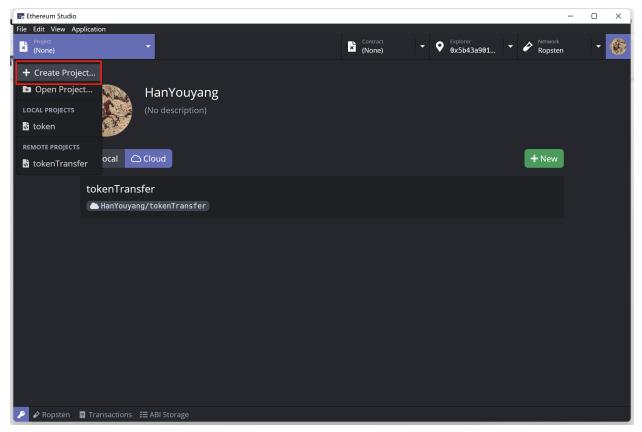
FOUR

PROJECT

Developers can create a new project, open projects and check all the local and remote projects in the upper left corner "Project" panel.

4.1 Create Project

Click "Create Project" and create a new project in either local or cloud under account. The created projects will show in both desktop and web clients in real-time. Creating a project and editing it can be done solely in the Ethereum Studio without other development tools like Visual Studio Code.



In "Create a New Project" panel, developers input the "Project name", while the "Project location" will be automatically generated a document path following "C:\Users\Administrator\Ethereum Studio" in Windows.

Ethereum Studio					- 0	×
File Edit View Application						
None)		Contract		• Explorer		
	Create a New Project			×		
	Local					
	Project location					
				Choose		
c 🖵 Local 🤇	Project name					
tokenTran	testTransfer					
	Template					
	Coin			-		
	Framework	Truffle	version			
	Truffle Hardhat Waffle Dockerized Truffle	v5.4.6				
	Npm client					
	npm yarn cnpm					
			Cancel	Create Project		
🔎 🔗 Ropsten 🛛 🗒 Transactions	🚍 ABI Storage					

Developers can change the path to any other documents. Click the "Choose" button on the left of the default path, and one will see all documents in the whole computer and choose another existing document or create a new document as the new project location. Be careful that the new project location must be an empty file.

	Ethereum Stud	io															- (×
	选择文件夹	ſ												>				
	$\leftrightarrow \rightarrow \sim$	\uparrow	🗅 > Adminis	trator > Ethe	reum Studio					C	搜索"Eth			م				
	组织 - 新建	文件夹												- ?				
	🕨 🚖 快速访问		名称			修改	收日期	类	型	大小								
	🗢 OneDrive	Dore	🧀 testT	ransfer		202	21/12/28 15:4	42 文	件夹									
		- Pers																
á i	▶ 🐟 WPS网盘																	
) .	🖌 💻 此电脑																	
C	> 🗾 视频																	
	> 🔀 图片																	
	> 🔤 文档		1															
o	> 业下载																	
	> 🕑 音乐 > 💷 卓而																	
4	2 Section		_{‡夹:} testTran:	sfor														
		×1	+								Ope	en	TH	消				
															.::			
				npm y														
										C	Cancel	Cre	eate Pr	oject				
C																		

4.1.1 Frameworks

Developers can choose 4 different frameworks: Truffle, Hardhat, Waffle and Dockerized Truffle. The detail information about those frameworks can be checked in "Supported Frameworks" under "Reference" of this doucument.

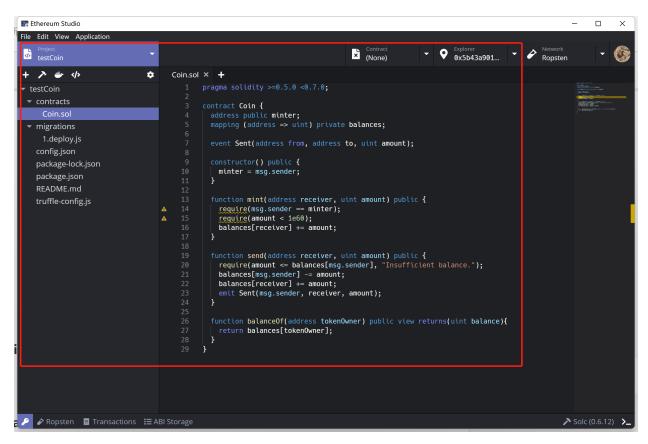
4.1.2 Template

In "Template", there are several different templates including "Empty", "Coin", "ERC20", "Basic" and "Metacoin".

In "Empty Project" template, there is a default empty project with contracts "Main.sol". The only "Main.sol" contract has no content. This template is a minimum viable product to build smart contracts from scratch.

F Ethereum Studio					-	- 0	×
File Edit View Application							
Project testEmpty		Contract (None)	- Ox5t	er 43a901 👻	Network Ropsten	-	۲
 * * * * * * * * * * * * * * * * * * *	Main.sol × + 1 pragma solidity >=0.5.0; 2 ▲ 3 contract Main 1 4						F
🕑 🔗 Ropsten 📋 Transactions 注目 Af	31 Storage				• جر	5olc (0.6.12	2) >_

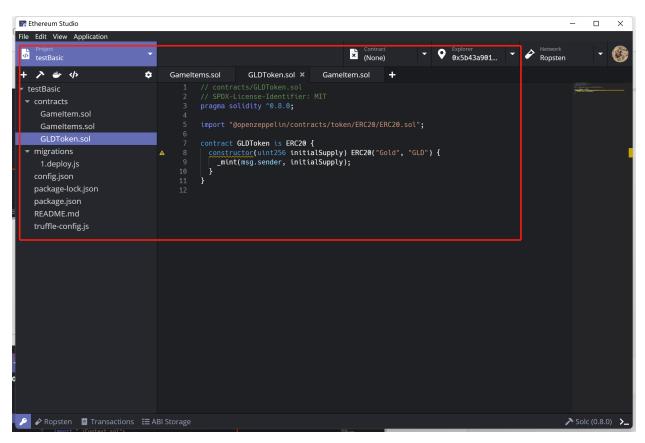
In "Coin" template, there is a default empty project with contracts "Coin.sol". This template is a primary contract with variables and functions that could help developers build coin-related contracts. Please remember that a definition of coin functions creates this contract and it does not follow any ERC standards.



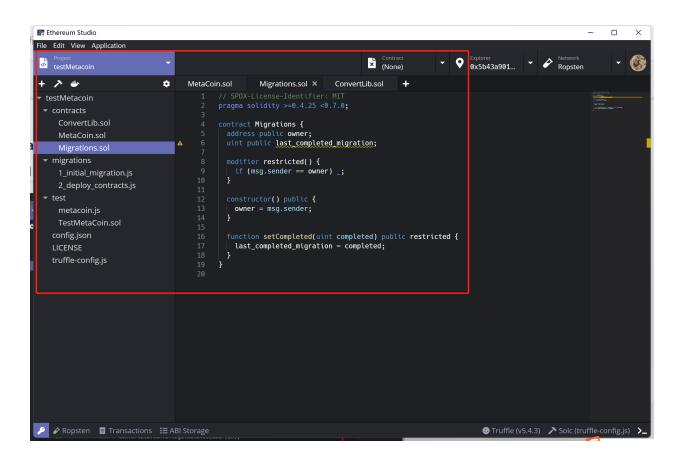
In "ERC20 token" template, there is an ERC20 standard token project including "ERC20.sol". The "ERC20.sol" contract has all functions the ERC20 standard required. There are also "IERC20" interface. Using the interface would reduce work of setting parameters and provide reliable third-party essential functions for developers to call. This template and generates ERC20 standard related contracts and interfaces locally.

F Ethereum Studio		- 🗆 ×
File Edit View Application		
Project testERC20	Contract V Explorer State None) V Explorer State None	
+ > 🐡 🚸	ERC20.sol × +	
▼ testERC20		
 contracts 		
Context.sol	3 pragma solidity ^0.8.0;	
ERC20.sol	5 import "./IERC20.sol";	
	6 import "./IERC20Metadata.sol";	Type has been been as a property of the second seco
IERC20.sol	7 import "./Context.sol";	
IERC20Metadata.sol		
 migrations 	9 /** 10 ★ @dev Implementation of the {IERC20} interface.	
1.deploy.js	10 * @dev Implementation of the {IERC20} interface. 11 *	
config.json	12 * This implementation is agnostic to the way tokens are created. This means	The second secon
package-lock.json	13 \star that a supply mechanism has to be added in a derived contract using {_mint}.	
package.json	14 * For a generic mechanism see {ERC20PresetMinterPauser}.	
README.md		
	16 * TIP: For a detailed writeup see our guide 17 * https://forum.zeppelin.solutions/t/how-to-implement-erc20-supply-mechanisms/226[How	
truffle-config.js	17 * Inters.//Torum.zeppetin.solutions/i/now-to-implement-erczy-supply-mechanisms/zzo_now 18 * to implement supply mechanisms].	A to the first second s
	20	
	21 * of returning `false` on failure. This behavior is nonetheless conventional	
	22 * and does not conflict with the expectations of ERC20 applications.	
	23 * 24 * Additionally, an {Approval} event is emitted on calls to {transferFrom}.	
	25 * This allows applications to reconstruct the allowance for all accounts just	
	26 * by listening to said events. Other implementations of the EIP may not emit	
	27 * these events, as it isn't required by the specification.	
	29 * Finally, the non-standard {decreaseAllowance} and {increaseAllowance}	
	30 * functions have been added to mitigate the well-known issues around setting 31 * allowances. See {IERC20-approve}.	And an and a second sec
	$31 \times \text{actowances}$. See {ILRC20-approve}.	
	33 contract ERC20 is Context, IERC20, IERC20Metadata {	
	<pre>34 mapping(address => uint256) private _balances;</pre>	
	35	Annual and a second sec
👂 🖉 Ropsten 🗒 Transactions ᠄ 🖃	ABI Storage	➢ Solc (0.8.0)

In "Basics - ERC20, ERC721 & ERC1155(v3.1+)" template, there are three ERC standard contracts including "GLDToken.sol", "GameItem.sol" and "GameItems.sol". Those three smart contracts inherited ERC20, ERC721 and ERC1155 standards respectively. All three projects import ERC standards through Open Zeppelin and realized only constructor function with basic parameters. Developers can use those smart contracts to understand how the ERC20 tokens and ERC721 and ERC1155 NFTs are minted. The "Basic OpenZeppline Template" will use interfaces online and developers only need to import ERC standard contracts through code.



In "Metacoin" template, there is a default framework "Dockerized Truffle" including "ConvertLib.sol", "MetaCoin.sol" and "Migrations.sol". In this project, developers choose "Dockerized Truffle" since ???



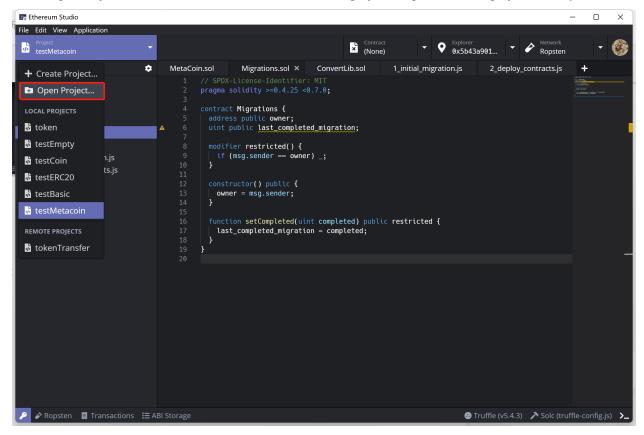
4.1.3 npm Clients

npm(Node Package Manager) stems from when npm first was created as a package manager for Node.js. All npm packages are defined in files called package.json. The content of package.json must be written in JSON. At least two fields must be present in the definition file: name and version.

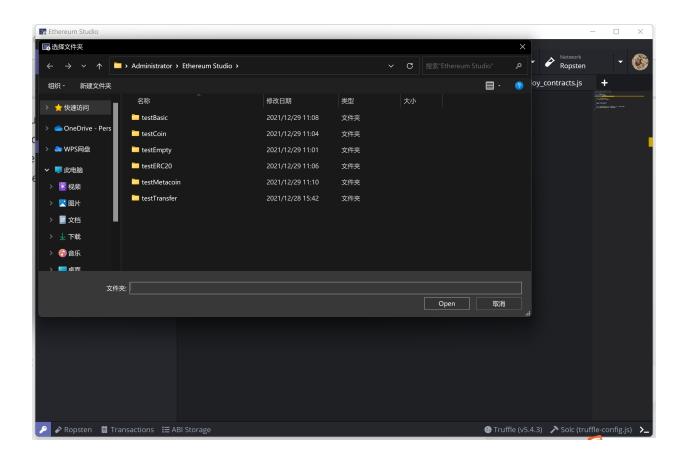
cnpm is faster than npm in China, because Taobao first requests the contents of foreign servers to its own domestic servers, so when we use cnpm, the download depends on downloading from domestic servers, which is much faster. It has a complete image of npmjs.org. At present, the synchronization frequency is once every 10 minutes to ensure that it is synchronized with the official service as much as possible.

yarn offers offline mode. If developers have installed a package before, developers can install it again without any Internet connection. Yarn has a lock file that records the exact version number of the installed module. Each time a file is added, yarn will create (or update) the yarn.lock file to ensure that the module version is the same each time the dependency is installed. Yarn reduce the different versions of dependent packages to a single version to avoid creating multiple copies.

4.2 Open Project

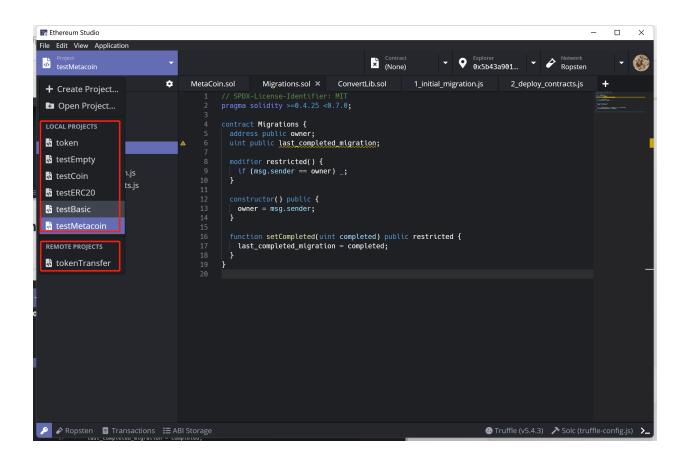


Click "Open Project", one can check the location of the current project or open an existed project in the system.



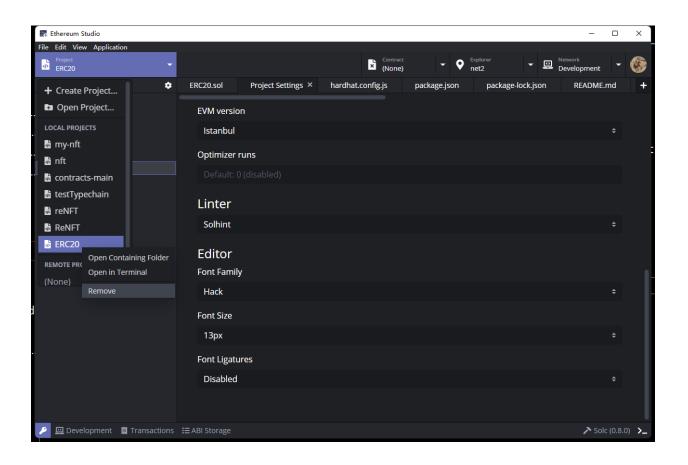
4.3 Local Projects and Cloud Projects

In the "Local Projects" and "Remote Projects" of the "File" panel, the user can see all the projects in local computer and cloud under one's account. Developers can quickly switch to another project through panel that is helpful for muti-project developers.



4.4 Delete Project

Right click the name of any project in the panel, developers can chose "Remove" and the project is deleted immediately. Be careful since the deleted project can not be restored from "Trash Can" on the desktop.



CHAPTER

FIVE

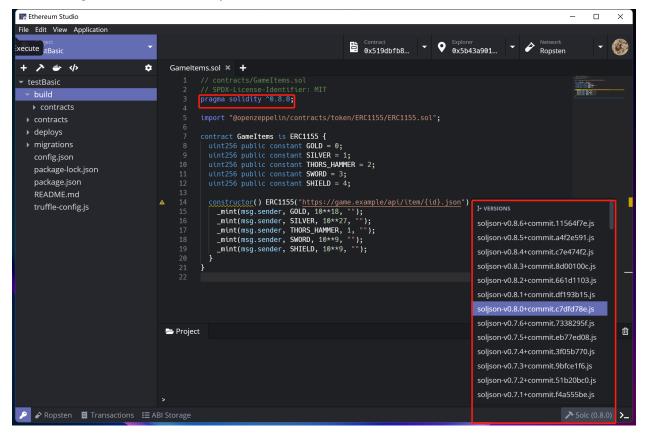
EDITOR

5.1 Build

After test and develop, developers needs to build smart contracts.

5.1.1 Build Preparation

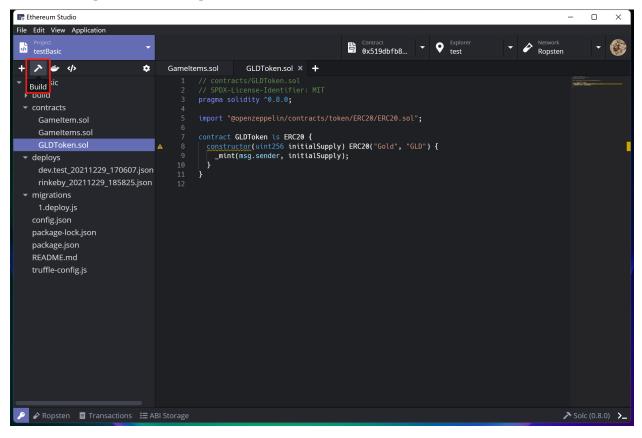
When an existed project is opened, the Ethereum Studio can automatically detect the solidity version written at the head of contracts. Developers can select the specific version of the solidity compiler by clicking the "hammer" icon at the bottom right corner with "Solc(0.x.y)".

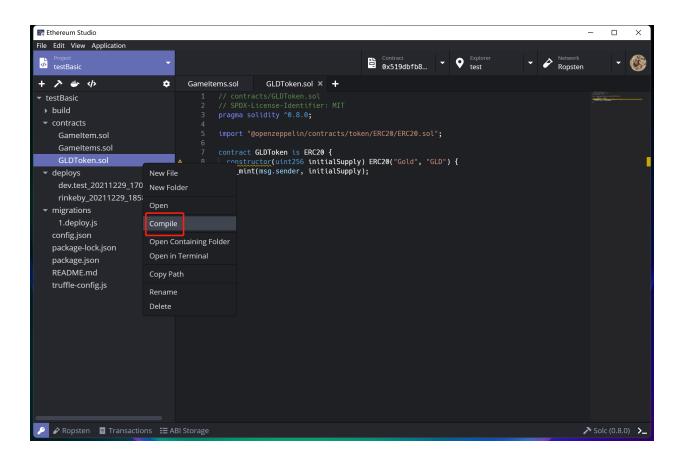


5.1.2 Build by Panel

In the desktop client, developers can click the "hammer" icon below the "Project" panel or right-click the target file and select "Compile" to build the contract. Now the Ethereum Studio only supports building all contracts together, and the single file building will be released later.

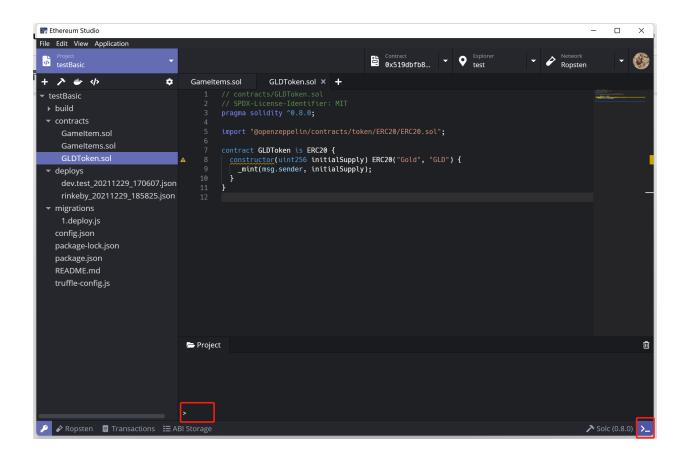
Developers can find differences between web and desktop clients on building a project. The detail explanation can be checked in **Chapter Install Desktop Client, Contrast with Web Client**.





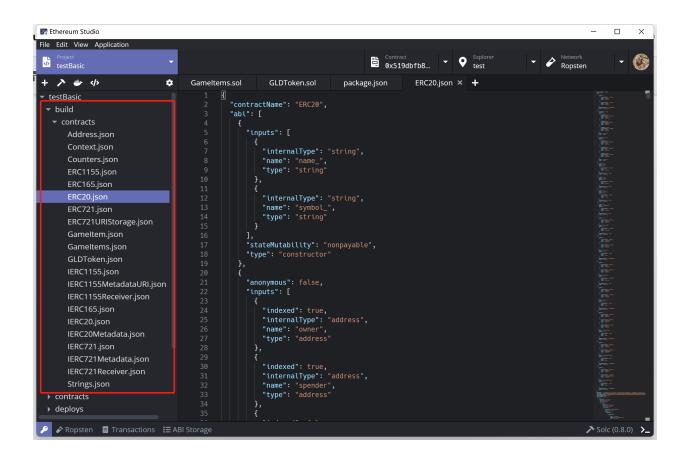
5.1.3 Build by Command Line

Developers can build contracts manually by opening the command line, clicking the "Terminal" icon and inputting the command through "Project" panel. Please note that the command must be corresponding to the selected framework during creating process. Developers can check the framework type in "package.json" with the commands following "scripts".



5.1.4 Check Building Details

After building successfully, the framework will build a new document named "build" containing a "contracts" document. In "contracts" document, there are all the JSON files generated in the building process. Each JSON file has the corresponding application binary interface(ABI). Developers will deploy ABI files on the network later.

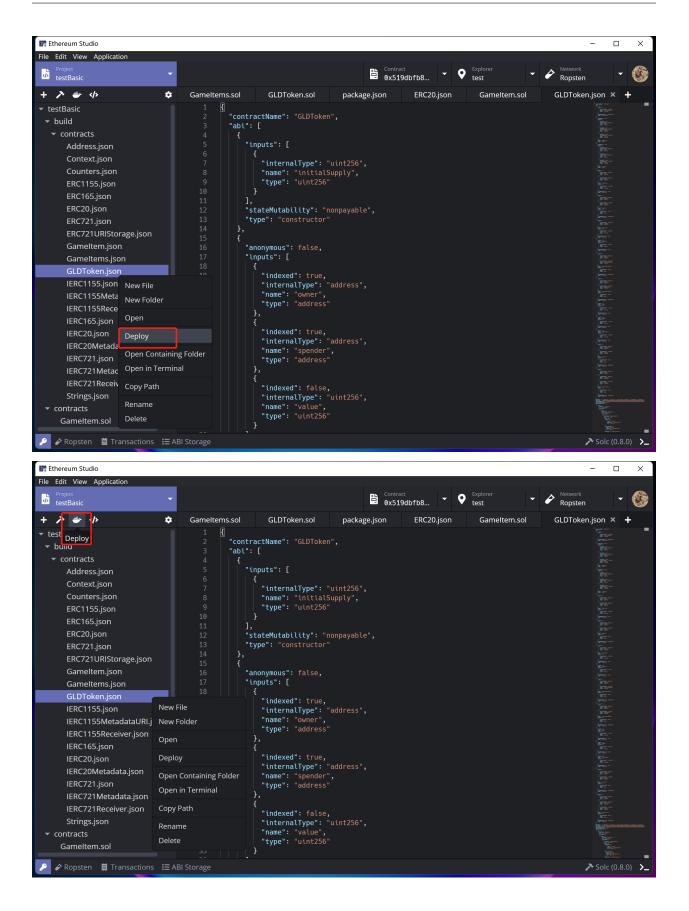


5.2 Deploy

After building process, developers can deploy target contracts with generated ABI files.

5.2.1 Deploy by Panel

Developers can right-click the file name of an contract and select "Deploy", and there will popup a "Deploy Contract" window. By other means, developers can click the "Docker" icon below the "Project" panel.



5.2.2 Deploy by Command Line

Developers can open the command line to deploy contracts manually by clicking the "Terminal" icon and input command through the "Project" panel. Please note that the commands must be corresponding to the selected framework during project creating process. Developers can check the framework in "package.json" with commands following "scripts".

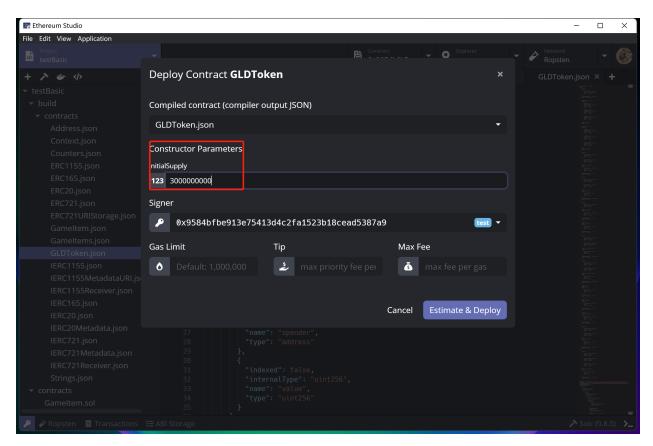
5.2.3 Deploy Preparation

There will be a "Deploy Contract" window in the preparation process. Developers can choose a JSON file to deploy on the network in this window.

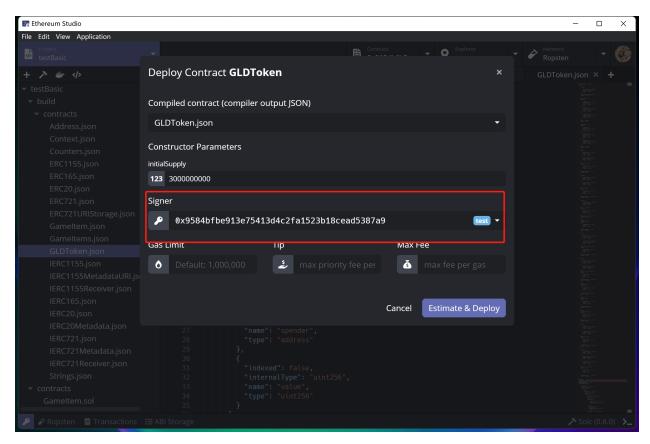
🛃 Ethereum Studio						-	×
File Edit View Application							
				- O Explorer			
	Deploy Contract GLDT	ken			× GI		
	Compiled contract (compiler	output JSON)					
	GLDToken.json						
	Constructor Parameters						
	initialSupply						
	123 300000000						
	Signor						
	Signer						
	0x9584bfbe913e754	13d4c2fa1523b18	cead5387a9	test) -		
	Gas Limit	Тір	Ma	ix Fee			
	o Default: 1,000,000	🛓 max prior	ity fee per	max fee per gas			
			c				
			Cance	Estimate & Depl	loy		

"Constructor Parameters" are corresponding to the constructor function parameters for users to input.

Ethereum Studio Docs

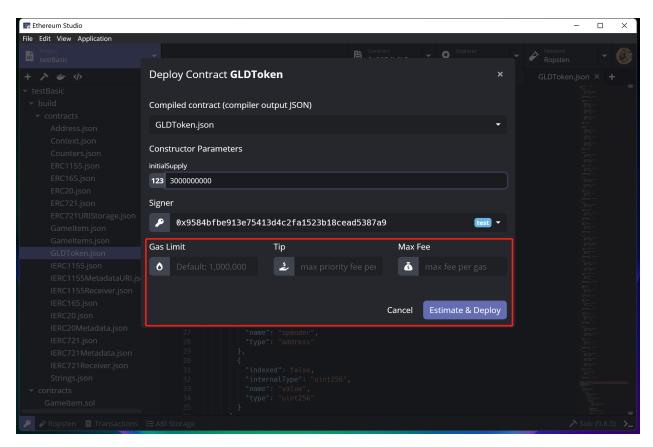


Developers can choose "Signer", the final payer of gas fee and tip for deploying this contract. Developers needs to ensure the signer has enough ETH on the target network. Otherwise, the later "Estimate & Deploy" process will not be successful.

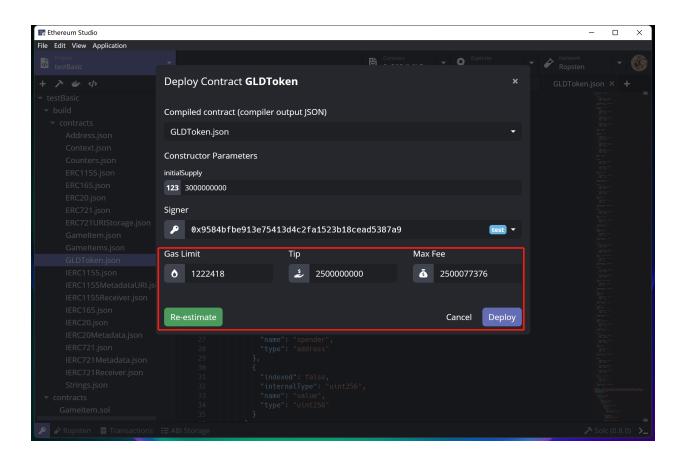


The "Estimate & Deploy" button locates at the bottom right corner. There will be a real-time estimation of "Gas Limit", "Tip", and "Max Fee" located below. It will be significantly different for the estimation time and price. Developers should carefully check the network gas fee before deploying.

Ethereum Studio Docs

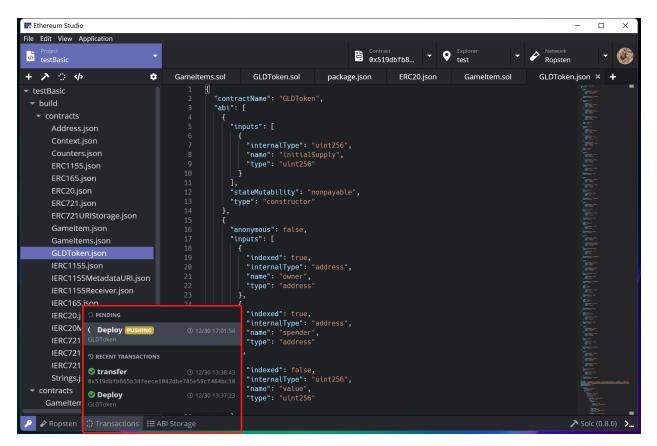


After estimating process, there are exact price numbers in each box. If developers feel the gas fee price is too high or the network is too busy, click the green "Re-estimate" button at the bottom left corner to estimate cost again. If developers supposes the price is fair, click the purple "Deploy" button on the bottom right corner to deploy the contract.

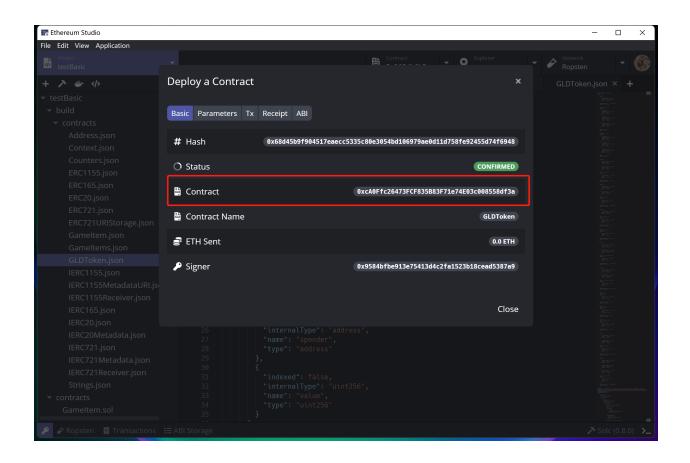


5.2.4 Check Deploy Transactions

After clicking "Deploy", developers can check deploy schedule by clicking the bottom "Transactions" button to review any transaction. Developers can check detailed information with the popup "Deploy a Contract" window.

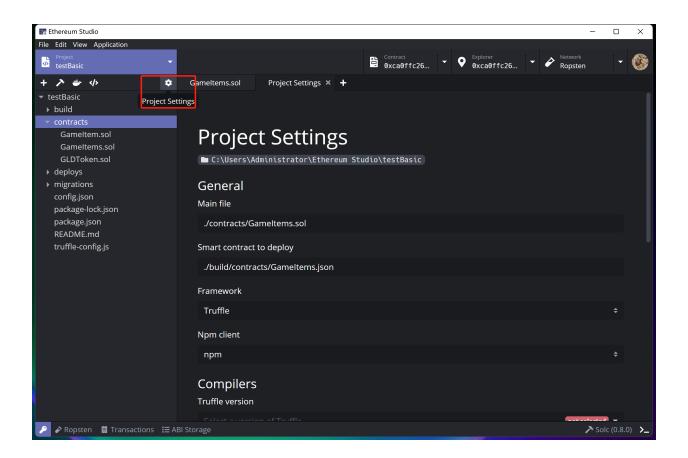


Five boxes show detailed information in the "Deploy a Contract" window. In the "Basic" panel, there are several most crucial pieces of information of deployment, including address. Developers can click on the "Contract" address, and the "Contract" panel will show the contract functions for developers to call.



5.3 Project Settings

At the right end of the toolbar, there is a "gear" icon named "Project Settings". Click the icon, and there will be the "Project Settings" panel in the editor. This panel is a graphic show of the "config.json" file. Developers can easily change settings in the project.



5.3.1 General

In the "General" part, the "Main file" is the default selection of deploying file.

Developers can switch the framework here and then use a new framework to build or deploy with more detailed configuration in "package.json".

Developers can change clients here and use a new client like yarn or cnpm. Please make sure that developers has installed the clients and runs directly in the command line.

F Ethereum Studio	- O X
File Edit View Application	
testBasic	🕒 Contract 🗸 V Sca0ffc26 V Sca0ffc26 V Ketwork Vetwork
+ > 🐡 🚸 💠	Project Settings × +
 ▼ testBasic > build ▼ contracts GameItem.sol GameItems.sol GLDToken.sol ▼ deploys 	Project Settings C:\Users\Administrator\Ethereum Studio\testBasic
dev.test_20211229_170607.json rinkeby_20211229_185825.json ropsten_20211230_170532.json ▶ migrations config.json package-lock.json	Main file
package.json README.md	./build/contracts/Gameltems.json
truffle-config.js	Framework
	Truffle ÷
	Npm client
	npm ÷
	Compilers
	Truffle version
🥟 🎤 Ropsten 🗒 Transactions 🖽 A	El Storage → Solc (0.8.0) >

5.3.2 Compilers

Developers can select the Truffle version while the default version is settled during the creating process. The "Solc(0.x.y)" version is identical to the pragma version in the head of solidity file. Developers can change the EVM version and Optimizer directly. The Ethereum Studio disabled Optimizer???

🛃 Ethereum Studio			×
File Edit View Application			
testBasic	B Contract 0xca0ffc26 V P Dxca0ffc26 V P Rops		
+ > 🐡 💠	Project Settings × +		
 ▼ testBasic > build ▼ contracts Gameltem.sol 	npm Compilers	¢	
Gameltems.sol GLDToken.sol	Select a version	t selected 🔻	
dev.test_20211229_170607.json rinkeby_20211229_185825.json	Solc version soljson-v0.8.0+commit.c7dfd78e.js	Ţ	
ropsten_20211230_170532.json ▶ migrations config.json	EVM version		
package-lock.json package.json	Istanbul		
README.md truffle-config.js	Optimizer runs		
	Default: 0 (disabled)		
	Linter		
	Solhint		
	Editor Font Family		
	Hack		
🤌 🖉 Ropsten 🗒 Transactions 🖽 AB	Storage	→ Solc (0.8.0)	>_

5.3.3 Linter

Linters analyze code for possible programmatic and styling errors automatically. In "Project Settings", there are Solhint and Ethlint. Developers can choose a familiar lint to complete codes.

🙀 Ethereum Studio		- D >
File Edit View Application Project testBasic		^{work} – 🦿
+ > 🐡 💠	Project Settings × +	
 ▼ testBasic > build ▼ contracts 	EVM version	
Gameltem.sol Gameltems.sol GLDToken.sol	Istanbul Optimizer runs	
 deploys dev.test_20211229_170607.json rinkeby_20211229_185825.json ropsten_20211230_170532.json 	Default: 0 (disabled) Linter	
 migrations config.json package-lock.json 	Solhint	
package.json README.md	Editor Font Family	
truffle-config.js	Hack Font Size	
	13px	
	Font Ligatures	
	Disabled	
🔎 🖉 Ropsten 🗒 Transactions 🖽 ABI	Storage	→ Solc (0.8.0)

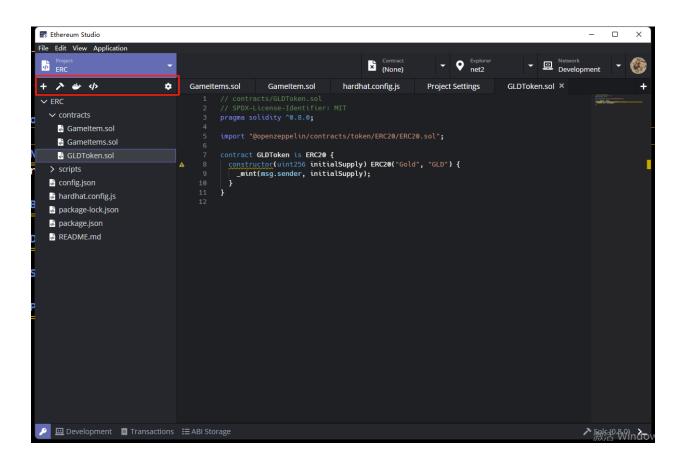
5.3.4 Editor

In "Editor", developers can choose a font-related configuration to make code more specific and direct as desired.

🛃 Ethereum Studio			X
File Edit View Application			
Project testBasic		B Contract 0xca0ffc26 C O Explorer 0xca0	offc26 🝷 🏈 Ropsten 🝷 🍪
+ > 🐡 💠 💠	Project Settings 🗙 🕂		
 testBasic build contracts Gameltem.sol Gameltems.sol GLDToken.sol deploys dev.test_20211229_170607.json rinkeby_20211229_185825.json ropsten_20211230_170532.json migrations 	EVM version Istanbul Optimizer runs Default: 0 (disabled) Linter Solhint		
config.json package-lock.json package.json	Editor		
README.md truffle-config.js	Font Family Hack		
	Font Size 13px		÷
	Font Ligatures		
	Disabled		÷
P P Ropsten	Storago		➢ Solc (0.8.0) >.

5.4 Tool Bar

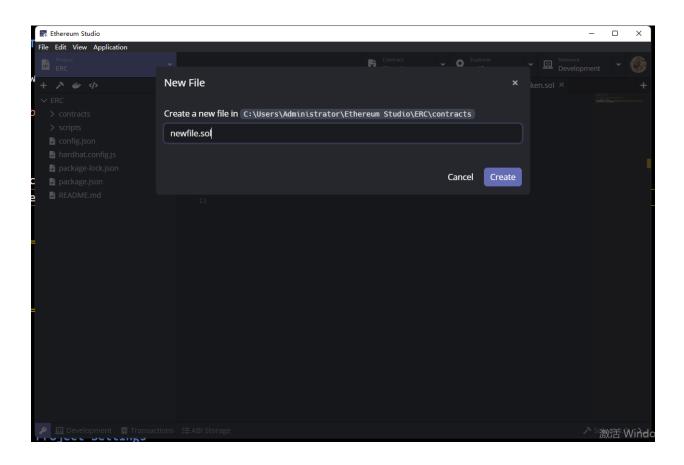
Between "Project" panel and fill tree, tool bar has several quick functions for developers.



5.4.1 New File

Clicking "plus" icon named "New File", developers can create a new file in the current path. Developers can define both name and type of the new file in the input box. Then, click the purple "Create" button and the new file will be generated successfully.

Ethereum Studio Docs



5.4.2 Build

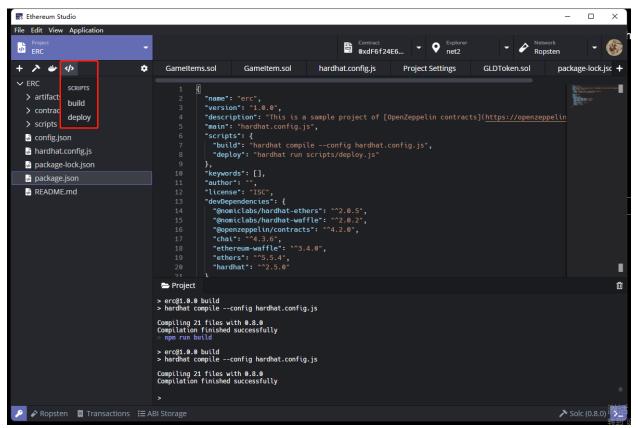
Developers can build all the contracts together quickly by clicking the "hammer" icon. The detail information for building can be checked in "Build" section above.

5.4.3 Deploy

Developers can deploy the contract by clicking the "docker" icon. The detail information for building can be checked in "Deploy" section above.

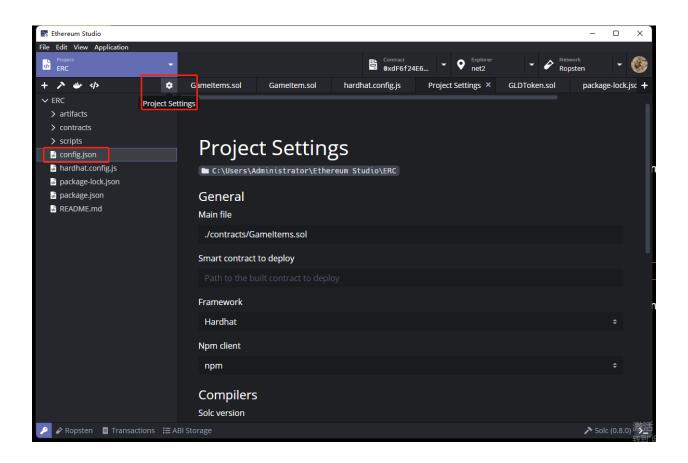
5.4.4 Script

Click the "code" icon and select "build" or "deploy" in "Script". Then there would be corresponding command line in "script" part of "package.json" inputted in the terminal and excuted automatically.



5.4.5 Project Settings

Click the "code" icon and select "build" or "deploy" in "Script". Then there would be corresponding command line in "script" part of "package.json" inputted in the terminal and excuted automatically.



CHAPTER

SIX

NETWORK

6.1 Local Development

Developers may want to run a smart contract on a local network to see how it works before deploying. In Ethereum Studio, developers can create a local blockchain instance to test smart contracts inside the IDE. This local network provides much faster develop iteration than a public testnet(for instance, you don't need to require test ETH from a testnet faucet).

6.1.1 Geth

6.1.2 Geth Version Manager

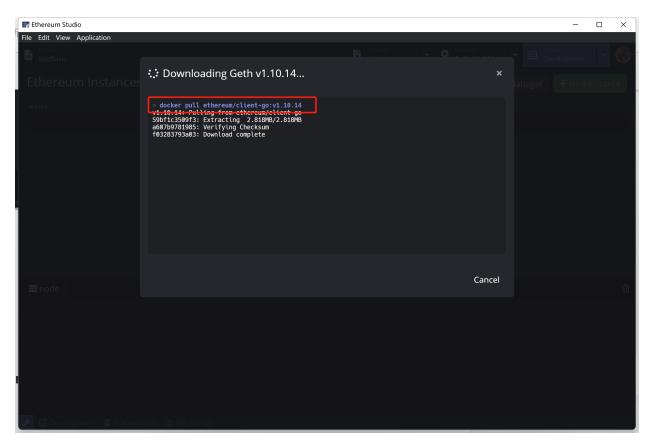
Click "Geth Version Manager" to set a specific version. Before first time launch of Ethereum Studio, developers had installed the Docker image of Geth, so there is a default version.

Ethereum Studio						_	
File Edit View Application							
Project		×	Contract (None)	-	♥ Explorer 0x5b43a901 ▼ 🖳 🕅	^{etwork} vevelopment	- 🛞
Ethereum Instances for	Development				Geth Version Manager	+ New In:	stance
NAME		NODE VERSION	CHAIN		HEIGHT		
		(No Geth instance)				
t r 1							
■ node							Û
🤎 🖳 Development 🗒 Transactions	i ≡ ABI Storage						

Developers can install a Geth version different to the default version.

Ethereum Studio					- 0 ×
File Edit View Application					
Project testBasic			ntract - O Explo		
Ethereum Instances	Geth Version Manag	er		× lanager	
NAME	VERSION	CREATED			
		(No Geth installed)			
	🛃 Install 🔻			Cancel	
· · · · · · · · · · · · · · · · · · ·	AVAILABLE VERSIONS				
	v1.10.14				
	v1.10.14-amd64				
	v1.10.13				
t	v1.10.13-arm64				
r	v1.10.13-amd64				
📰 node	v1.10.12				
	v1.10.12-arm64				
	v1.10.12-amd64				
	v1.10.11				
	v1.10.11-arm64				
	v1.10.11-amd64				
n Development 🗒 Transact	ions 🛛 🗄 ABI Storage				

Geth is installed through Docker image. Developers has to start Docker before installing.



After Geth installed, developers can check Geth version in "Geth Version Manager". There will be a blue number icon beside the "Geth Version Manager" button indicating how many versions are in the manager.

If developers want to remove the installed Geth, double-click the "trash can" icon. After the first click, the "trash can" icon will turn red, and developers can click it again to delete this Geth.

F Ethereum Studio				- 🗆 X
File Edit View Application				
testBasic		Contract	Explorer	
Ethereum Instances	Geth Version Manager		×	
NAME	VERSION	CREATED	sız リ Click again to uninstall	
	v1.10.14	December 23, 2021	50. ^э мв 💼	
	🛓 Install 🗸		Cancel	
ŀ				
📰 node				
c				
v				
n 🖉 🖳 Development 🗒 Transact				
	Horis := ABI Storage			

6.1.3 New Instance

Click "New Instance" and a "New Instance (dev)" window will popup.

Free Ethereum Studio					– 🗆 X
File Edit View Application					
Project testBasic		× (N	ontract Ione)	€ Explorer 0x5b43a901 ▼ □ De	evelopment
Ethereum Instances for	Development			Geth Version Manager 1	+ New Instance
NAME		NODE VERSION	CHAIN	HEIGHT	
		(No Geth instance)			
node					创
🦻 🖳 Development 📲 Transactions	₩ ABI Storage				

Developers can set "Instance name" in the window and change Geth version if there are different versions of Geth. Besides, developers can set "Miner" as the target account. Then click the "Create" button on the bottom right to make an instance.

🔄 🖬 Ethereum Studio					-	о x
File Edit View Application						
Project testBasic		Contract	- O Explorer			- 6
Ethereum Instances	New Instance (dev)			х аg		
NAME	Instance name					
	test			✓		
	Geth version					
	v1.10.14			-		
n an	Miner					
	0x5b43a9013306ea090c9f76ab80730e4d	196a59fa3		-		
9			Cancel Crea	ate		
= node						
L. L						
👂 🖭 Development 📲 Transac						

After a new instance with target Geth version created, the instance will list on the panel. Click "Start" to run the development network. Developers can create an etheruem network and connect to it locally. This local ethereum network provides default 50 ETH for user.

F Ethereum Studio					– 🗆 ×
File Edit View Application			ntract	Explorer	work _
testBasic		× Con	one)	• Explorer 0x5b43a901 • De De	velopment
Ethereum Instances for	⁻ Development			Geth Version Manager 🕦	+ New Instance
NAME		NODE VERSION	CHAIN	HEIGHT	
test	► Start	⊱ v1.10.14	dev		
📑 node					<u>ش</u>

6.1.4 Node Panel

Click the green "Start" button on the Geth instance will start it. With information running on the node panel, the development network is prepared for developers to deploy a contract instantly. This local node cost only local ETH while users already have 50 since node running, so it is easy to test smart contracts on the private network.

ile Edit View Application				- 🗆 X
Project testBasic		one)		etwork evelopment
Ethereum Instances for Developm	nent		Geth Version Manager 1	+ New Instance
NAME	NODE VERSION	CHAIN	HEIGHT	
test Stop	} • v1.10.14	dev	0	
≣ node				ſ
<pre>docker run -itrmname eth-test-v1.10.14 -p 8545:</pre>	=pwd ev mode ETH=50 LES=0 total=50 fisabling err="stat /run/pcscd/pcscd.com	m: no such file		

6.2 Remote

6.2.1 Mainnet

A mainnet is an independent blockchain running its network with its technology and protocol. It is a live blockchain where its cryptocurrencies or tokens are in use, compared to a testnet or projects running on top of other popular networks such as Ethereum.

Ethereum Mainnet is the primary public Ethereum production blockchain, where actual-value transactions occur on the distributed ledger. Ethereum mainnet uses real ETH as currency to transfer assets and pay gas fees and tips. There are many different Ethereum testnets, and each testnet uses its own test ETH as currency, respectively. Developers may deploy and test contracts on at least one testnet before the final release on the mainnet.

6.2.2 Testnets

In addition to Mainnet, there are public testnets. These networks are used by protocol developers or smart contract developers to test both protocol upgrades and potential smart contracts in a production-like environment before deployment to Mainnet.

It's generally essential to test all smart contracts code on a testnet before deploying it to the Mainnet. Suppose developers building a dapp that integrates with existing smart contracts. In that case, most projects have copies deployed to testnets that you can interact with it.

Most testnets use a proof-of-authority consensus mechanism. This means a small number of nodes are chosen to validate transactions and create new blocks – staking their identity in the process. It's hard to incentivize mining on a

proof-of-work testnet which can leave it vulnerable.

Ropsten is a proof-of-work testnet for those running Geth, Besu and all other Ethereum clients. This means it's the best like-for-like representation of Ethereum. Ropsten started in November 2016 and it can be used with all clients.

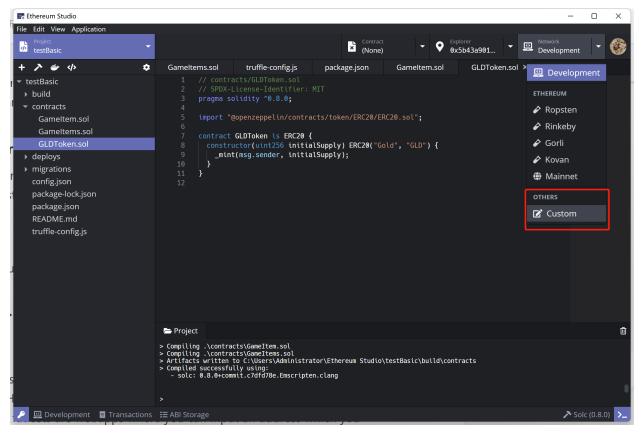
Rinkeby is a proof-of-authority(clique) testnet for those running Geth, Besu, Nethermind, and OpenEthereum client. Rinkeby started in April 2017 and is immune to spam attacks(as Trusted parties control ether supply).

Goerli is a proof-of-authority testnet that works across clients. Goerli started in November 2018. Goerli doesn't fully reproduce the current production environment as it uses PoA.

Kovan is a proof-of-authority testnet for those running OpenEthereum clients. Kovan started in March 2017 and is immune to spam attacks. Kovan doesn't fully reproduce the current production environment as it uses PoA.

6.2.3 Custom Network

In the Ethereum network, there are private networks and public networks. The Ethereum Studio can set "Custom Network" to connect the target network. Connecting to a network, developers can join the network of other nodes instead of establishing a network by oneself. Especially, developers can use a company's network service like Infura **Link**(https: //infura.io/docs).



New Connection

In the "Custom Network" panel, click the "gear" icon, and there will popup a window named "Custom Network".

Ethereum Studio							_		×
File Edit View Application									•
Project d∕ testBasic		× Contract (None)	-	•	Explorer 0x5b43a901	•	Custom	-	Ø
Custom Network		Blocks							
Node URL (https://ropsten.infura.io/v3/11c4d53b53154c9d89041a05e0f5/	jeec2								
🤌 🕼 Custom 📋 Transactions 🔚 ABI Storage									

Click the "New Connection" button, and there will jump out a window for developers to add more network connections. Since each connection represents a node from one of the public or private networks, there will usually be a lot of different connections for developers to connect.

Ethereum Studio					-	οx
File Edit View Application						
testBasic		R.		Explorer Explorer		- 6
Custom Network	Custom Network			×		
	NAME RPC URL					
		(No Custom Networ	ˈks)			
n						
	New Connection			Cancel		
🔎 🗹 Custom l 🇮 Transactions						

In the "New Custom Network Connection" window, developers can input the name and URL of node RPC. In this picture, there is a node of Ropsten network from Infura. Infura is a Web3 backend and Infrastructure-as-a-Service (IaaS) provider that offers blockchain developers a range of services and tools. Developers can use Infura as a fundamental infrastructure of Ethereum projects. Besides, developers can join other Geth nodes with the node parameters.

🔄 🖬 Ethereum Studio	- 0	x u
File Edit View Application		
Project testBasic		- 69
Custom Network	New Custom Network Connection ×	
Y Node URL (https://rop	Name 1177	12190
e	test December 29th, 18:	00-19
t	URL of node rpc	
C	https://ropsten.infura.io/v3/11c4d53b53154c9d89041a05e0f5eec2	
2	Cancel Check Network	
N		
N		
U		
N		
C		

After clicking "Check Network," a "Network info" will be added below the original panel. The panel shows detailed network information to be joined with "URL of node RPC". Developers can check the information and join the network by clicking the "Add Network" button.

Ethereum Studio Docs

F Ethereum Studio	- [o x
File Edit View Application		
testBasic		- (3)
Custom Network	New Custom Network Connection ×	
Node URL https://rop	Name	
N	test	
	URL of node rpc	
U	https://ropsten.infura.io/v3/9c974bfb78fd43779ac5d3bb50c72a02	
	Network info { "name": "ropsten", "chainId": 3, "ensAddress": "0x00000000002E074eC69A0dFb2997BA6C7d2e1e" }	
	Cancel Add Network	
Custom Transactions		

After adding a network, click the green "Connect" button, and there will be a "Blocks" panel showing the "Block Number", "Block Time", and "Difficulty" in real-time.

Ethereum Studio						-	- 0	×
File Edit View Applie	cation							
Project testBasic				Contract	Explorer Explorer			
Custom Net	twork	Custom Netwo	rk		×			
Node URL		NAME	RPC URL					
		test	https://ropsten.infura.io/v	3/11c4d53b53154c9d	Connect 🖋 🗓			
2								
k		New Connection	J		Cancel			
5								
N								
N								
U								
N								
}								
🧨 🗹 Custom 🗒	Transactions	I≡ ABI Storage						
Ethereum Studio						-	- 0	×
File Edit View Applie	cation			Contract	_ Explorer	Network		\$ \$\$.
testBasic				× Contract (None)	• • 0x5b43a901	Custom	▼	Ø
Custom Net	twork		\$	Blocks				
U Node URL	https://rops	ten.infura.io/v3/11c4d53	b53154c9d89041a05e0f5eec2	Block Number			117122	09
N				Block Time		December 2	29th, 18:05:	33
ł				Difficulty			115127445	85
_								
)								
<								

SEVEN

BLOCK EXPLORER

7.1 Account

In the "Account" panel, there are "Balance" and "Nounce". "Balance" represents the ETH amount of the developers. The "Nounce" represents the transaction experience. Specifically, the nonce in the ETH wallet is a scalar value equal to the number of transactions sent from this address or the number of contract creations made by this account. Nonce can be changed manually.

Ethereum Studio								-		×
File Edit View Ap	oplication				Contract	Explorer		Network		
testBasic					0x6b35c451	• 0x5b43a9	01 🔽 🧲	Rinkeby	-	
0x5b43 9fa	3 × +									
`C 💿 0x5b43a	a9013306ea0	90c9f76ab80730e	4d96a59	9fa3					\$	له د
Account				Information						
Balance		18.737 ET	Ð	√ ⊁ Code					(None)	
# Nonce			6							
Transactio	ons									
ТІМЕ	BLOCK	TX HASH		FROM		VALUE	GAS USED		FE	E
12/22 11:24:24	9856576	0x16b7a3fe41	153e07	0x5b43a90133 a59fa3	0x6b35c451ee 1010b6	0 ETH	27,620	27	,620 Gwe	
12/22 10:26:37	9856345	0x6028cb09ae	09e777	0x5b43a90133a59fa3	contract creation 0x6b35c451ee 1010b6	0 ETH	1,222,418		0.003 ETH	
12/20 18:04:12	9846659	0x84b03e5bba	fca711	0x5b43a90133 a59fa3	contract creation 0xf4e94e1b58 d1cc06	0 ETH	1,270,601		0.003 ETH	
12/20 17:02:10	9846411	0xa9e38e12f0	4c3a31	0x5b43a90133 a59fa3	contract creation 0xa0ebb58985 ce7276	0 ETH	1,286,353		0.003 ETH	
2 12/20 16:26:52	9846270	0xefe96e78e4	c5e0e7	0x5b43a90133 a59fa3	contract creation 0x1d0a84f9f0 82e450	0 ETH	1,222,418		0.003 ETH	
12/20 16:12:52	9846214	0x3bc8658739	9f91b8	0x5b43a90133a59fa3	0x5b43a90133a59fa3	2 ETH	21,000	52	,500 Gwe	
12/20 16:12:52	9846214	0x3bc8658739	9f91b8	0x5b43a90133a59fa3	0x5b43a90133 a59fa3	2 ETH	21,000	52	,500 Gwe	
12/20 16:10:07	9846203	nyf9fh779c4e ns ≔ ABI Storage	c75f∆h	®v२1hQRd1₫@@ 1R₫573	Mv5h∆२aQM1२२ a5Qfa२	18 75 ETH	21 000	21	000 Gwe	

7.2 Information

Input one of the contract address in the search box. After pressing down the "Enter" button on the keyboard, the detailed information is shown on the "Information" panel. There is the "Code Hash" of the contract in the search box in the picture.

Ethereum Studio							– 🗆 X
File Edit View Ap	plication						
Project testBasic				Contract 0xb627ec4b	explorer 0x6b35c451	👻 🔶 Networ Rinkel	
0x5b43 9fa	3 0x6b	35 10b6 × 🕂					
C (0x6b35c	:451ee1f65e3	a9c0e375ffe4a9cba6101	0b6				🖈 🗘 🛣
Account			Information				
Balance		0 ETH	 ✓ Code Hash 	0x956e00b4711e6a4	116a6ab55889ac9	98287d4e2bd09b40fb2	4e87ce3ae23fbe7f
# Nonce							
Transactio	ons						
ТІМЕ	BLOCK	тх назн	FROM		VALUE	GAS USED	FEE
2 12/22 11:24:24	9856576	0x16b7a3fe41 153e07	0x5b43a90133 a59fa3	0x6b35c451ee 1010b6	0 ETH	27,620	27,620 Gwei
12/22 10:26:37	9856345	0x6028cb09ae 09e777	0x5b43a90133 a59fa3	contract creation 0x6b35c451ee 1010b6	0 ETH	1,222,418	0.003 ETH
🤌 🔗 Rinkeby 🛛	Transaction:	s ᠄ 🗄 ABI Storage					

7.3 Transactions

In the "Transactions" panel, there is specific information of each transaction on the address, including "Time", "Blockheight", "Transaction Hash", "Owner Address", "Receiver Address", "Value" and so on.

Developers can check the colour of the transaction value to know the input or the output of ETH since the input is green and the output is red. Besides, there is detail information of the receiver address reminding the action status. Developers can click the link of address to show the address in Block Explorer to check the history information.

Ethereum Studio Edit View A							- 0
Project testBasic	ppilcation			Contract 0xb627ec4b	explorer 0x5b43a9		Network Rinkeby
0x5b43 9fa	a3 × 0x0	5b35 10b6 🕂					
• 0x5b43	a9013306ea	090c9f76ab80730e4d96a5	9fa3				☆ ¢
🔁 Balance		18.737 ETH					(None)
# Nonce							
ransacti	ons						
	BLOCK	TX HASH	FROM		VALUE	GAS USED	FEE
2/22 11:24:24	9856576	0x16b7a3fe41 153e07	0x5b43a90133 a59fa3	0x6b35c451ee 1010b6	0 ETH	27,620	27,620 Gwei
2/22 10:26:37	9856345	0x6028cb09ae 09e777	<u>0x5b43a90133 a59fa3</u>	contract creation 0x6b35c451ee 1010b6	0 ETH	1,222,418	0.003 ETH
2/20 18:04:12	9846659	0x84b03e5bba fca711	0x5b43a90133 a59fa3	contract creation 0xf4e94e1b58 d1cc06	0 ETH	1,270,601	0.003 ETH
12/20 17:02:10	9846411	0xa9e38e12f04c3a31	0x5b43a90133 a59fa3	contract creation 0xa0ebb58985 ce7276	0 ETH	1,286,353	0.003 ETH
2/20 16:26:52	9846270	0xefe96e78e4c5e0e7	0x5b43a90133 a59fa3	contract creation 0x1d0a84f9f0 82e450	0 ETH	1,222,418	0.003 ETH
2/20 16:12:52	9846214	0x3bc86587399f91b8	0x5b43a90133a59fa3	0x5b43a90133a59fa3	2 ETH	21,000	52,500 Gwei
2/20 16:12:52	9846214	0x3bc8658739 9f91b8	0x5b43a90133 a59fa3	0x5b43a90133 a59fa3	2 ETH	21,000	52,500 Gwei
2/20 16:10:07	9846203	0xf9fb779c4ec75f4b	0x31b98d1400 184523	0x5b43a90133 a59fa3	18.75 ETH	21,000	21,000 Gwei
🔗 Rinkeby	🗒 Transasti	ons ᠄ 🗄 ABI Storage					

7.4 Transfer

On the right of the address search bar, the "arrow cycle" icon provides transfer function for developers send ETH between address. Click it and a "Transfer" window will popup to let developers to input ETH amount and receiver address for a quick token preparation between address.

Ethereum Studio Docs

Ethereum Studio	b						
le Edit View A	pplication						
Project					Explorer		
		a Transfer				×	
• • 0xe166	85a704eafeb0						☆ ¢
Account		Amount					
Account		0.1					
		Recipient					
		♥ 0xe16685a70	04eafeb05cda60e9912ea	ae67ff5ec568	neti	-	
Transacti	ons					_	
				Cancel	Sign and P	ush SED	
				contract creation 0xdf6f24e60a 1c727b			

7.5 Faucet

There are currencies on the Ethereum network such as ETH and test ETH. Unlike ETH in the mainnet, test ETH has no real value. Therefore, there are no markets for testnet ETH. Most people get testnet ETH from faucets. Faucets are web apps where developers can input an address and the requested test ETH will be sent automatically.

First, choose one of the Ethereum testnets. Then set the account of Block Explorer. There is a "faucet" icon at the right end of the address search bar. Click the icon and it will turn to the faucet page of the corresponding testnet. The Ethereum Studio only supports faucets of Ropsten, Rinkeby and Kovan. Developers can check the Goerli faucet link by oneself.

🔄 🖙 Ethereum Studio										-		×
File Edit View Ap	pplication						_					
Project testBasic						× Contract (None)	· ·	Explorer 0x5b43a901	. 💌 🔗	Network Ropsten	-	
0x5b43 9fa	13 × +											
C 💿 0x5b43	a9013306ea09	0c9f76ab8	0730e4d96a5	9fa3							☆	¢ 🐔
Account				Informa	tion							Faucet
r 🗇 Balance			0 ETH								(No	one)
# Nonce												
Transactio	ons											
тіме	BLOCK	тх назн		FROM				VALUE	GAS USED			FEE
					No Transaction	ns Found						
1												
3												
🧷 🖉 Ropsten	🗒 Transaction		orage									
Kopsterr			onage									

7.5.1 Ropsten Faucet

Click the "Faucet" icon and the page will jump to the Ropsten Faucet. Developers can copy the wallet address and paste it into the box. Then developers clicks the button "Send me test Ether" and wait for a few minutes. There will be several test ETH of Ropensten testnet on balance.

7.5.2 Rinkeby Faucet

Following faucet instruction via inputting specific Twitter or Facebook message links, developers can get test ETH on Rinkeby. Developers can change the amount of test ETH with different lengths of time.

7.5.3 Kovan Faucet

???

EIGHT

CONTRACT

In "Contract" panel, there are three different panels from left to right including "Write Functions", "View Functions" and "Events" respectively. Developers can interact with "Write Functions", mainly calling functions with assets and check the address status with "View Functions". Besides, developers can set parameters and check related events in "Events".

🛃 Ethereum Studio				- 0 X
File Edit View Application				
Project testBasic		Contract 0xca0ffc26		- 🎯
0xca0fdf3a 🗙 🕂				
C • 0xca0ffc26473fcf835b83f71e74	e03c008558df3a			☆
f∞ transfer - 🕨	💿 allowance 🔻 🕨	🛗 Approval 👻 🕨		
WRITE FUNCTIONS	Parameters	Parameters		
approve	owner	Range		
decreaseAllowance	Select or type an address			Clear
increaseAllowance	spender	Event Logs		_
transfer	Select or type an address	BLOCK OWNER	SPENDER VAL	
	Result	address	address uin	t256
transferFrom Estimate Gas Limit	(None)		(no data)	
O Default: 1,000,000				
Tip				
ax priority fee per gas				
Max Fee				
💰 max fee per gas				
Authorization				
Signer				
0x9584bfbe913e75413d4c2 (test)				
Result				
(None)				
🤌 🖉 Ropsten 🗒 Transactions 🖽 ABI Sto	rage			

8.1 Write Functions

8.1.1 Parameters

In "Parameters", developers mannually set function required parameters. Usually, there are owner, receiver, amount and so on for different functions. Please remember the token unit here is "wei".

🚰 Ethereum Studio		-	
ile Edit View Application			
× Project (None)		Contract 0xca0ffc26 Contract 0xca0ffc26	- 🙆
0xca0fdf3a 🗙 🕂			
C • 0xca0ffc26473fcf835b83f71e	74e03c008558df3a		☆
f«) approve 🔻 🕨	👁 allowance 👻 🕨	🛗 Approval 👻 🕨	
Parameters	Parameters	Parameters	
spender	owner	Range	
Select or type an address	Select or type an address		Clear
amount	spender	Event Logs	
123 uint256	• Select or type an address	BLOCK OWNER SPENDER VALUE address address uint256	
Gas Estima	ate Result	(no data)	
Gas Limit Default: 1,000,000 Tip Time max priority fee per gas Max Fee Time Time Time Time Time Time Time Ti	(None)		
Authorization Signer P 0x9584bfbe913e75413d4c2 🚥			
Result (None)			
🔎 🔗 Ropsten l 🏾 Transactions 😑 ABI !			

8.1.2 Gas

If developers want to call functions, there would be some gas fee and tip required by the miner of blockchain nodes. Especially when the network is congested, the gas fee can be very large, so developers should set a "Max Fee" in case of an unexpectedly high cost. Developers can click the purple "Estimate" button to get the real-time gas price of deploying contract on network. If developers supposes the price is not fair, they can click again or wait for a period and the Ethereum Studio will show changed price.

File Edit View Application View Applicat	lio					_	
0xca0f df3a x + C • 0xca0ffc26473fcf835b83f71e74e0c08558df3a fw) approve • > > Parameters Parameters spender • Select or type an address amount 123 uint256 Gas Estimate Select or type an address Result (None) WALUE Max Fee max fee per gas							
fvi approve < > Image: Compare the second sec			Contract 0xca0ffc26	Explore test	er 🗸 🏈		- 🍪
fm approve ▼ ▶ Image: Control of the second of the se	f3a 🗙 🕂						
Parameters Parameters spender owner Select or type an address Select or type an address amount spender 123 uint256 Select or type an address Gas Estimate None None	ffc26473fcf835b83f71e74e03c0	008558df3a					☆
spender owner Range o Select or type an address Select or type an address Iatest - 999 - Iatest amount spender Select or type an address Event Logs Gas Estimate None MALPEG Ino data) VALUES Max Free max fee per gas Ino data) Ino data) VALUE		> allowance 🔻 🕨	🛗 Approval 🔻				
Select or type an address Select or type an address amount spender 123 uint256 Select or type an address Gas Estimate Gas Limit OWNER O Default: 1,000,000 None) Tip (None) Tip (None) Tip (None)	Pa	arameters	Parameters				
amount 123 uint256 Select or type an address Gas Estimate Gas Limit Default: 1,000,000 Tip max priority fee per gas max fee per gas	d	owner	Range				
123 uint256 Select or type an address BLock OWNERs of ddress SPENDER uint256 VALUE uint256 Gas Estimate Result (no data) (no data) Integration of the person of the p		• Select or type an address					Clear
123 uint256 Select or type an address BLOCK OWNER address SPENDER uint256 Gas Estimate Result (no data) (no data) Jo Default: 1,000,000 (None) (None) (None) (None) Max Fee Imax fee per gas	s	spender	Event Logs				
Gas Limit (None) Figure 2 (Sock address address uint256 (no data) (no data		• Select or type an address		OWNER	SPENDER	VALUE	
Gas Limit (None) O Default: 1,000,000 Imax Priority fee per gas Imax Priority fee per gas Imax fee per gas	Estimate D	osult	BLOCK	address	address	uint256	
Authorization Signer	1,000,000 prity fee per gas per gas	None)					

8.1.3 Authorization

In "Authorization", developers can choose "Signer" to pay gas fee and tip.

F Ethereum Studio				- 🗆 X
File Edit View Application				
× Project (None)		Contract 0xca0ffc26	♥ ^{Explorer} test ▼	Ropsten - With
0xca0fdf3a 🗙 🕂				
C • 0xca0ffc26473fcf835b83f71e74e	03c008558df3a			
f⇔ transfer - ►	💿 allowance 🔻 🕨	🛗 Approval 👻 🕨		
Parameters	Parameters	Parameters		
recipient	owner	Range		
♀ 0x9584bfbe913e75413d4c2fa test	• Select or type an address			Clear
amount	spender	Event Logs		
123 123	Select or type an address	вьоск оwne addre		VALUE uint256
Gas Estimate	Result		(no data)	
Gas Limit 30420 Tip	(None)			
250000000				
Max Fee				
ā 6192511662				
Authorization				
Signer				
0x9584bfbe913e75413d4c2 test				
Result				
(None)				
🦻 🖉 Ropsten 🗒 Transactions ᠄ 🖃 ABI Stora	age			

8.1.4 Results

In "Results", there are some direct results of the return value of functions. Developers check error or successful messages through results.

F Ethereum Studio					-	
File Edit View Application						
(None)		Contract 0xca0ffc26	- • t	est T	Ropsten	- 🎯
0xca0fdf3a 🗙 🕂						
C (0xca0ffc26473fcf835b83f71e74	e03c008558df3a					
f⇔ transfer 👻 🕨	④ balanceOf ▼ ▶	🖮 Approval 👻				
recipient	Parameters	Parameters				
• 0x9584bfbe913e75413d4c2 test	account	Range				
amount	• 0x9584bfbe913e75413d4c2fatest					Clear
123 123	Result 🛇 Pretty Raw	Event Logs				
Gas Estimate	(1 return	BLOCK	OWNER	SPENDER	VALUE	
Gas Limit	3000000000 uint256	block			uint256	
o 30420	,			(no data)		
Тір						
250000000						
Max Fee						
<u>دة</u> 6192511662						
Authorization						
Signer						
⊘ 0x9584bfbe913e75413d4 test						
Result 🔗 Pretty Raw						
[
true]						
- P P Ropsten III Transactions 注 ABI Sto						
ABI SIC	n age					

8.2 View Functions

The middle pannel represents "View Functions" in deployed contracts. View functions ensure that they will not modify the state. Most of time, developers can quickly check variable state by calling view functions. Yet view functions still could receive variables and return newly created data structure and variables. So this pannel could help developers check if they get wanted middle results.

8.2.1 Parameters

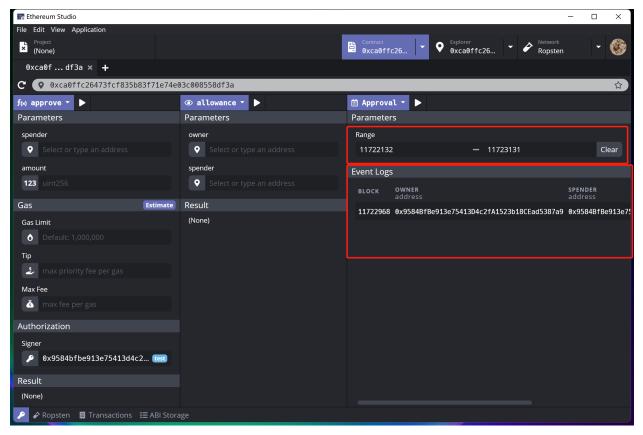
Some of view functions receive variables so they have "Parameters". In "Parameters", developers could switch variable types between "hex" bytes or "utf8" uint as functions required. By excuting function, developers check and get desire results without changing variables in contracts.

8.2.2 Results

There are two types of results, "Pretty" and "Raw". In "Raw", there are original return values of excuted view function. Switching to "Pretty", the return values has more detail information such as number and types of returnd value.

8.3 Events

"Events" locates on the right pannel, where developers can choose and check detail information of events on the network. The types of events are corresponding to events in contract. For example, in the deployed ERC20 contract, there are "Approval" and "Transfer" events. Click the "Execute" triangle beside the "Approval" button, developers can check the latest event with its block number in the settled range.



8.3.1 Parameters

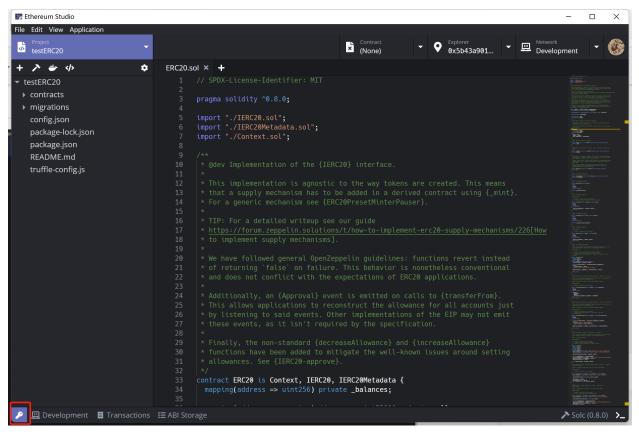
In "Parameter", developers set numbers to quickly check the range of events in a list. If developers do not set numbers, the data range is the latest 100 by default.

8.3.2 Events Log

Detail information of required events will show on the "Events Log" with settled parameters and excuted events.

KEYPAIR MANAGER

Click the purple "key" icon in the bottom left of panel and developers can see the "Keypair Manager" window popup.

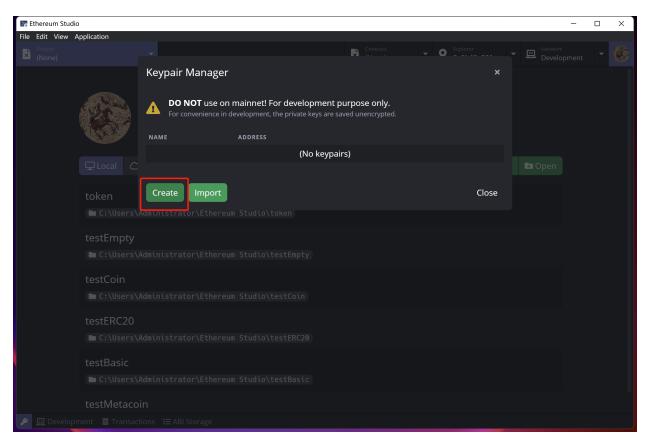


In "Keypair Manager" panel, there is a reminder showing that keypairs kept in the manager should not be used on mainnet. For the convenience of development, the keypairs are all unencrypted, private keys will easily be checked by anyone using a desktop client under the account. Since keypairs used on mainnet containing real ETH assets and private keys, it is dangerous to lose ETH and assets if developers keep Ethereum mainnet keypairs in the keypair manager.

Ethereum Studio			- D >
e Edit View Application			
		Contract Contract Deve	
· > =	Keypair M	anager ×	
	51		
		DT use on mainnet! For development purpose only. venience in development, the private keys are saved unencrypted.	
	- For conv	/enience in development, the private keys are saved unencrypted.	
		ADDRESS	
	NAME	ADDRESS	
		(No keypairs)	
	Create	Import	
	Create	t).	

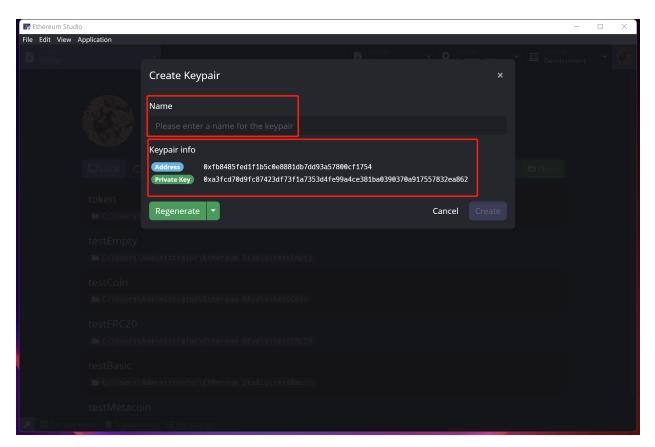
9.1 Create Keypair

Click the "Create" button on the bottom left corner. Developers can easily generate a random keypair for later deployment.



In "Create Keypair" window, there will be a newly generated address and private key in "Keypair info". The newly created key pair has no name, and developers can input a desired name for it as long as the new name is not the same as any other keypairs name. Otherwise, there would be an error message reminding the name has been used. Names can be easily changed in "Keypair Manager" later.

The keypair will show its name rather than address string in the Block Explorer panel since names may contain a more meaningful message for developers to recognize.



In "Regenerate", an inverted triangle shows that developers can either regenerate a private key or regenerate a mnemonic. In the blockchain, mnemonic means a fixed-length number of words that mnemonic can generate to one and only one private key for its user. Compared with private keys including random permutations of characters and numbers, the mnemonic can be easily remembered by a user who wants to keep critical information in a meaningful way.

If developers wants another keypair, they can click "Regenerate" and have a different private key or mnemonic with a corresponding address. After generating keypair and name, developers click "Create" in the bottom right corner of the panel and save this keypair in "Keypair Manager".

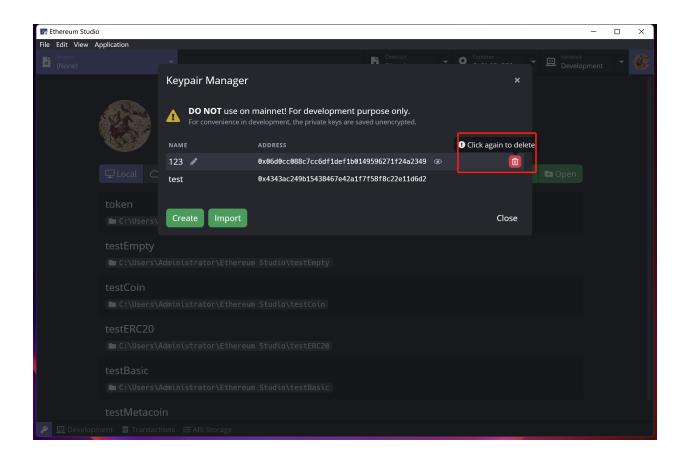
F Ethereum Studio			- 0	×
File Edit View Application				
	Create Keypair	×		
	Name			
	test			
	Keypair info			
	Address 0x894a5cb544d52c743f7be6eae952d28d962071a8 Private Key 0x2fa07b55ed4484728e1b7bd8fd75352812670f860dcb14ab4c7194548e3cef22			
	Aunare web 045140407360440415067010001012222015010100000007014004011243496526155			
	Regenerate Cancel Create	e		
	Regenerate from private key			
	VAdmintstrator\t			
testBasic				
	ctions 🗄 🗄 ABI Storage			

9.2 Check Keypair

In "Keypair Manager", there is a list of keypairs showing names and adresses. Developers can quickly select whole address by double clicking the address. If developers want to check private key of any keypair, they can move mouse upon the address and there will be a "eye" icon showing on the right of address. Double click the icon and there will be a window named "View Private Key" showing "Address" and "Private Key" of selected keypair.

9.3 Delete Keypair

After creating several keypairs, developers can delete the unwanted keypair by double-clicking the "trash can" icon at the right end of the address of keypair. Please remember that the keypair deleted can not be restored unless one has kept the keypair information before and import the keypair again mannually.



9.4 Import Keypair

Except for creating or recreating a random key pair, developers can import an existed keypair by clicking the green "Import" button on the bottom left. There will be an "Import Keypair" window after clicking. Developers can copy the existed keypair information and paste it into the box. The corresponding address is automatically generated after inputting the private key or mnemonic. A green "checkmark" will be at the end of the box, indicating a correct input under rules. Then, click the "Import" button on the bottom right corner to import the keypair.

Ethereum Studio		- 0	×
File Edit View Applicat	ion		
	Import Keypair ×		
	Name		
	test		
	Enter the private key / mnemonic you want to import		
	C limb focus card aunt point arrow ugly clever flee salt again game		
	Address: 0x4343ac249b15438467e42a1f7f58f8c22e11d6d2		
	Cancel Import		
	tEmpty		
tes			
	🗐 Transactions 🛛 🗄 ABI Storage		

9.4.1 Import MetaMask Mnemonic

In MetaMask, developers uses a mnemonic as a private key to enter the wallet. Developers can import the mnemonic of MetaMask as a login MetaMask wallet from a newly installed browser extension of MetaMask. Developers can check detail information in link.

TEN

RPC CLIENT

ELEVEN

SUPPORTED FAUCETS

There are currencies on the Ethereum network such as ETH and test ETH. Unlike ETH in the mainnet, test ETH has no real value. Therefore, there are no markets for testnet ETH. Most people get testnet ETH from faucets. Faucets are web apps where developers can input an address and the requested test ETH will be sent automatically.

First, choose one of the Ethereum testnets. Then set the account of Block Explorer. There is a "faucet" icon at the right end of the address search bar. Click the icon and it will turn to the faucet page of the corresponding testnet. The Ethereum Studio only supports faucets of Ropsten, Rinkeby and Kovan. Developers can check the Goerli faucet link by oneself.

11.1 Ropsten Faucet

Click the "Faucet" icon and the page will jump to the Ropsten Faucet. Developers can copy the wallet address and paste it into the box. Then developers clicks the button "Send me test Ether" and wait for a few minutes. There will be several test ETH of Ropensten testnet on balance.

- Ropsten Faucet 1
- Ropsten Faucet 2
- Ropsten Faucet 3

11.2 Rinkeby Faucet

Following faucet instruction via inputting specific Twitter or Facebook message links, developers can get test ETH on Rinkeby. Developers can change the amount of test ETH with different lengths of time.

- Rinkeby Faucet 1
- Rinkeby Faucet 2

11.3 Goerli Faucet

Following faucet instruction via inputting specific Twitter or Facebook message links, developers can get test ETH on Rinkeby. Developers can change the amount of test ETH with different lengths of time. This procedure is similar to Rinkeby Faucet.

- Goerli Faucet 1
- Goerli Faucet 2

11.4 Kovan Faucet

Click the link of "Kovan Faucet 1" and request test ETH as instruction.

- Kovan Faucet 1
- Kovan Faucet 2

TWELVE

SUPPORTED FRAMEWORKS

12.1 Open Zeppelin

Open Zeppelin provides a complete suite of security products to build, manage, and inspect all aspects of software development and operations for Ethereum projects.

In the Ethereum Studio, developers can import Open Zeppelin contracts through the template. Detail information about Open Zeppelin can be checked in link.

12.2 Hardhat

Hardhat is a development environment to compile, deploy, test, and debug Ethereum software. It helps developers manage and automate the recurring tasks inherent to the process of building smart contracts and dApps and quickly introduces more functionality around this workflow. This framework means compiling, running and testing smart contracts at the very core.

In the Ethereum Studio, developers can use Hardhat as a development framework. Developers can choose it during the creation of projects. Developers can also use this framework during the development process by the command line in the Ethereum Studio. Detail information about Hardhat can be checked in link.

12.3 Truffle

Truffle is a development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM). With Truffle, users get a built-in smart contract compilation, linking, deployment and binary management. Truffle also has network management for deploying to any number of public & private networks.

In the Ethereum Studio, developers can use Hardhat as a development framework. Developers can choose it during the creation of projects. Developers can also use this framework during the development process by the command line in the Ethereum Studio. Detailed information about Hardhat can be checked in link.

Dockerized Truffle used only Truffle in Docker image without other dependencies like Node.js and Npm in local. The other two Frameworks, Hardhat and Waffle, require Node.js and Npm.

12.4 Waffle

Waffle is a library for writing and testing smart contracts. Sweeter, simpler and faster than Truffle. In Waffle, "Simpler" means minimalistic, few dependencies; "Sweeter" means nice syntax, easy to extend; "Faster" means to focus on the speed of tests execution. Waffle uses a set of chai matches, and it can import contracts from npm modules easily. There is also a fast compilation with native and dockerized solidity contracts. It provides fixtures that help write fast and maintainable test suites.

In the Ethereum Studio, developers can use Waffle as a development framework. Developers choose Waffle during the creation of projects. Developers can also use this framework during the development process by the command line in the Ethereum Studio. Detailed information about Waffle can be checked in link.

THIRTEEN

SUPPORTED TESTNETS

Ethereum Mainnet is the primary public Ethereum production blockchain, where actual-value transactions occur on the distributed ledger. There are many different Ethereum Testnets, and each testnet uses its own test ETH as currency, respectively. It's generally essential to test all smart contracts code on a testnet before deploying it to the Mainnet.

Most testnets use a proof-of-authority consensus mechanism. This means a small number of nodes are chosen to validate transactions and create new blocks – staking their identity in the process. It's hard to incentivize mining on a proof-of-work testnet which can leave it vulnerable.

13.1 Ropsten

Ropsten is a proof-of-work testnet for those running Geth, Besu and all other Ethereum clients. This means it's the best like-for-like representation of Ethereum. Ropsten started in November 2016 and it can be used with all clients. Detail information can be checked in link.

13.2 Rinkeby

Rinkeby is a proof-of-authority(clique) testnet for those running Geth, Besu, Nethermind, and OpenEthereum client. Rinkeby started in April 2017 and is immune to spam attacks(as Trusted parties control ether supply). Detail information can be checked in link.

13.3 Goerli

Goerli is a proof-of-authority testnet that works across clients. Goerli started in November 2018. Goerli doesn't fully reproduce the current production environment as it uses PoA. Detail information can be checked in link.

13.4 Kovan

Kovan is a proof-of-authority testnet for those running OpenEthereum clients. Kovan started in March 2017 and is immune to spam attacks. Kovan doesn't fully reproduce the current production environment as it uses PoA. Detail information can be checked in link.

FOURTEEN

TRANSACTION HISTORY

14.1 Transaction Detail

After calling functions, developers can check the transaction status in the "Transaction" button at the bottom of IDE. Click the function name, and there will be a popup "Call a Contract" window.

🛃 Ethereum Studio				o x
File Edit View Application				
× Project (None)		Contract Contract 0xca0ffc26 Explore test		- 🛞
0xca0fdf3a 🗙 🕂				
C • 0xca0ffc26473fcf835b83f71e	74e03c008558df3a			
f⇔ transfer - 🕨	● balanceOf ▼ ▶	🛗 Transfer 🔻 🕨		
recipient	Parameters	Parameters		
• 0x9584bfbe913e75413d4c2 test	account	Range		
amount	• 0x9584bfbe913e75413d4c2fa test	0	— 3000	Clear
123 123				
	Result Pretty Raw	Event Logs		
Gas	C (1 return 3000000000 uint256	BLOCK FROM address	TO VALUE address uint256	
Gas Limit)		(no data)	
ð 30420				
Тір				
الح 250000000				
Max Fee				
<u>دة</u> 6192511662				
Authorization				
Signer 0xca0ffc26473fcf835b8}f				
Øx9584b [™] ⊘ Deploy GLDToken				
Result 🧭 😔 transfer				
[0x519dbfb865b34feece1)42				
true © Deploy] GLDToken				
🥟 🖉 Ropsten 🗒 Transactions 🖽 🖂	Storage			

There is detailed information about the transaction in the "Call a Contract" window. In this window, the contract address means where the function is called. There is other detailed information for developers to check in different panels.

Ethereum Studio Docs

🛃 Ethereum Studio						-	
File Edit View Application							
None)				Explorer			
0xca0fdf3a × 🕂	Call a Contract				×		
C 💿 0xca0ffc26473fcf835	_						
f⇔ transfer - ►	Basic Parameters Tx	Receipt Result					
recipient 0x9584bfbe913e75413d4	# Hash	0x26cae6b4dc963b8dcb641bff6	f0ef3f509a99d005e1	L5a2f8b38c7a3e2a3f2ffb			
amount 123 123	O Status			CONFIRMED			
	🖺 Contract	0xc	a0ffc26473fcf835b8	3f71e74e03c008558df3a	•		
Gas	f⊛ Function			transfer	R S		
Gas Limit	JW Function			transier			
Tip	ETH Sent			0.0 ETH	D		
250000000	🔎 Signer	0x9	584bfbe913e75413d4	lc2fa1523b18cead5387a9			
Max Fee							
ā 6192511662				Clos	e		
Authorization							
Signer							
<pre>0x9584bfbe913e75413d4.</pre>							
Result 📀 🛛 🛛 Pre							
[true							
]							
🔎 🖉 Ropsten 🛛 🗮 Transactions	i⊟ ABI Storage						

FIFTEEN

TRUFFLE MIGRATION SCRIPT

SIXTEEN

ABI STORAGE