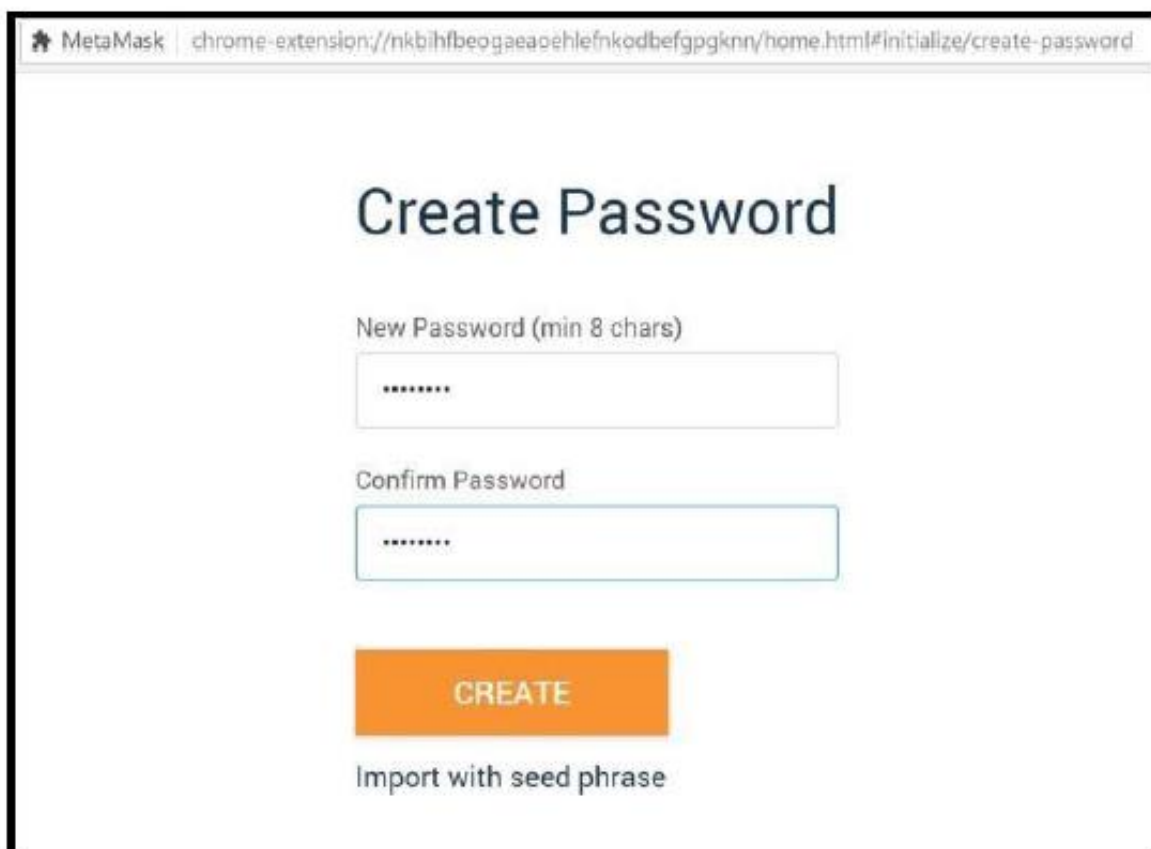
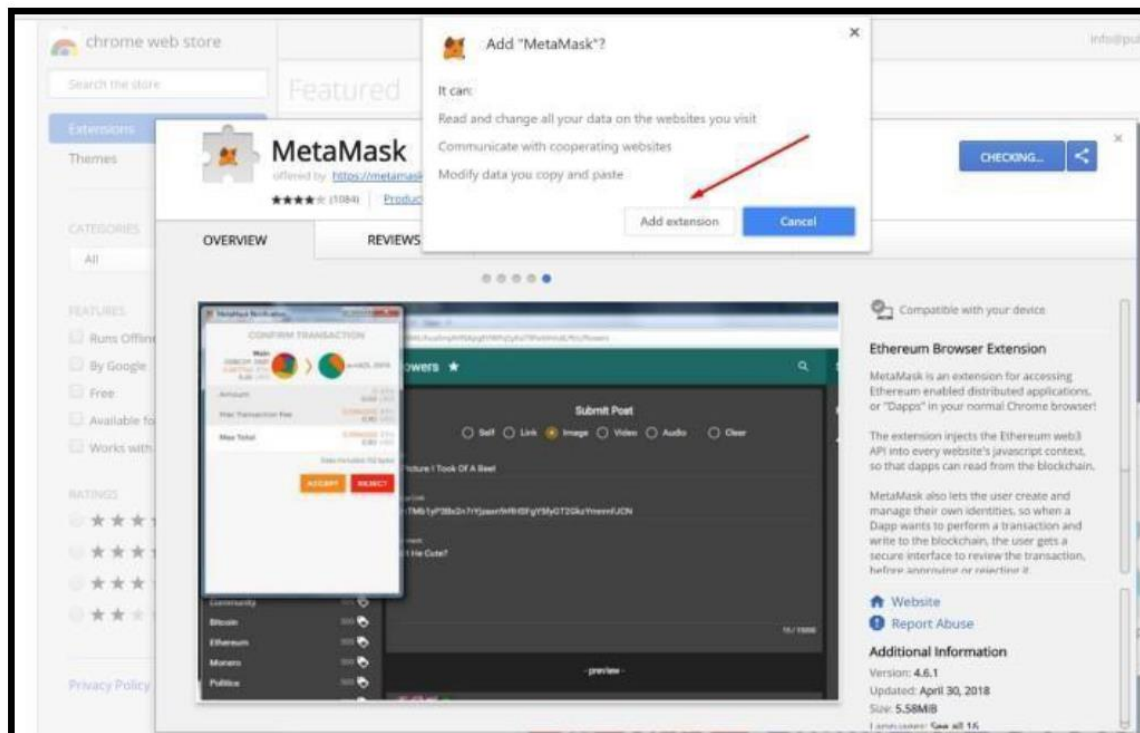
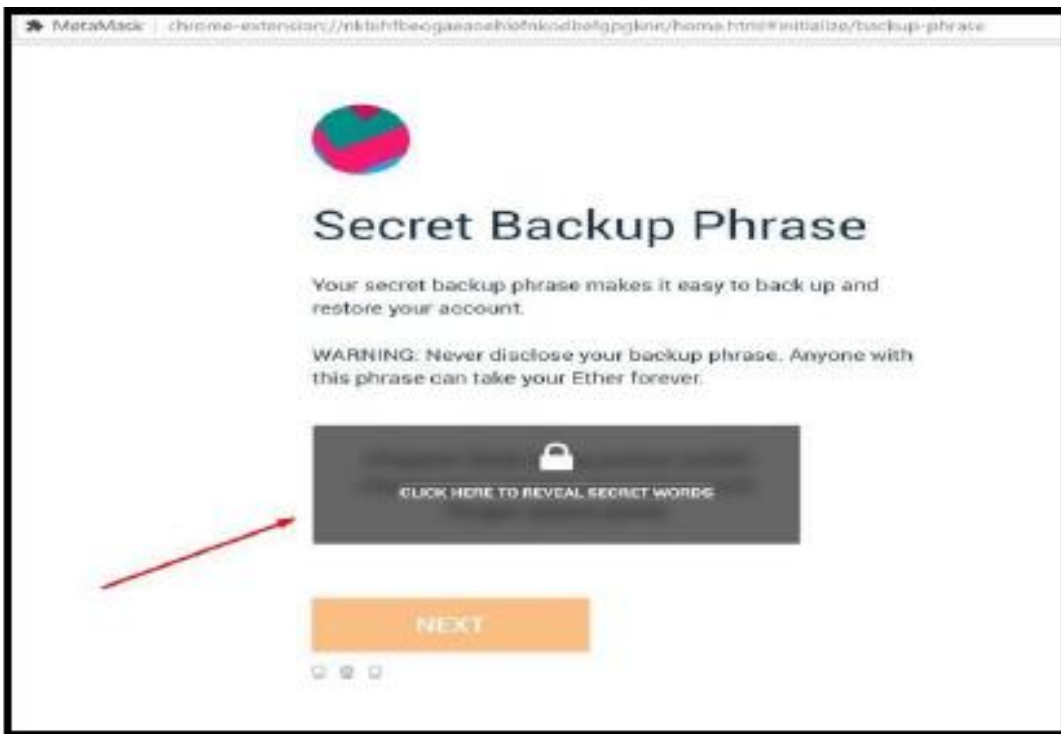
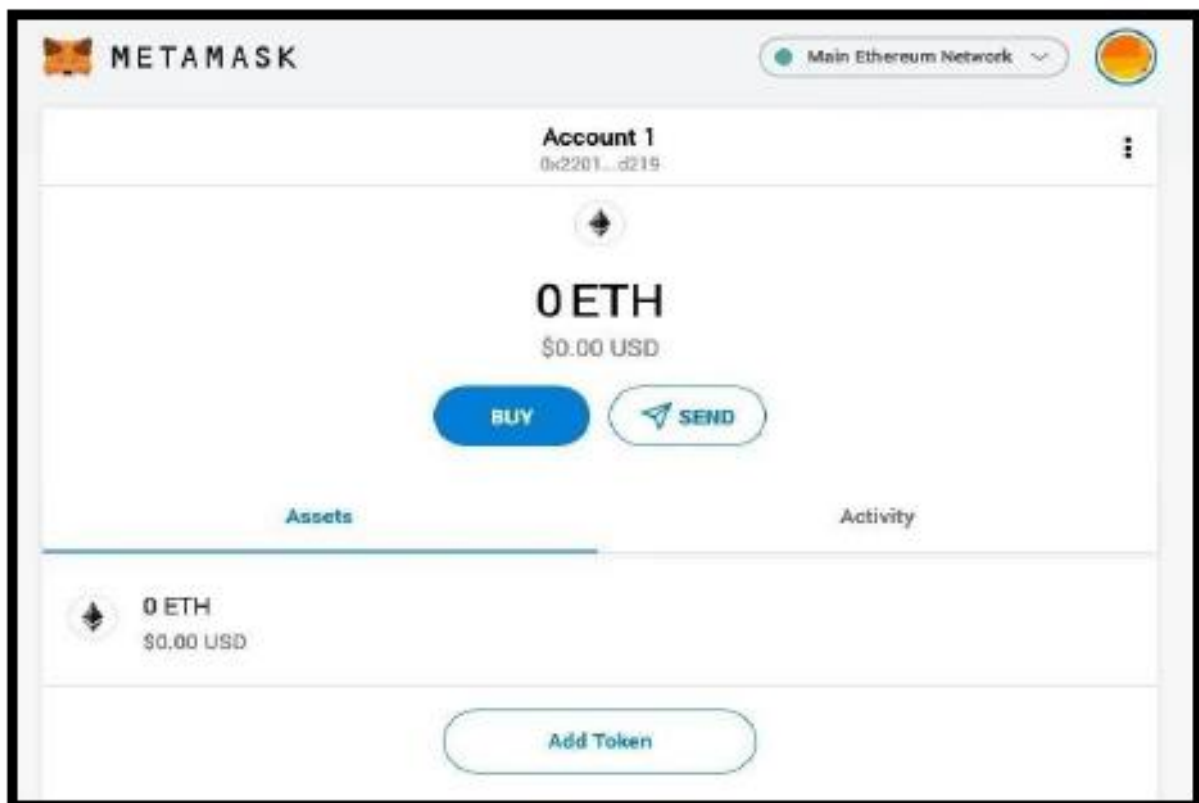
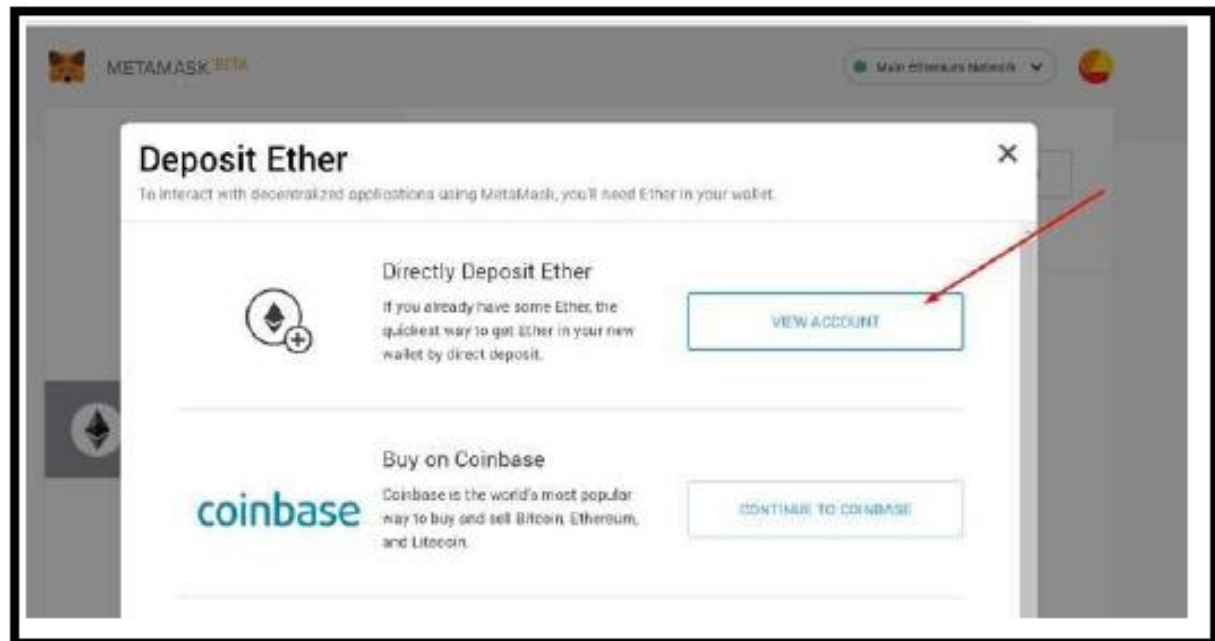



BC1













Main Ethereum Network




Send ETHCancel

New address detected! Click here to add to your address book.

Asset:

 **ETH**
Balance: 0.06889 ETH

Amount:

0.060448 ETH
\$23.45 USD

Max

↑↓

Transaction Fee:

Slow	Average	Fast
0.00447 ETH \$1.74	0.00503 ETH \$2.53	0.00844 ETH \$3.28

Advanced Options

Cancel

Next

Let's get started

Trusted by millions, MetaMask is a secure wallet making the world of web3 accessible to all.



☒ I agree to MetaMask's [Terms of use](#)

Create a new wallet

Import an existing wallet



Wallet creation successful

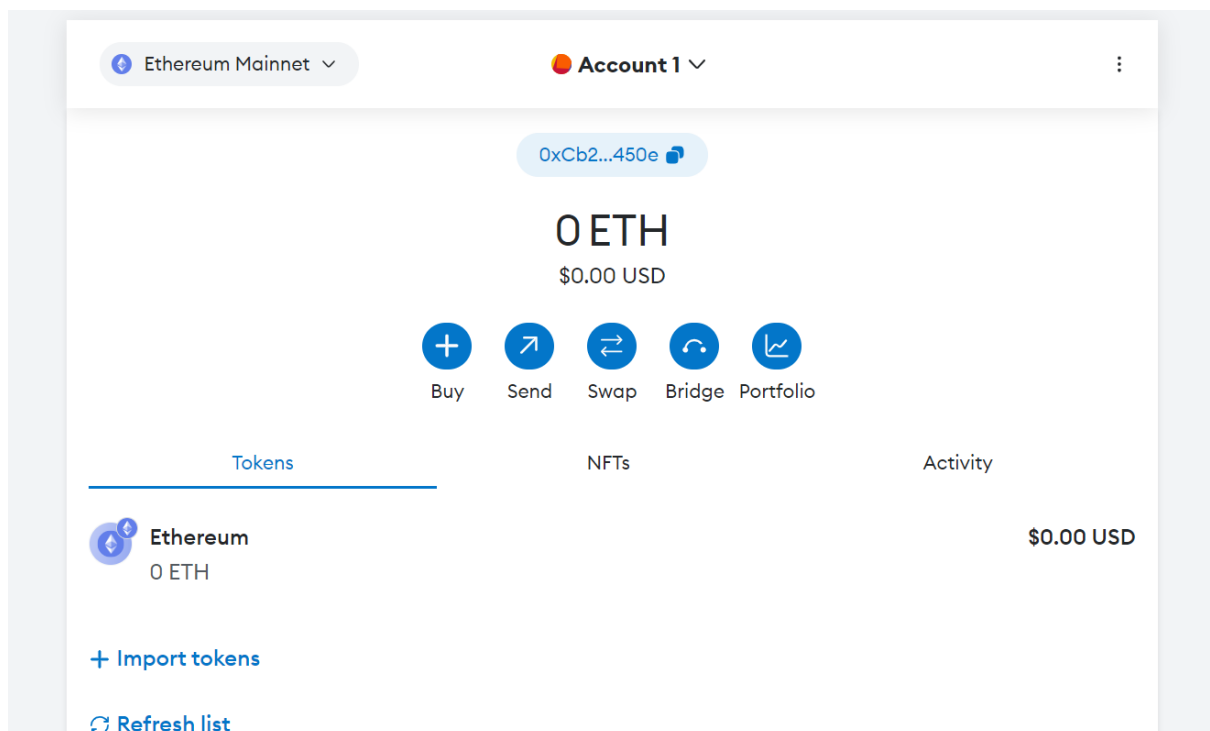
You've successfully protected your wallet. Keep your Secret Recovery Phrase safe and secret -- it's your responsibility!

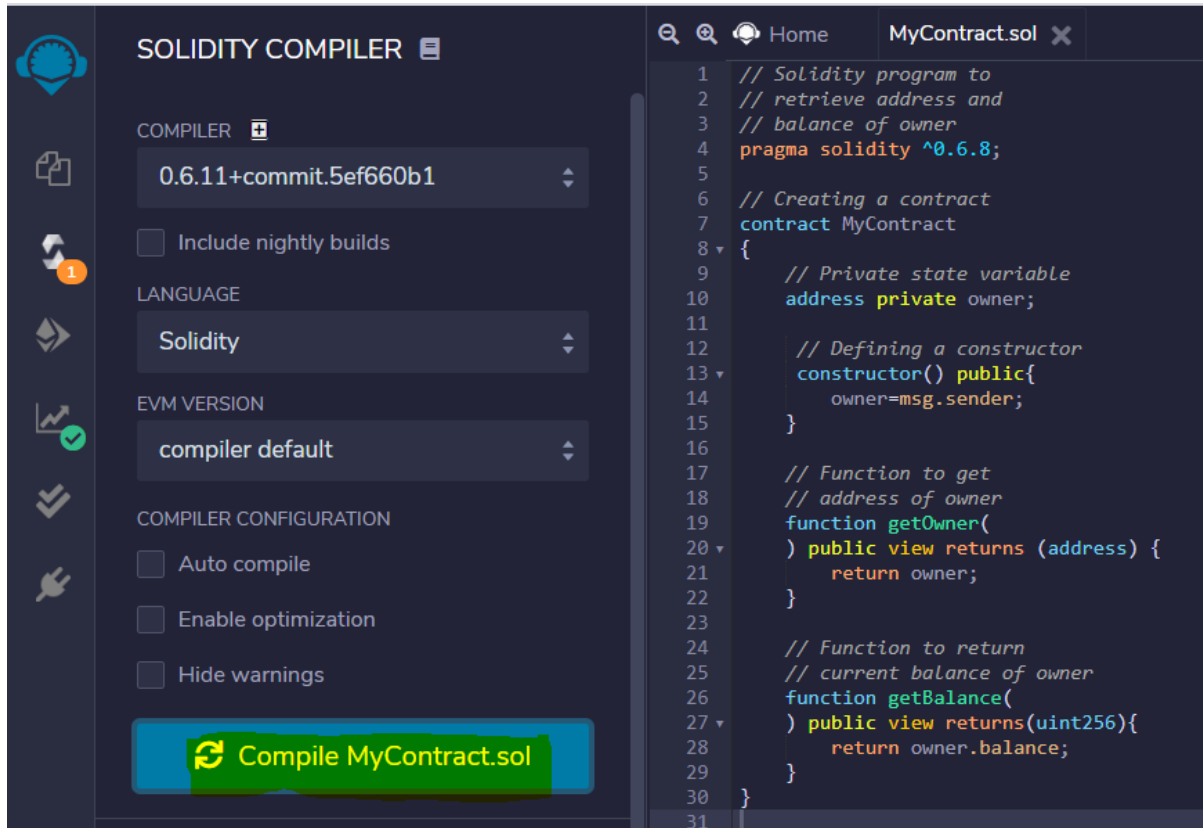
Remember:

- MetaMask can't recover your Secret Recovery Phrase.
- MetaMask will never ask you for your Secret Recovery Phrase.
- **Never share your Secret Recovery Phrase** with anyone or risk your funds being stolen
- [Learn more](#)

[Advanced configuration](#)

Got it!












The image shows the Solidity Compiler web interface. On the left, there is a sidebar with icons for a compiler, a file, a refresh button, a graph, a checkmark, and a plug. The main panel is titled "SOLIDITY COMPILER" and contains the following sections:

- COMPILER**: A dropdown menu showing "0.6.11+commit.5ef660b1".
- ☐ Include nightly builds
- LANGUAGE**: A dropdown menu showing "Solidity".
- EVM VERSION**: A dropdown menu showing "compiler default".
- COMPILER CONFIGURATION**: Three checkboxes: "Auto compile", "Enable optimization", and "Hide warnings".
- A large green button with a refresh icon and the text "Compile MyContract.sol".

On the right, there is a code editor with a tab labeled "MyContract.sol". The code is as follows:

```
1 // Solidity program to
2 // retrieve address and
3 // balance of owner
4 pragma solidity ^0.6.8;
5
6 // Creating a contract
7 contract MyContract
8 {
9     // Private state variable
10    address private owner;
11
12    // Defining a constructor
13    constructor() public{
14        owner=msg.sender;
15    }
16
17    // Function to get
18    // address of owner
19    function getOwner(
20    ) public view returns (address) {
21        return owner;
22    }
23
24    // Function to return
25    // current balance of owner
26    function getBalance(
27    ) public view returns(uint256){
28        return owner.balance;
29    }
30 }
31
```



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

JavaScript VM

ACCOUNT

0xFc9...5fD85 (99.9999999)

GAS LIMIT

3000000

VALUE

0 wei

CONTRACT

MyContract - browser/MyContract.s

Deploy

☐ PUBLISH TO IPFS

OR

At Address Load contract from Address

Transactions recorded 1

Home MyContract.sol

```
1 // Solidity program to
2 // retrieve address and
3 // balance of owner
4 pragma solidity ^0.6.8;
5
6 // Creating a contract
7 contract MyContract
8 {
9     // Private state variable
10    address private owner;
11
12    // Defining a constructor
13    constructor() public{
14        owner=msg.sender;
15    }
16
17    // Function to get
18    // address of owner
19    function getOwner(
20    ) public view returns (address) {
21        return owner;
22    }
23
24    // Function to return
25    // current balance of owner
26    function getBalance(
27    ) public view returns(uint256){
28        return owner.balance;
29    }
30 }
31
```


Deployed Contracts



▼ MYCONTRACT AT 0X426...7A748 (MEMORY)



getBalance

0: uint256: 99999999999999996834857

getOwner

0: address: 0xFc966D2d07Ef456E9edAadA
41dCe4AE12225fD85

Survey Report on Types of Blockchains and Real-Time Use Cases

Date: October 12, 2023

Introduction

Blockchain technology has revolutionized various industries by offering a decentralized and secure way to record and verify transactions. This survey report explores the different types of blockchains and provides real-time use cases to demonstrate their applications in today's world.

Types of Blockchains

Blockchains can be categorized into three main types:

1. Public Blockchains These are open and permissionless networks accessible to anyone. They are often used for cryptocurrencies like Bitcoin. Real-time use cases include:

- Cryptocurrencies: Bitcoin, Ethereum, and others enable peer-to-peer financial transactions and decentralized applications (DApps).
- Supply Chain Management: Public blockchains like VeChain are used to track and verify the authenticity and origin of products.

2. Private Blockchains: These are permissioned networks where access is restricted to selected participants. Real-time use cases include:

- Financial Services: JP Morgan's Quorum blockchain is used for interbank transactions and improving settlement times.
- Healthcare: Patient records and clinical trials data can be securely managed and shared through private blockchains.

3. Consortium Blockchains: A hybrid of public and private blockchains, where a group of organizations jointly manages the network. Real-time use cases include:

- Trade Finance: We.Trade is a consortium blockchain used for transparent and efficient trade finance processes.
- Real Estate: Consortia like Propy streamline property transactions, reducing fraud and paperwork.

Real-Time Use Cases

Blockchain technology has found its way into numerous industries, offering innovative solutions to age-old problems:

1. Finance

- Cross-Border Payments: Ripple's XRP and Stellar facilitate real-time cross-border payments, reducing fees and settlement times.
- Tokenization of Assets: Security tokens on blockchain enable fractional ownership of assets like real estate, art, and commodities.

2. Supply Chain

- Food Safety: IBM's Food Trust uses blockchain to trace the origin of food products, reducing the impact of recalls and ensuring food safety.
- Counterfeit Prevention: Luxury brands like LVMH use blockchain to authenticate luxury products, protecting consumers from counterfeit goods.

3. Healthcare

- Patient Records: MedRec employs blockchain for secure and interoperable patient health records, improving data access for healthcare providers.
- Drug Traceability: Blockchain ensures the authenticity of pharmaceuticals, helping to combat counterfeit drugs in the market.

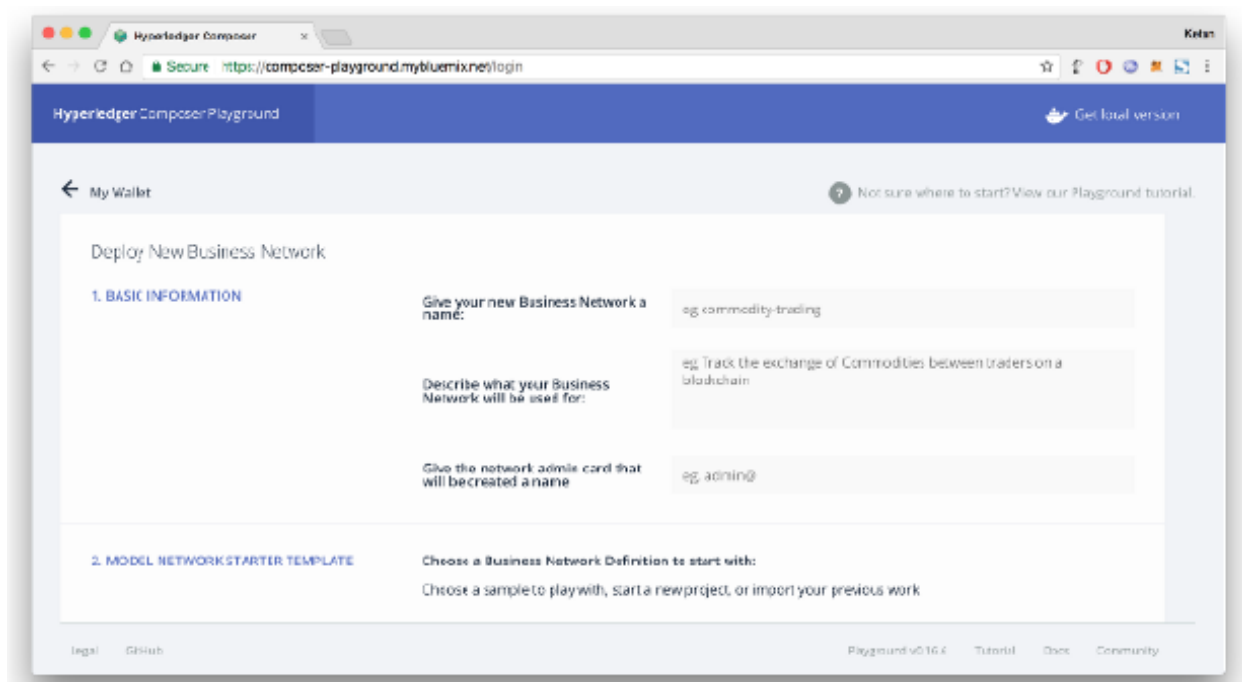
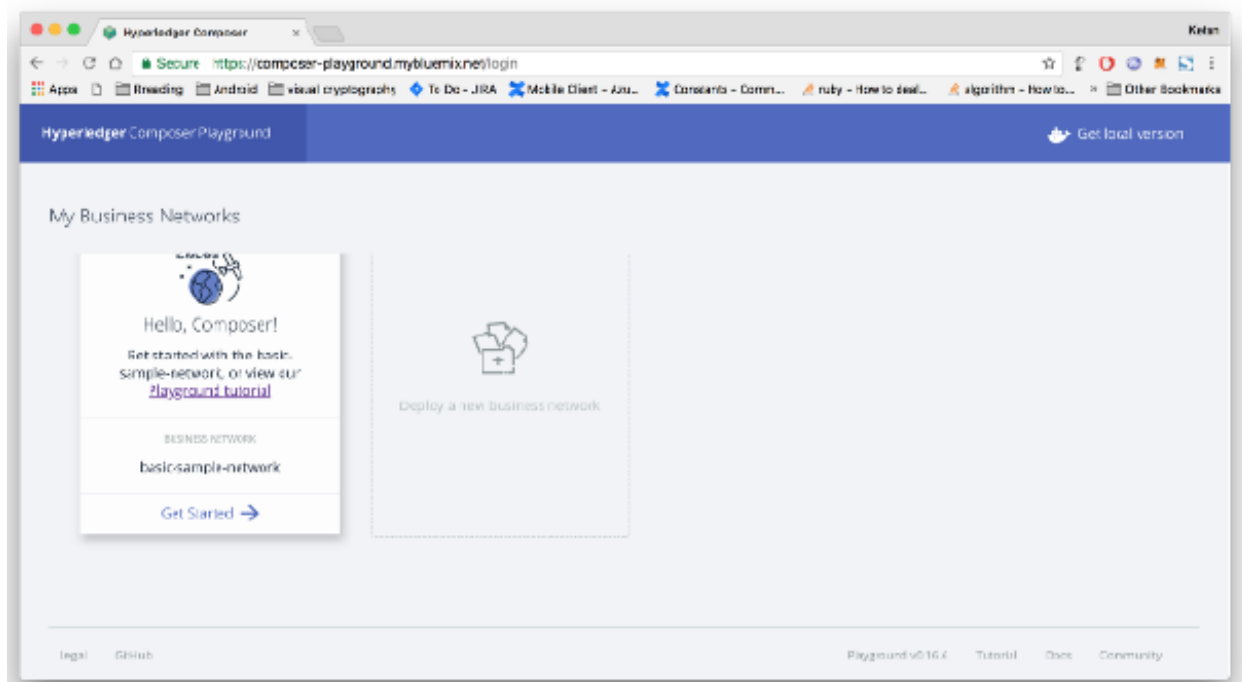
4. Government and Public Services

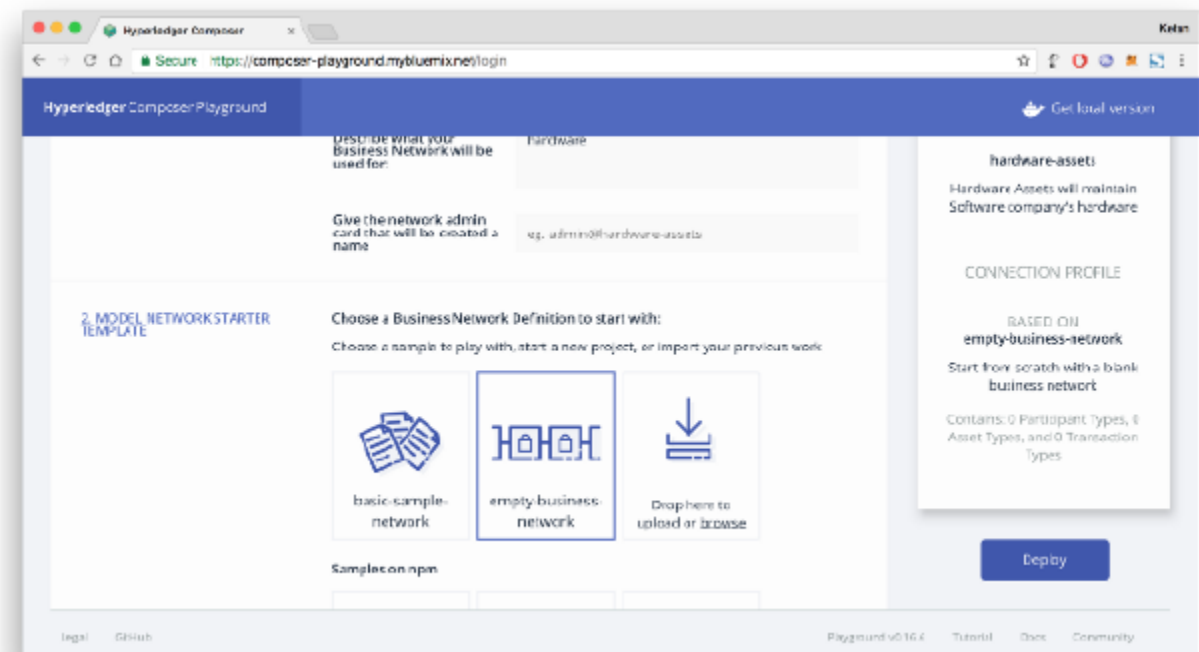
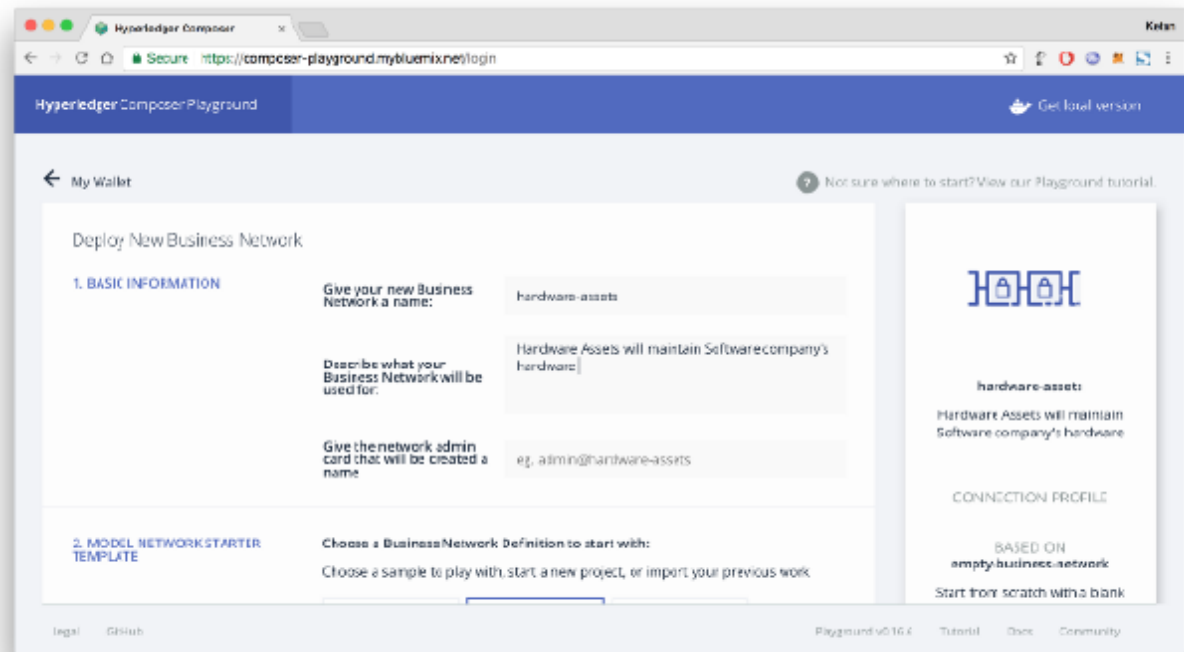
- Voting Systems: Blockchain can provide secure and transparent voting systems, reducing fraud and ensuring fair elections.
- Identity Verification: Governments are exploring blockchain for secure and tamper-proof digital identity solutions.

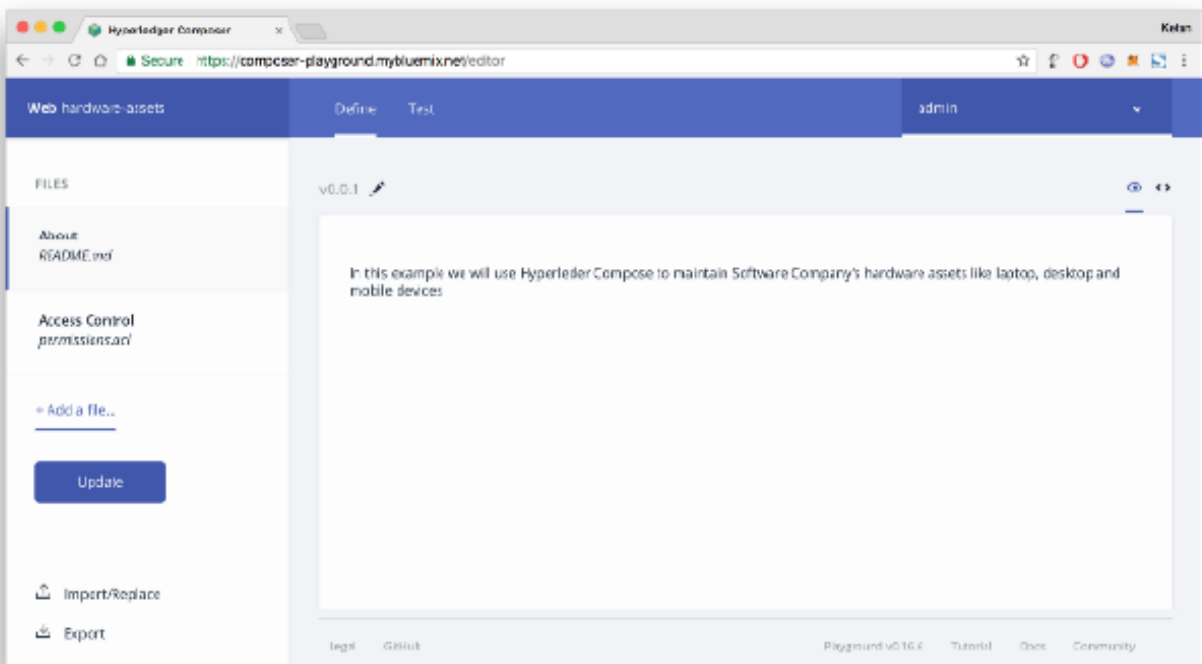
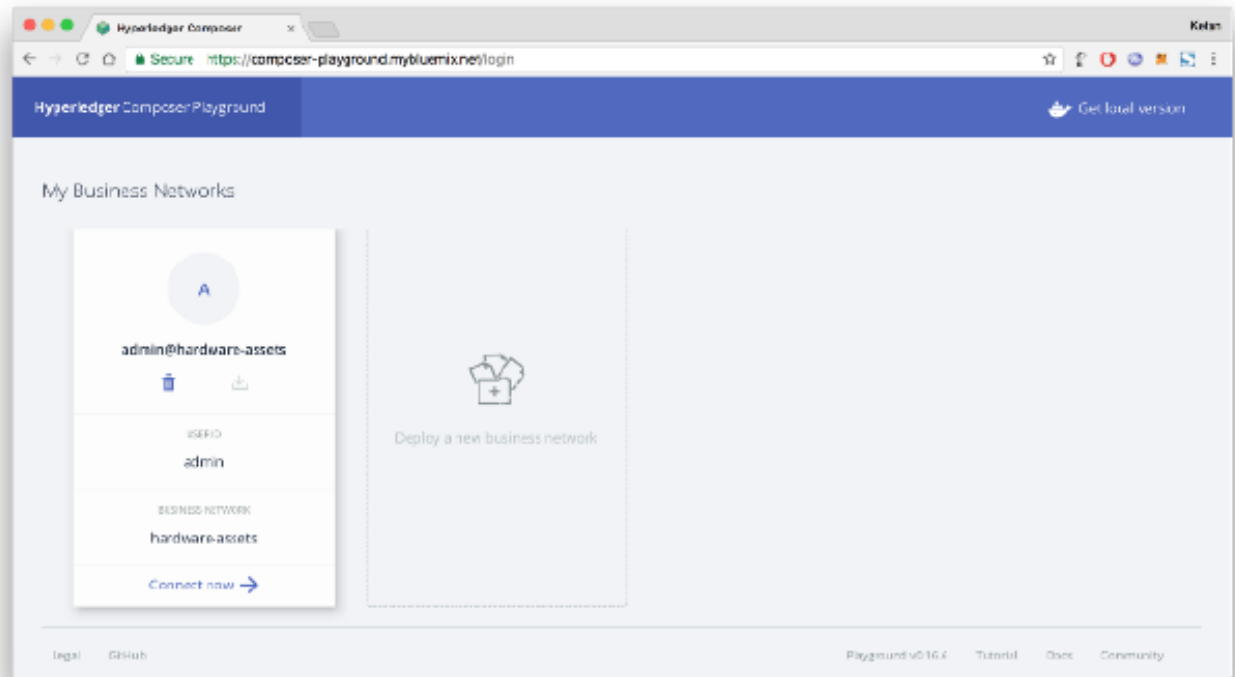
Conclusion

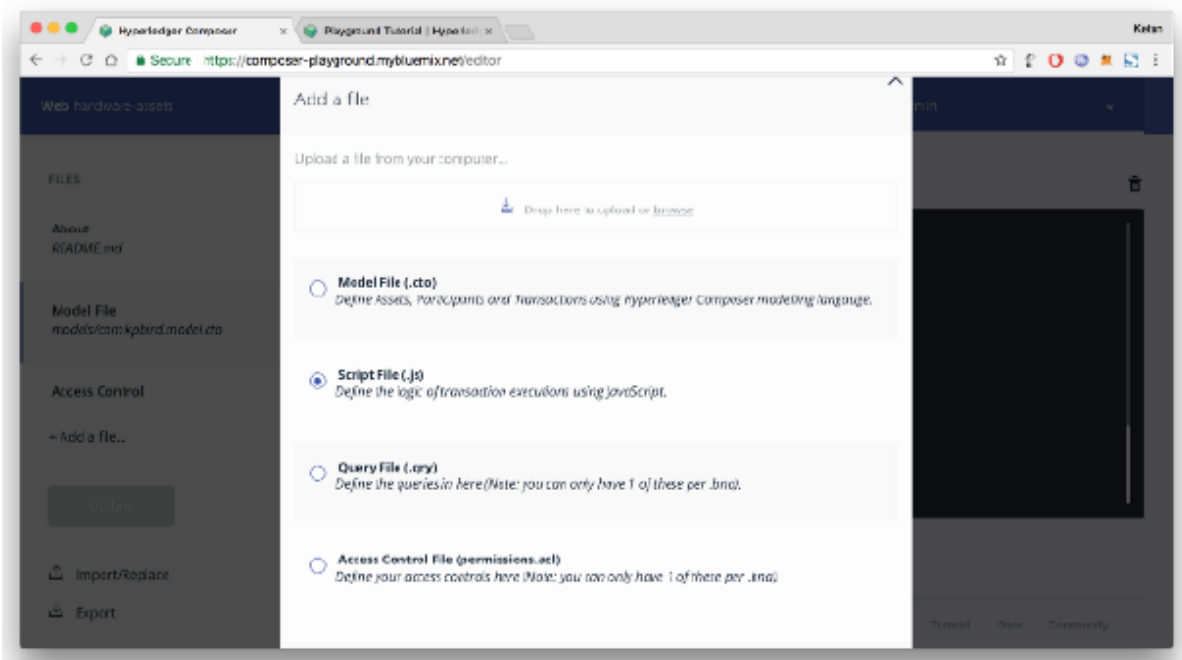
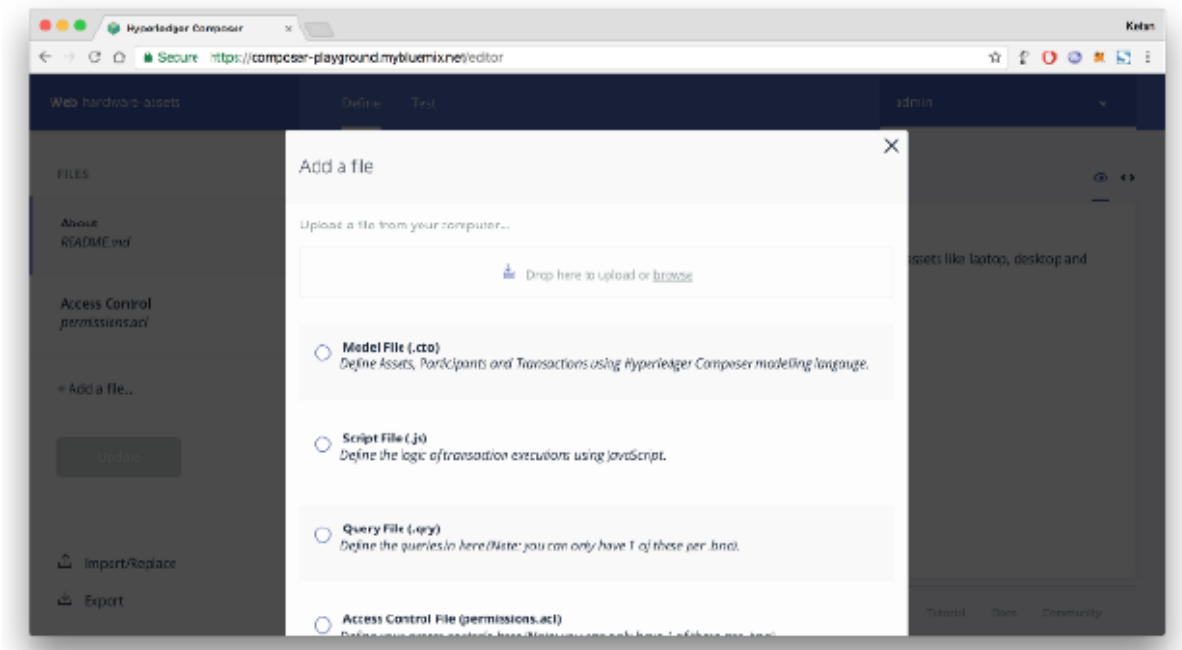
Blockchain technology has evolved to accommodate various needs, and its real-time use cases continue to expand across different sectors. Public, private, and consortium blockchains offer distinct advantages, enabling secure, transparent, and efficient solutions for businesses and individuals alike. As we move forward, it is crucial to keep an eye on the ever-expanding potential of blockchain technology and its impact on our daily lives.

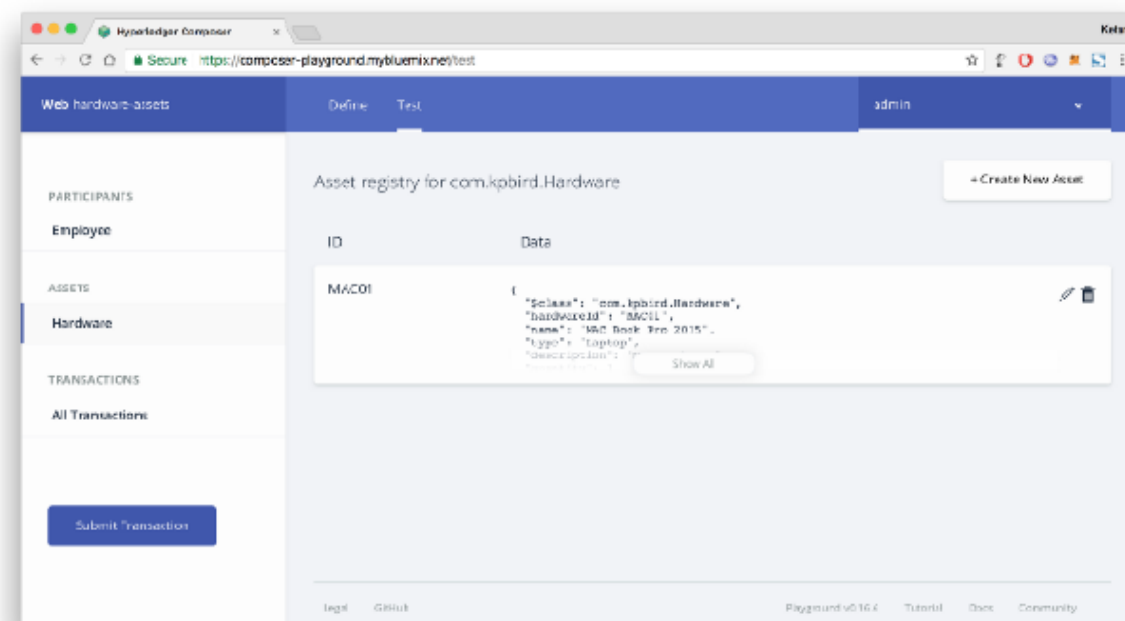
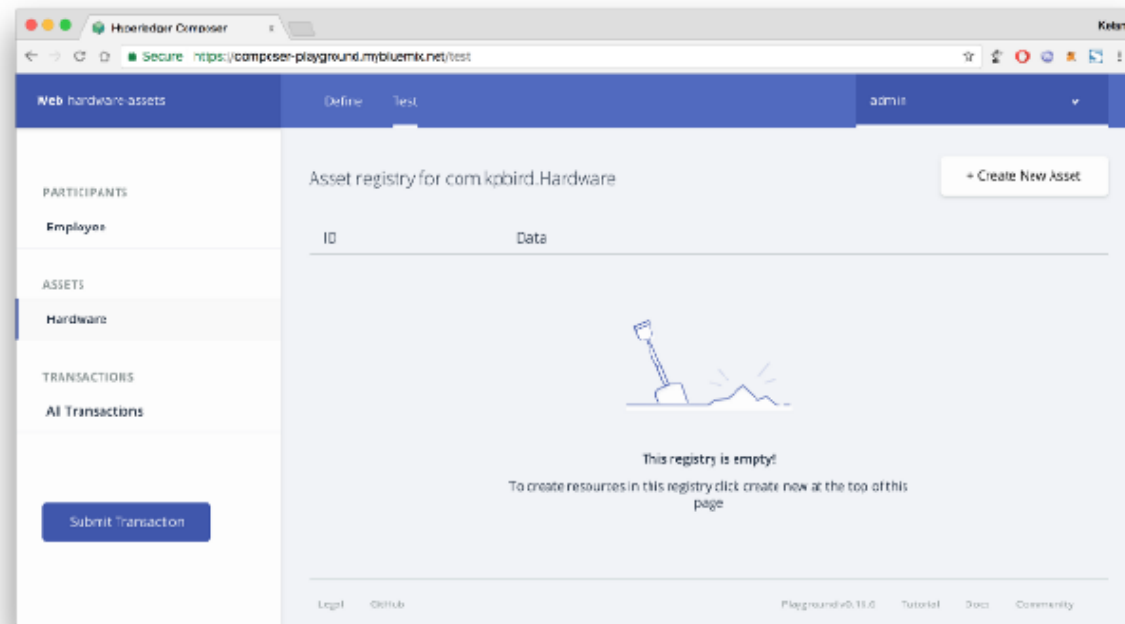
BC 5











Hyperledger Composer Playground

Web hardware assets

Define Test admin

PARTICIPANTS

Employee

ASSETS

Hardware

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for com.kpbird.Employee

+ Create New Participant

ID	Data
01	<pre>{ "class": "com.kpbird.Employee", "employeeid": "01", "firstName": "Ketan", "lastName": "Pansari" }</pre>
02	<pre>{ "class": "com.kpbird.Employee", "employeeid": "02", "firstName": "Niraj", "lastName": "Pansari" }</pre>

log out Get Help Playground v0.16.4 Tutorial Docs Community

Hyperledger Composer Playground

Web hardware assets

PARTICIPANTS

Employee

ASSETS

Hardware

TRANSACTIONS

All Transactions

Submit Transaction

Submit Transaction

Transaction Type: Allocate

JSON Data Preview

```
1 {
2   "class": "com.kpbird.Allocate",
3   "hardware": "resource:com.kpbird.Hardware#MC61",
4   "newOwner": "resource:com.kpbird.Employee#02"
5 }
```

☐ Optional Properties

+ Create New Participant

Tutorial Docs Community

Hyperledger Composer Playground

Web hardware assets

Define Test

admin

PARTICIPANTS

Employee

ASSETS

Hardware

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for com.kpbird.Employee

ID	Data
01	<pre>{ "schema": "com.kpbird.Employee", "employeeId": "01", "firstName": "Ketan", "lastName": "Pansari" }</pre>
02	<pre>{ "schema": "com.kpbird.Employee", "employeeId": "02", "firstName": "Nirja", "lastName": "Pansari" }</pre>

Submit Transaction Successful

Transaction ID: 69098091-cd22-4bec-ad09-d2eb2e83d1c was submitted

Legal GitHub Playground v0.16.4 Tutorial Docs Community

Hyperledger Composer Playground

Web hardware assets

Define Test

admin

PARTICIPANTS

Employee

ASSETS

Hardware

TRANSACTIONS

All Transactions

Submit Transaction

Date, Time	Entry Type	Participant
2018-03-25, 09:27:37	Allocate	admin (NetworkAdmin) view record
2018-03-25, 09:28:19	AddParticipant	admin (NetworkAdmin) view record
2018-03-25, 09:22:50	AddParticipant	admin (NetworkAdmin) view record
2018-03-25, 09:20:21	AddAsset	admin (NetworkAdmin) view record

Legal GitHub Playground v0.16.4 Tutorial Docs Community

