PRODUCT ONE

MAKE USING OF BLOCKCHAIN EASIER

LedgerLeopard

ABOUT US

Ledger Leopard is your partner for enterprise-ready **blockchain** solutions. We help organizations explore blockchain technology and deliver custom decentralized web and mobile applications.

We have a development team that creates custom blockchain web and mobile applications.

We have experience in a variety of fields like:

- healthcare;
- construction;
- insurance;
- finance;
- government;
- notary;
- asset management.

CONTACT US

PHONE: +31 (0)88-6440404

EMAIL: info@ledgerleopard.com

WEB: https://www.ledgerleopard.com/

INTRODUCTION

ProductOne system is intended for organizations and specialists, who are engaged in software development based on Blockchain technology.

The goal of creating ProductOne system is to help developers and system administrators in creating environment for their application. ProductOne should make creating and managing Blockchain networks easier.

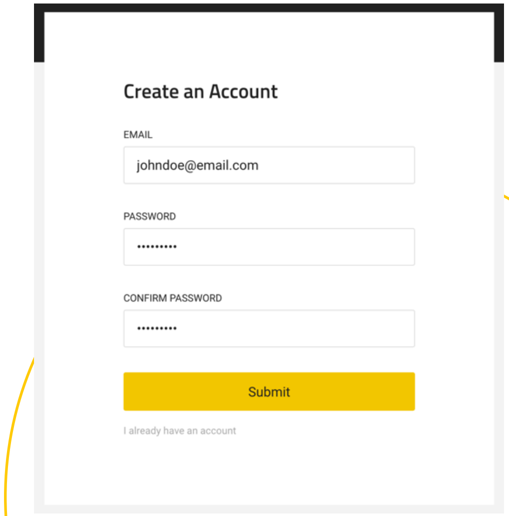
This system provides following opportunities:

- managing Blockchain networks and nodes;
- easy creating Smart Contracts;
- deploying Smart Contracts into network;
- monitoring for transactions in networks.

TABLE OF CONTENTS

HOW TO START: CREATE AN ACCOUNT

You need to create an account in ProductOne to start work. Open portal and click “Create Account” link. Enter your email and password and click button “Submit”.



Create an Account

EMAIL
johndoe@email.com

PASSWORD
....

CONFIRM PASSWORD
....

Submit

I already have an account

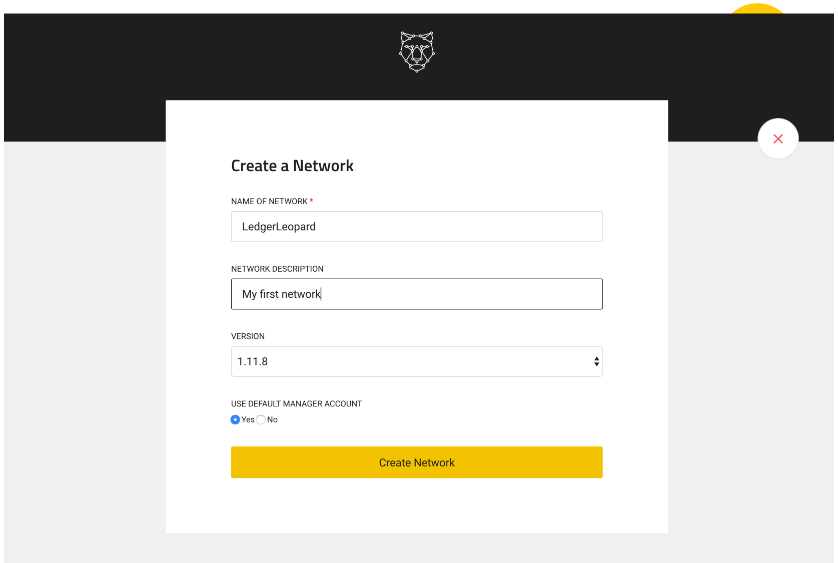
ProductOne uses Azure Active Directory B2C, which means that your profiles are stored in Microsoft.

HOW TO START: LOGIN

After creating an account login into system to start working.

HOW TO START: SETUP A NETWORK

Start with creating your first blockchain network: click button (+) to add new network.



Create a Network

NAME OF NETWORK *
LedgerLeopard

NETWORK DESCRIPTION
My first network

VERSION
1.11.8

USE DEFAULT MANAGER ACCOUNT
● Yes ○ No

Define name of network, it's description (enter the goal, for which it this network will be used, or provide some other information to your fellow members).

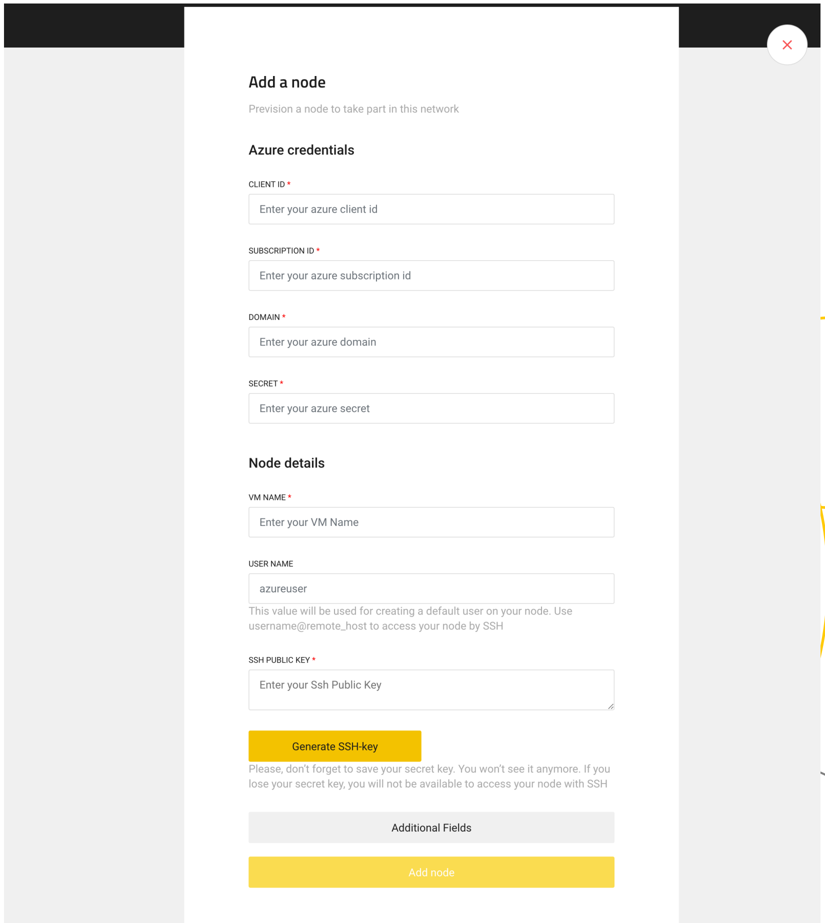
Now ProductOne allows to work with Parity Ethereum. Choose the version of Parity and decide to use default blockchain settings.

HOW TO START: ADD NODES

Now you have an empty network. But networks should contain nodes to work. Let's add our first node.

Choose your network in list and click the (+) to add a node. Then you should fill all required fields for adding node.

ProductOne suggests using Azure Cloud to create nodes, so you should have a subscription in Microsoft. Provide subscription's fields to add node in Azure Cloud.



Add a node

Prevision a node to take part in this network

Azure credentials

CLIENT ID *
Enter your azure client id

SUBSCRIPTION ID *
Enter your azure subscription id

DOMAIN *
Enter your azure domain

SECRET *
Enter your azure secret

Node details

VM NAME *
Enter your VM Name

USER NAME
azureuser
This value will be used for creating a default user on your node. Use username@remote_host to access your node by SSH

SSH PUBLIC KEY *
Enter your Ssh Public Key

Please, don't forget to save your secret key. You won't see it anymore. If you lose your secret key, you will not be available to access your node with SSH

Next step – give a name to your node. It will be displayed in network details.

You can access your node by SSH, so you should enter the username and SSH public key. Ensure, that private key is stored, and you have access to it.

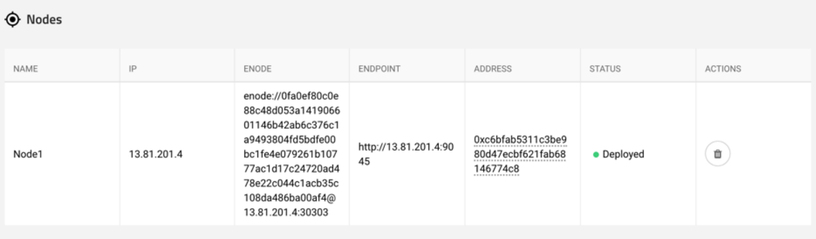
Generate SSH keys, if you don't have yours.

Fill additional fields to get more flexibility in creating nodes. You can configure the following settings:

- location – physical position of your machine;
- group name – name of resource group in Azure Cloud;
- deployment name – name of deployment in Azure Cloud;
- virtual network name – name of network in Azure Cloud;
- VM size – size of RAM and CPU features of machine;
- data disk size – size of HDD in gigabytes;
- Parity RPC, WS and Net ports – ports, where Parity will listen RPC calls, websockets and TCP calls;
- DNS – host name of machine;
- address prefix - public IP address range.

Don’t fill these fields, if you want to set them by default.

Click button “Add node” and wait for ending the deployment process. Status of the node will appear in network details in ProductOne:



Nodes

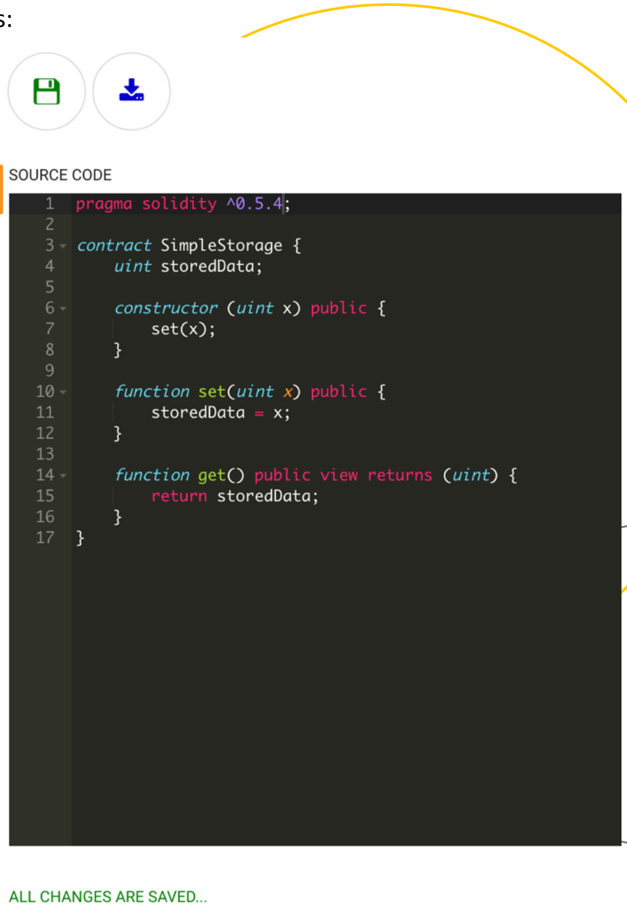
NAME	IP	ENODE	ENDPOINT	ADDRESS	STATUS	ACTIONS
Node1	13.81.201.4	enode://0fa0ef80c0e88c48d053a141906601146b42ab6c376c1a9493804fd5bdfe00bc1fe4e079261b10777ac1d17c24720ad478e22c044c1acb35c108da486ba00af4@13.81.201.4:30303	http://13.81.201.4:9045	0xc6bfab5311c3be980d47ecbf621fab68146774c8	● Deployed	🗑

There you can find parameters of your just created node: name, static IP, enode, endpoint of JSON RPC, node address in blockchain, it’s status. You can use this network right now: just access the node by SSH and get autogenerated password of this node account in blockchain. This password may be used for signing transactions in blockchain.

HOW TO START: CREATE YOUR CONTRACT

Now, when there are some nodes in network, you can create your first smart contract for Ethereum. Open network details an choose item “Add contract” in dropdown menu. You will see the form for adding contract. Add the name of contract item and define the description for it. Notice, that these contracts can be used by your fellow members, and description should describe the idea of a contract.

Now let’s write our first contract. Ethereum use Solidity language to write contracts:



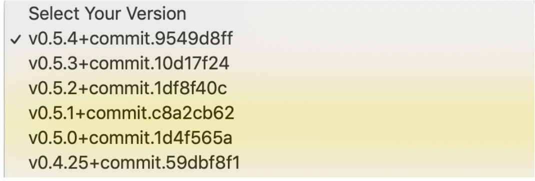
SOURCE CODE

```
pragma solidity ^0.5.4;

contract SimpleStorage {
    uint storedData;

    constructor (uint x) public {
        set(x);
    }

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

ALL CHANGES ARE SAVED...

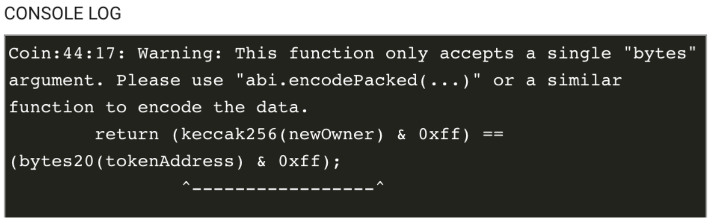
Develop your code in field for editing code, save it and continue working later. Download contract file to store it on your computer.

You can compile your contract with different versions of Solidity compiler:



Select Your Version
✓ v0.5.4+commit.9549d8ff
v0.5.3+commit.10d17f24
v0.5.2+commit.1df8f40c
v0.5.1+commit.c8a2cb62
v0.5.0+commit.1d4f565a
v0.4.25+commit.59dbf8f1

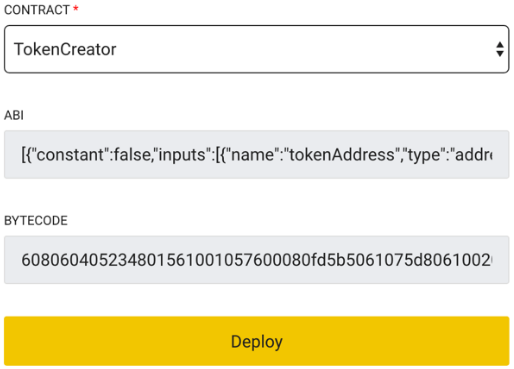
ProductOne will show you warnings and errors, if they will appear during compilation. Also you can create several contracts in one source and deploy them one by one.



CONSOLE LOG

```
Coin:44:17: Warning: This function only accepts a single "bytes"
argument. Please use "abi.encodePacked(...)" or a similar
function to encode the data.
        return (keccak256(newOwner) & 0xff) ==
(bytes20(tokenAddress) & 0xff);
                ^-----------------^
```

Save metadata of contracts or deploy them right from ProductOne editor. ProductOne will tell you about any problems, which may arise during compilation and contract deploy.



CONTRACT *

TokenCreator

ABI

[{"constant":false,"inputs":[{"name":"tokenAddress","type":"addr

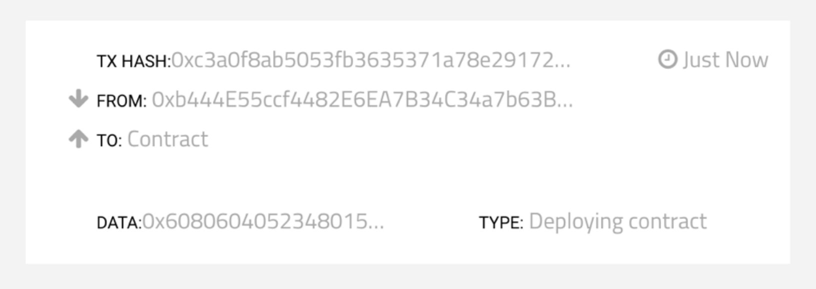
BYTECODE

6080604052348015610010576000 80fd5b5061075d8061002

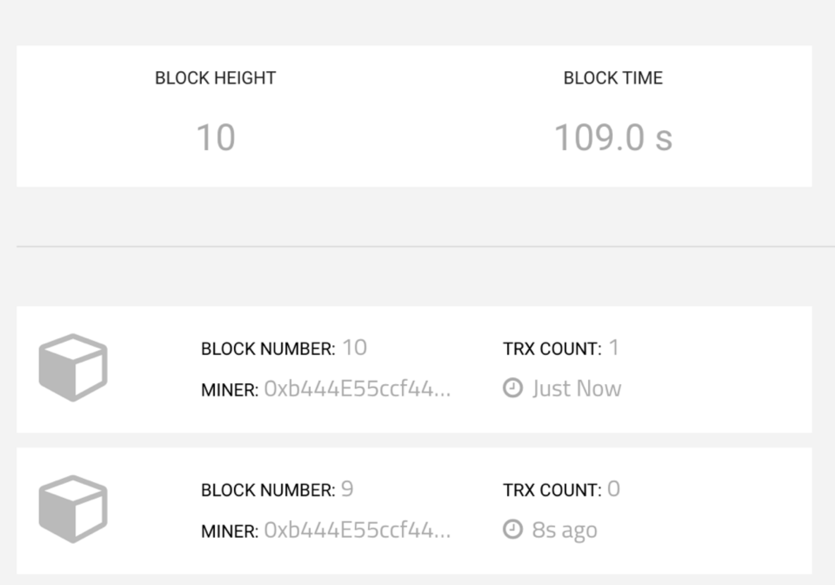
All deployed contract and their addresses you can find in network details.

HOW TO START: TRANSACTIONS EXPLORER

Use Transactions explorer to track blocks and transactions, that are mined in your network. Open Transactions explorer and wait for a first block or transaction – it will appear on screen!



TX HASH:0xc3a0f8ab5053fb3635371a78e29172... Just Now
FROM: 0xb444E55ccf4482E6EA7B34C34a7b63B...
TO: Contract
DATA:0x6080604052348015... TYPE: Deploying contract



BLOCK HEIGHT: 10
BLOCK TIME: 109.0 s

BLOCK NUMBER: 10 TRX COUNT: 1
MINER: 0xb444E55ccf44... Just Now

BLOCK NUMBER: 9 TRX COUNT: 0
MINER: 0xb444E55ccf44... 8s ago

OTHER FEATURES

In ProductOne you also can:

- edit your profile settings;
- invite other people to work together;
- delete nodes;
- delete old networks
- and other things.

FUTURE PLANS

We plan to add next things:

- association of users in the companies for credential management in Azure Subscriptions;
- adding existing machine to network and installing necessary services for it;
- notifications and more else.