



Algorand 2.0 Workshop

Russ Fustino
Technical Evangelist
Algorand

russ@algorand.com



Safe Harbor

The documents and statements presented on (or directly accessible from) this presentation may contain forward-looking statements. These statements relate to future events or Algorand Network's future performance, financial or otherwise. Any statements that are not statements of historical fact (including without limitation statements to the effect that the Company or its management "believes", "expects", "anticipates", "plans" (and similar expressions) should be considered forward looking statements. There are a number of important factors that could cause Algorand Network's actual results and outcomes to differ materially from those indicated by the forward looking statements. The Algorand or any other party contained within disclaim any obligation to update any forward looking statement.



BORDERLESS ECONOMY. BOUNDLESS OPPORTUNITY.
Algorand is defining the standard for blockchain technology.

Blockchain is Not the Future – It's The Present

Blockchain's usage is becoming ubiquitous across all sections of the economy

- Medical Field
- Charities
- Automotive
- Telecom
- Food industry

Source: Forkast.Insights
China Blockchain Report 2019-2020

Healthcare Data

- Treatment for a condition
 - Patient history
 - Medications
 - Allergies
 - Patient may not be able to remember all
 - important to store the data where the treatment team can retrieve it
- Drug traceability
 - complex supply chain
 - record of where each drug was sourced.
- Data must be accessible
 - wherever people are
 - given the sensitive nature of this data, it **must encrypted and secured.**
- blockchain technology's key strengths: **interoperability and security.**

Source: Forkast.Insights
China Blockchain Report 2019-2020

Tamper-proof charity organizations

Red Cross extensively misused funds in its response to the Sichuan earthquakes in 2008 and 2013.

The ministry's action plan for 2018–2022 aims to use blockchain and other technologies

- to improve transparency of social services
- for tracking of donations for charity

Used car salesman?

- Notorious for low trust and high incidents of fraud
- This industry is ripe for innovation
- The used-car market is growing
- Ideal for blockchain acting as an honest ledger of car maintenance history

Source: Forkast.Insights
China Blockchain Report 2019-2020

Telecom – need better ID verification and fraud detection

Among the numerous use cases for blockchain in the telecom sector, these are by far the most notable practicable applications:

- Digital identity management
- More efficient roaming
- Combating fraud

Developers, developers, developers!

“There are about 21 million professional developers in the world. Meanwhile, there are only about 10,000 blockchain developers in this industry. So 21 million [versus] 10,000? We need to attract those 21 million who are not [currently] developing for blockchain.”

Source: Forkast.Insights
China Blockchain Report 2019-2020



- **Algorand's Core Protocol**
- **What Can Algorand 2.0 Do?**
- **Algorand Foundation**
- **Algorand BetaNet, TestNet, MainNet**
- **Local Nodes and Standup Instances**
- **APIs, SDKs and Command Line tools**
- **Algorand 2.0 Features**
- **Workshop: Algorand Standard Assets**

Get the goods!

- Tutorial, Completed Solution Code, PowerPoint
- <https://github.com/rfustino/TutorialASA/>

Algorand's Core Protocol

Algorand's Core Protocol Video

Pure Proof of Stake

- <https://www.youtube.com/watch?v=gACVKaNqxPs>

What Algorand 2.0 Can Do?

What Algorand Can Do

- **Permissionless Blockchain:** Scales to billions of users
- **Enterprise Speed:** More than 1000 transactions per second, 4.27 second latency
- **Instant Finality:** There are no soft forks. Every transaction committed to the chain is final
- **Layer 1 Assets:** Organizations can easily create their own assets, with double-spending protection guaranteed by the protocol
- **Atomic Multi-Party Transactions:** Transactions can be grouped together. Either all transactions in the group are executed, or none of them are executed
- **Algorand Smart Contracts:** Small programs written in an assembly-like language that can be used as a replacement for signatures within a transaction using **Teal:** Transaction Execution Approval Language

Algorand Foundation

Algorand Foundation Rewards

- See Foundation site
<https://algorand.foundation/token-dynamics>
- All users online and offline are eligible to earn rewards

Overall Token Distribution

Estimated algos to be injected into circulation (initially via auction) over the first 5 years	3.0 billion
Estimated participation rewards (distributed over time)	1.75 billion
Relay node runners (distributed over time)	2.5 billion
Algorand Foundation & Algorand, Inc.	2.5 billion
End user grants (distributed over time)	0.25 billion

Algorand Foundation announces first Development Awards

The Foundation is interested in development areas such as

- applications
- utilities
- monitoring
- tooling
- deployment
- libraries / SDKs
- any other areas of innovation around the Algorand blockchain

Applications will be judged on creativity, utility and code conciseness.

<https://algorand.foundation/developer-incentive-awards-program>

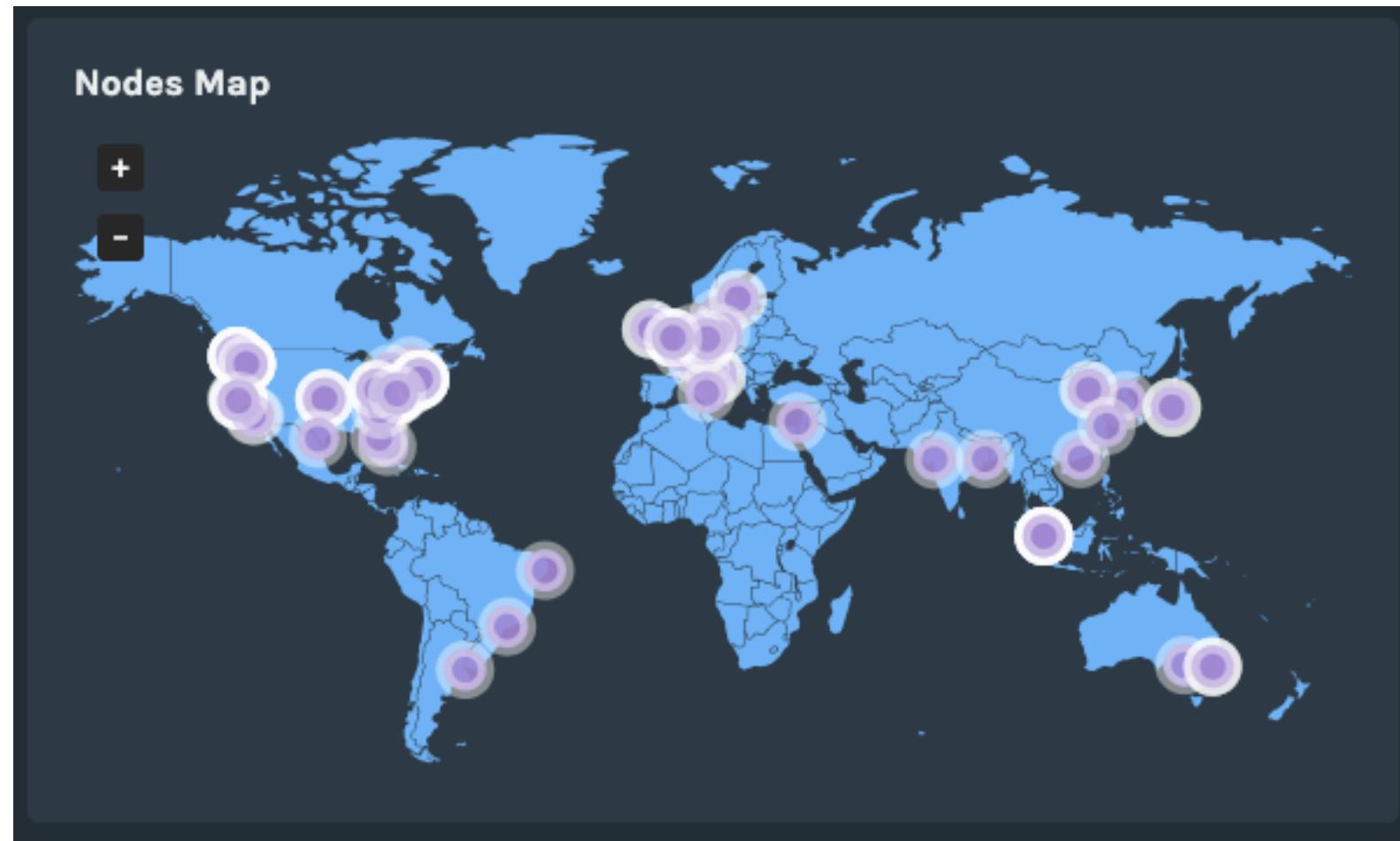
Developer Rewards program

The Foundation divides awards into four categories based on complexity and usefulness to the community:

- Extra small - 5K-20K Algo award.
- Small - 20K-50K Algo award.
- Medium - 50K-100K Algo award.
- Large - 100K-250K Algo award. This represents the maximum award a project will receive from the Foundation.

Algorand BetaNet, TestNet, MainNet

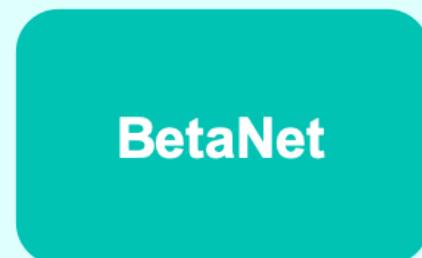
Worldwide BetaNet, TestNet and MainNet Networks



Algorand BetaNet is here!!!

<https://developer.algorand.org/docs/betanet>

Protocol Version - Future



Access the newest protocol-level features.

Protocol Version - Current



Test applications with realistic network conditions prior to deploying them to MainNet.



The Algorand MainNet, where the Algo and real assets are traded.

Private Networks

Create an isolated development environment using any available protocol version.

Explorers AlgoExplorer and GoalSeeker

TestNet and MainNet

Algoexplorer Search by Address / Tx Id / Block [SEARCH](#) [MAINNET ▾](#)

MyAlgo Web Wallet

Mainnet

Online Stake Circulating Supply	1,637,189,037.60124 Algos 3,011,268,403.743323 Algos
Latest Block	3751032
Algo Price	\$0.2432370
Daily TXs	3206

GOALSEEKER

Algorand TestNet

Staking Rewards Program

Next Distribution:
March 1, 2020

Latest Blocks

Round: 3751032	Proposer: M7XKTBXVQARLS7I...
9 secs ago	Txs: 0

Latest Transactions

TxDID: COZI4P2WSQHREZSXO...	Type: ...
From: NJCLL5UPIZKXTHQ42...	To: ...
Amount: 0 Algos	Fee: ...

<https://algoexplorer.io>

<https://goalseeker.purestake.io/algorand/>

LEDGER HOME BLOCKS ASSETS TRANSACTIONS DEVELOPER PORTAL			
Search by Block Address Tx ID Asset			
RECENT BLOCKS			
Round	Transactions	Proposed by	Time
4356608	0	LHHQJ6UMXRGEPEXBVFKT7SY26BQOIK64VVPCLVRL3RNQLX5ZMBYG6ZHBM...	12 seconds ago
4356607	0	CQW2QBBUW5AGFDXMURQBRJN2AM3OHHQWXI4PEJXRVCTEJ3E5VBTNRTEAE...	16 seconds ago
4356606	0	HXPCKXQZF4LDL3CE5ERWC5V2BQZTKXUUT3JE6AXXNLF3OJL4XUAW5WYXM...	20 seconds ago
4356605	0	DBGTTXBPGKL4TBBISC73RMB3NNZIBSH2EICWZTQRA42QKNA4S2W4SP7U...	25 seconds ago
4356604	0	DBGTTXBPGKL4TBBISC73RMB3NNZIBSH2EICWZTQRA42QKNA4S2W4SP7U...	29 seconds ago
4356603	0	2YNZ5XDUHYXL2COTVLZBRYV2A7VETFKQZQCPYMRBOKTAANHP37DUH5BOI...	33 seconds ago
4356602	0	VKM6KSCTDHEM6KGEAMSYCNEGIPFJMHDSMIRQLK76CDIRMMDHKAIRMFQ...	37 seconds ago
4356601	0	RDHKWTWXOE5AOUWUTROSR4WFLAHMUCRDZIA7OFBXXMMRBXGQ4BYQRPOXXU...	42 seconds ago
4356600	0	MZZS43WEFY56LV3WXEVLR0T3LYFLEBZ536UY3Z3J56S7EI3YYOJV06YRM...	46 seconds ago

Local Nodes and Standup Instances

Install your node!

Install new node

- Log into <https://developer.algorand.org/>
- <https://developer.algorand.org/docs/introduction-installing-node>
- Three Modes of Operation For Participation Nodes
 - Normal – Only 300 latest blocks are on disk
 - Archival – Stores all blocks to disk but can only search txs in 1k ranges
 - Indexer – Quicker searches and unbounded ranges. Additional API calls Available, two addition calls
- Configuration of node types described here:
 - <https://developer.algorand.org/docs/algorand-node-types>

PureStake

- PureStake offers API as service
 - Instant Access to TestNet & MainNet – no Sync times
 - No node installment required
 - Is configured with Archival and Indexer capabilities
 - Highly Available 24x7x365
 - Used by adding a header that is provided when you sign up
-
- Sign Up Here for Free Access: <https://developer.purestake.io/>
 - API Samples: <https://github.com/PureStake/api-examples>



Algorand API-as-a-Service



- Click to edit Master text styles

Basic	Pro	Enterprise	Custom
<p>Free <small>Fifth level</small></p> <p>Unlimited Testnet Requests per day 5,000 Mainnet Requests per day 5 Mainnet / Testnet Queries per second Support: Community Chat</p>	<p>\$49/Month</p> <ul style="list-style-type: none">• Unlimited Testnet Requests per day• 100,000 Mainnet Requests per day• 5 Mainnet / Testnet Queries per second• Support: Community Chat & Direct Support via Web and Email	<p>\$249/Month</p> <ul style="list-style-type: none">• Unlimited Testnet Requests per day• 500,000 Mainnet Requests per day• 10 Mainnet / Testnet Queries per second• Support: Community Chat & Direct Support via Web and Email with 24 hour response time• SLA – 99.95% Guaranteed Uptime	<p>Custom</p> <ul style="list-style-type: none">• Unlimited Testnet Requests per day• Custom packages for Requests per day / per second• Support: Custom chat, email and phone options including 24x7 response support• SLA – Custom based on requirements

<https://github.com/PureStake/api-examples>

Sign up at: <https://www.purestake.com/algorand-api>

Sandbox – The quickest way to get going Docker Instance

<https://github.com/algorand/sandbox>

```
Russ-MacBook-Pro-2:sandbox Russ$ ./sandbox -h
sandbox commands:
  up [mainnet||testnet||betanet] [-s||--skip-snapshot]
    -> spin up the sandbox environment, uses testnet by default.
        Optionally provide -s to skip initializing with a snapshot.
  down  -> tear down the sandbox environment
  restart -> restart the sandbox
  enter  -> enter the sandbox container
  clean   -> stops and deletes containers and data directory
  test    -> runs some tests to make sure everything is working correctly
```

```
algorand commands:
  logs      -> stream algorand logs with the carpenter utility
  status     -> get node status
  goal (args) -> run goal command like 'goal node status'
```

```
tutorials:
  introduction -> learn how to get Algos on testnet and create a transaction
```

<https://medium.com/algorand/introducing-sandbox-the-quick-way-to-get-started-on-algorand-8082c2d18854>

Hackathon / Ambassador / Devrel Algod Instances

Local node syncing? Need Archive or indexing? Sample code defaults to Hackathon TestNet

// Algorand Hackathon TestNet

```
algodAddress = "http://hackathon.algodev.network:9100"
```

```
algodToken = "ef920e2e7e002953f4b29a8af720efe8e4ecc75ff102b165e0472834b25832c1"
```

//Algorand Hackathon BetaNet

```
algodAddress = http://betanet-hackathon.algodev.network:8180
```

```
algodToken = "050e81d219d12a0888dafddaeafb5ff8d181bf1256d1c749345995678b16902f"
```

// Purestake TestNet

```
algodAddress = "https://testnet-algorand.api.purestake.io/ps1"
```

```
algodToken = "B3SU4KcVKi94Jap2VXkK83xx38bsv95K5UZm2lab"
```

// Purestake BetaNet

```
algodAddress = "https://betanet-algorand.api.purestake.io/ps1"
```

```
algodToken = "B3SU4KcVKi94Jap2VXkK83xx38bsv95K5UZm2lab"
```

Node recap

- **Create and run your own**
 - Mac
 - Ubuntu
 - Other Linux Distros (Example - Raspberry Pi)
 - Can take several hours to sync > 10 hours
- **If at hackathon or workshop, use stand up instance instances**
- **Purestake API service**
 - Indexed, Archived
 - Can access from SDK code on platforms that do not have nodes (Windows)
- **Sandbox**
 - Docker
 - Not for production
 - Snapshot start from current node
 - No sync time

DevRel Repository

<https://github.com/algorand-devrel>

- [Hackathon](#)
- [AlgorandDeveloperTasks](#)
[react-workshop](#)
- [Paytx](#)
- [JavaOfflineSign](#)
- [Raspberry-pi-rfid-setup](#)
- [js-webapp](#)
- [Chessexample](#)



LAB Exercise - Algorand Hackathon Developer

Getting Started for Hackathon
 Install Algorand node
 Alternatives
 Option 2 – Purestake Token
 Examples of a PureStake GET and POST
 Apply current updates
 Install VS Code or alternate IDE (optional)
 Start two terminal sessions
 Replace tokens and addresses in the sample code.
 Goal Command Line tools / AlgoExplorer.io
 Goal
 AlgoExplorer.io
 SDKs
 JavaScript SDK
 JavaScript SDK Sample webapp
 Encode/Decode Note Field
 Node Example: Retrieving Latest Block Information
 More Examples
 Go SDK
 kmdclient.go - kmd client go
 backupwallet.go - Backing up a Wallet
 signsubmit.go - Signing and submitting a transaction
 signoffline.go - Sign a transaction offline
 submittransfilefrom.go - Submit the transaction from a file
 manipulatemultisig.go - Manipulating multisig transactions
 Java SDK
 GetBlock.java – gets the status and lastround

LAB Exercise - Algorand Hackathon Developer

Getting Started for Hackathon

Time Estimate: 1 hour

This section provides guidance on installing an Algorand node and the tools that will be useful in your hackathon efforts. Happy coding! Download [algorandsamples.zip](#) at <http://github.com/algorand-devrel/hackathon> and unzip into a folder off of your \$HOME folder. To get your Home folder location, in terminal type in:

```
echo $HOME
```

Install Algorand node

Time Estimate 10 minutes

In this section, we will install an Algorand node.

A synchronized node will be provided for this hackathon, however, you need to install your own node, so you can do this lab exercise and continue to work on the solution after the hackathon is over, as well as build other solutions. Follow the instructions on how to install your node are here:

<https://developer.algorand.org/docs/introduction-installing-node>

By default, an Algorand installation is configured to run on MainNet. For most users, this is the desired outcome. Developers, however, need access to our TestNet or DevNet networks. This [guide](#) will walk you through how to switch networks, if you have not already done so.

Your node may take a while to sync (several hours). You can proceed to the following steps in the meantime, noting if the goal command does not seem to be working, like transferring funds for example, it may be because the node is not synced yet. To see if it is synced use this command from terminal:

```
goal node status -d ~/node/data
```

The data directory is the data directory for the node. It may be simpler to set ALGORAND_DATA env variable rather than specifying each time. In that case, the -d should be removed above.

```
export ALGORAND_DATA=~/node/data
```

If your solution will need to search transactions and blocks, such as the sample Chess Game here:

Algorand Devrel on Medium

<https://medium.com/algorand/developers>

Articles published every Tuesday

- Sandbox
- 2019 Wrap-up
- Smart Xcontracts
- Atomic transfers
- Algorand Assets with React
- Standard Assets
- BetaNet News
- Pay Transactions
- Hack the North Sample Code
- Algorand Blockchain Basics

Developers

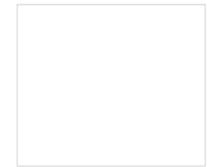
Introducing Sandbox: The quick way to get started on Algorand

Running a node on Algorand is a very fast and straightforward process.



Sam Abbassi

Jan 7 · 5 min read



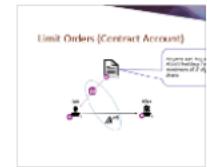
2019 Wrap-Up for Algorand Devs

As 2019 comes to a close, here is a wrap-up of the three latest features launched on Algorand at layer-1 with links to content to help...



Liz Baran

Dec 27, 2019 · 4 min read



Understanding Algorand Smart Contracts

As part of the Algorand Version 2.0 release, the network now supports Algorand Smart Contracts (ASC1), which are layer-1 smart contracts...



Jason Weathersby

Dec 10, 2019 · 12 min read



Algorand Atomic Transfers

Atomic Transfers are irreducible batch transactions that allow groups of transactions to be submitted at one time.

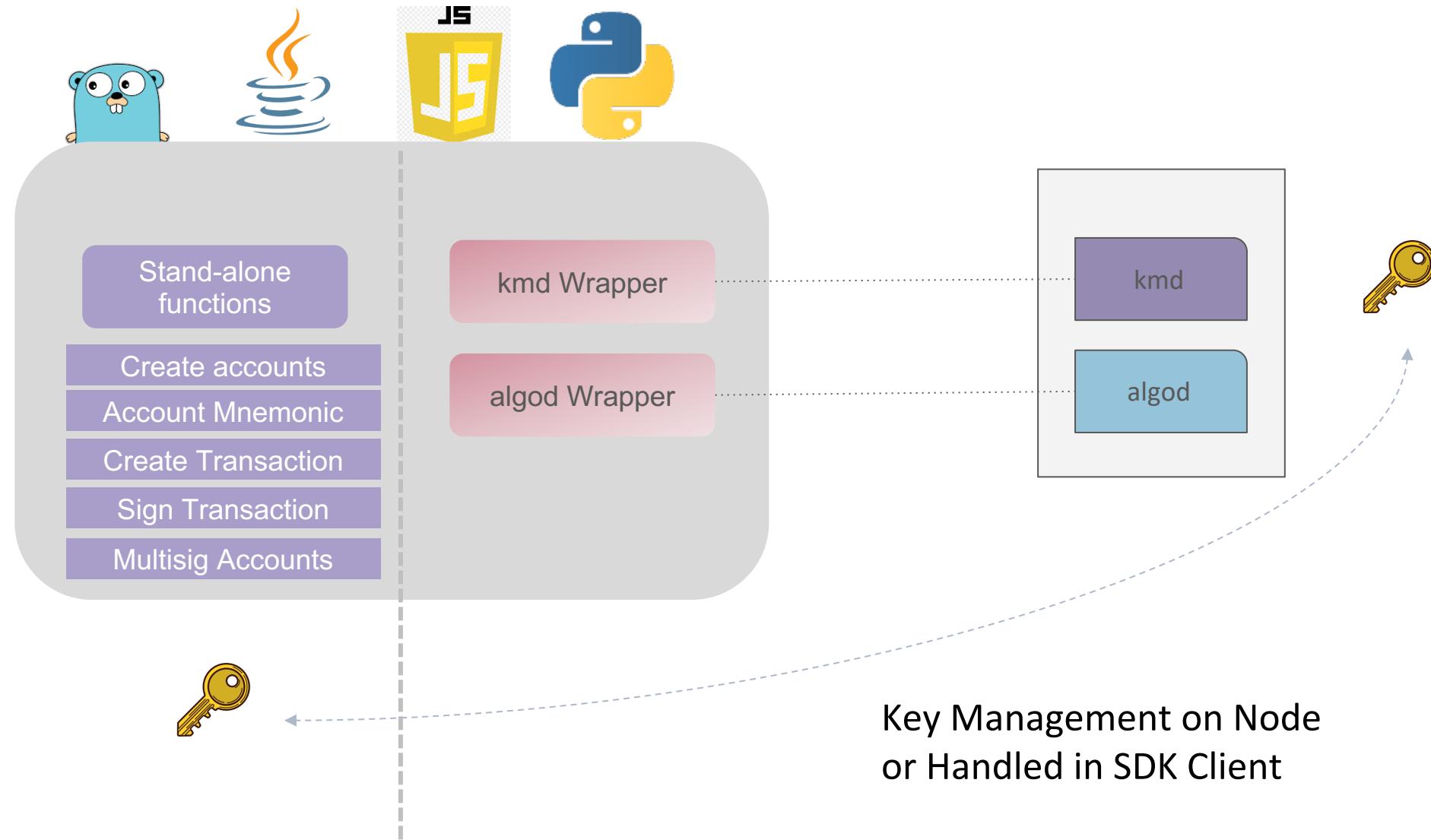


Russ Fustino



APIs, SDKs and Command Line

SDK Functionality



New community C# SDK now available!

<https://github.com/RileyGe/dotnet-algorand-sdk>

The screenshot shows the GitHub repository page for 'RileyGe / dotnet-algorand-sdk'. The repository has 30 commits, 1 branch, 0 packages, 0 releases, 2 contributors, and is licensed under GPL-3.0. The latest commit was made 26 days ago. The repository contains files like .gitignore, LICENSE, README.md, README_cn.md, and dotnet-algorand-sdk.sln. The README.md file contains the following content:

Dotnet Algorand SDK

Finally, the dotnet algorand sdk is ready for use, give it a try!

1. Prerequisites

This library is compliant to .Net Standard 2.0, so you can use this library on any platform which can use the .Net Standard 2.0.

2. How to Install

Open the NuGet command line and type:

How do you download the swagger files?

There are two swagger definition json files,
one for Algod and one for KMD

For Alogd use this:

<http://localhost:8080/swagger.json>

curl [http://\\$\(cat ~/node/data/algod.net\)/swagger.json](http://$(cat ~/node/data/algod.net)/swagger.json) > swagger.json

For kmd use this:

<http://localhost:7833/swagger.json>

curl [http://\\$\(cat ~/node/data/kmd-v0.5/kmd.net\)/swagger.json](http://$(cat ~/node/data/kmd-v0.5/kmd.net)/swagger.json) > swaggerkmd.json

```
{  
    "consumes": [  
        "application/json"  
    ],  
    "produces": [  
        "application/json"  
    ],  
    "schemes": [  
        "http"  
    ],  
    "swagger": "2.0",  
    "info": {  
        "description": "API Endpoint for AlgoD Operations.",  
        "title": "Algod REST API.",  
        "contact": {  
            "email": "contact@algorand.com"  
        },  
        "version": "0.0.1"  
    },  
    "host": "localhost",  
    "basePath": "/",  
    "paths": {  
        "/health": {  
            "get": {  
                "produces": [  
                    "application/json"  
                ]  
            }  
        }  
    }  
}
```

SDK Documentation at [Developer.Algorand.org](https://developer.algorand.org/docs/javascript-sdk)

The screenshot shows a Mac OS X style window for a web browser. The title bar says "JavaScript SDK | Algorand Dev". The address bar shows the URL "developer.algorand.org/docs/javascript-sdk". The page content is the documentation for the JavaScript SDK.

Table of Contents:

- Install the SDK
- Using Stand-Alone Functions
- Using the Client Wrapper Functions
- Examples with Web Apps
 - Using Standalone functions
 - Using Client Wrapper Functions
- Examples with Node
 - Using Standalone Functions
 - Retrieving Latest Block Information
 - Creating a New Wallet and Account Using kmd
 - Backing Up and Restoring a Wallet
 - Signing and Submitting a Transaction
 - Locating a Transaction
 - Writing to the Note Field of a Transaction
 - Reading the Note Field of a Transaction

Testing Your Applications

- Tutorials
 - Creating a Private Network
 - Creating a new Wallet
 - Creating a New Account and a Participation Key
 - Write a Raw Transaction and Post to Algod REST Server
 - Creating a Multisignature Account
 - algod REST API Example
 - kmd REST API Example
- Using the SDKs and REST APIs
 - algod REST Paths
 - algod REST Models
 - kmd REST Paths
 - kmd REST Models
 - Go SDK
 - JavaScript SDK
 - Java SDK
 - Developer FAQ
 - Latest Updates

Install the SDK

See the [readme](#) in the [github repository](#) for additional information.

Node.js

```
npm install algosdk
```

Browser

Download the SDK's [minified JavaScript file](#) from our repository.

```
<script src="algosdk.min.js"/>
```

Using Stand-Alone Functions

The stand-alone functions provide functions that are not reliant on the kmd or algod processes. These include the following functions:

- **generateAccount** - This function can be used to create an account that can be used with the Algorand blockchain. It returns a JSON object containing an address and a secret key.
- **isValidAddress** - This function allows you to verify that an address is a valid address. It returns either true or false.
- **mnemonicToSecretKey** - This function takes a 25 word mnemonic and returns a JSON object containing an account and a secret key. This functions primary purpose is to recover an account.
- **secretKeyToMnemonic** - This function retrieves the 25 word mnemonic when passed an account's secret key. When creating a new account this is often used to store the backup phrase.

Algorand 2.0 Features

ASA

ASC1

Atomic Transactions

Algorand Standard Assets (ASA)

- A token is fungible if any two units of that token are interchangeable.
- Fungible tokens are the basis of many financial applications.
- Uses of fungible tokens include
 - Currency
 - Points in a loyalty program
 - Shares of an asset
 - Securities such as stocks, bonds, and derivatives
- On other blockchains, these tokens are created using smart contracts (e.g., ERC-20).
 - Errors in contracts defining new currencies, and incompatibilities with existing apps, can and have resulted in significant losses.

Algorand Standard Assets (ASA)

- Any user can create a new fungible token.
 - The potential supply of the new token is set at creation.
 - A token may be freezable. (A token that is not freezable can never be made freezable.)
 - The token creator is its initial token manager.
- A token manager can
 - Increase the circulating supply of the token (up to the potential supply).
 - Freeze (and unfreeze) the tokens in a given account, if the token is freezable.
 - Remove tokens it owns from the circulating supply.
 - Destroy the token if no tokens are circulating.
 - Transfer token management to another user.
- To receive a token, an account must allocate space to maintain a balance for that token.
 - Maintaining a balance for a token increases the account's minimum balance.

Atomic Transfers

- Multiple transfers can be grouped together.
- Either all transfers occur, or none of them do.
- Transfers may involve multiple fungible tokens (including Algos).
- Each transfer is signed separately by the originator of the transfer.
- Eliminates the need for hashed time-locks for atomic swaps in other blockchains.
 - Avoids the delays inherent with hashed timed-locks.

Some applications of Atomic Transfers

- Atomic swap or bilateral exchange
 - Purchasing a digital asset with cryptocurrency, or trading digital assets.
- Circular trades
 - For example, Alice pays Bob if and only if Bob pays Clare if and only if Clare pays Alice.
- Group payments
 - Either everyone pays or no one does.
- Payments to multiple recipients
- Decentralized exchanges
 - Atomic multi-party transfers enable trades without trusted intermediaries.
- Internal units of accounting
 - Several companies can settle their accounts with each other with a single transaction.
 - Advantageous for supply chains, where delays in payments introduces massive friction.

Algorand Smart Contracts Layer 1 (ASC1)

- Algorand Smart Contracts are small programs written in an assembly-like language that can be used as a replacement for signatures within a transaction.
- This language is named Transaction Execution Approval Langauge or TEAL.
- TEAL programs have one primary function and that is to return true or false and are used to analyze and approve transactions. Standard transactions are signed with either a private key or multisig set of private keys. With the introduction of ASC1, they can now be signed with a TEAL program. This is called a logic signature.

ASC1 Usage scenarios

Examples:

- Escrow accounts
- Loan payments
- Limit and stop orders
- Subscription payments
- Collateralized obligations

TEAL code has two basic usage scenarios...

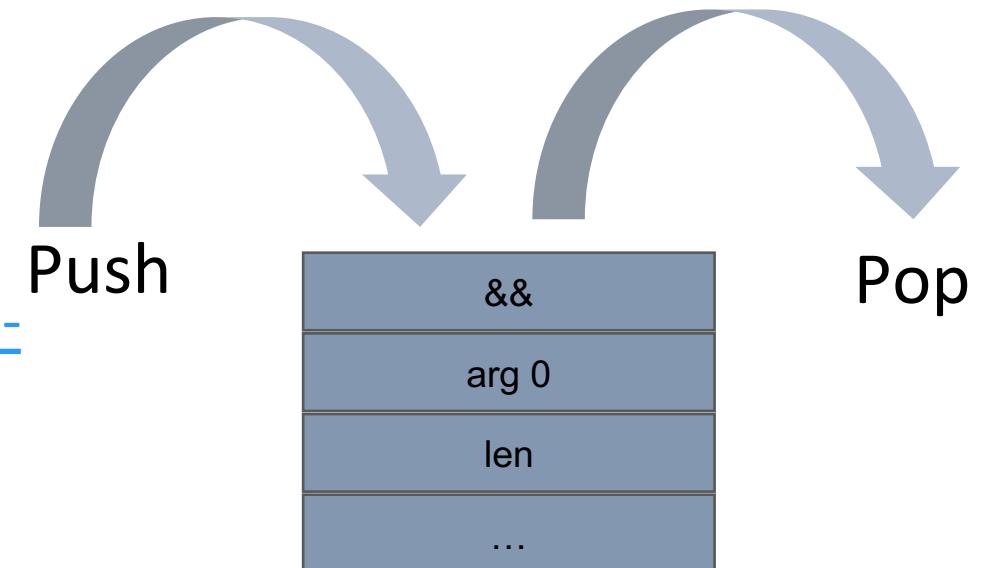
- Contract account
- Delegated signature

Video on TEAL

<https://youtu.be/OWFRP9McBmk>

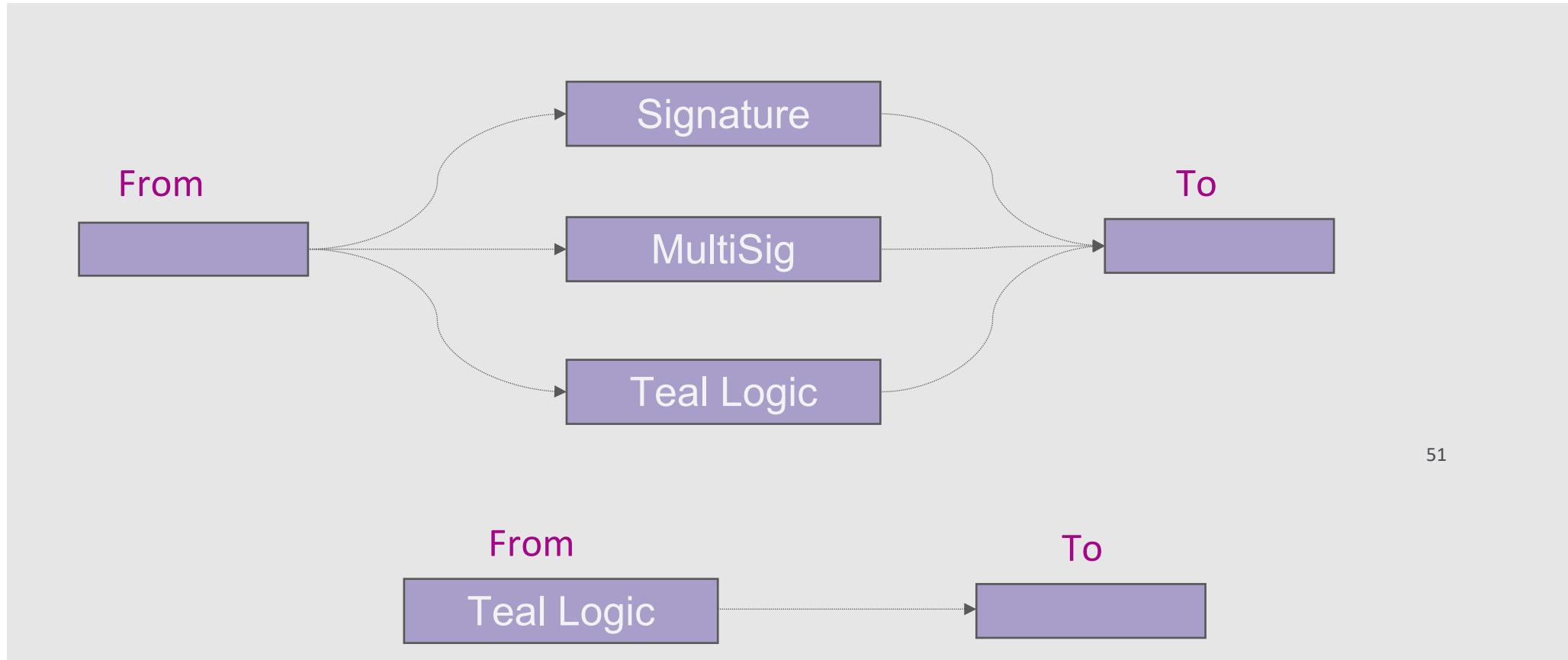
Teal – Transaction Execution Approval Language

- Bytecode based stack language
- Replaces signature in transaction
- Similar to Bitcoin Script
- Returns True or False
- Support in the SDKs to run specific Templates
- <https://developer.algorand.org/docs/algorand-smart-contract-templates>



Teal. - Transaction Approval Pipeline

- Many Templates (Escrow limit withdrawals, Split withdrawals, Periodic Payments, limit orders, etc) <https://github.com/algorand/go-algorand/tree/master/tools/teal/templates>



Resources

Become An Algorand Ambassador

- Work directly with the Algorand team and learn from the best in the field
- Host your own hackathon or developer meetup to help others in your community build on top of Algorand
- Connect with other developers from all over the world who are also building on Algorand
- Get the latest news and updates about our platform before anyone else hears about it

→ Visit community.algorand.org



Contact Stephen Duignan
stephen@algorand.foundation

ASA Resources

- Using in goal <https://developer.algorand.org/docs/asa>
- To test this feature out on BetaNet
<https://developer.algorand.org/docs/getting-started-betanet>
- *See here for SDK documentation: [ASA SDK Usage](#)*

Atomic Transfers resources

- *Using in goal:* <https://developer.algorand.org/docs/atomic-transfers>
- *See here for SDK Documentation:* [Atomic Transfer SDK Usage](#)

ASC1 Resources -

<https://developer.algorand.org/docs/asc>

- [ASC1 SDK Usage](#) - Reference docs for ASC1s in Go, Java, JavaScript, and Python
- [ASC1 Escrow Example](#) - A walkthrough example of implementing an escrow account with TEAL.
- [ASC1 Tutorial](#) - A walkthrough using **goal** to submit a TEAL transaction to the network.
- [Transaction Execution Approval Language \(TEAL\)](#) - Reference documentation for TEAL, the language powering Algorand Smart Contracts
- [TEAL Opcodes](#) - The full list of opcodes available in TEAL.
- [Algorand Smart Contract Templates](#) - List of currently supported templates with SDK and TEAL documentation.

Workshop

Algorand Standard Assets – Tutorial

<https://github.com/rfustino/TutorialASA>

SDKs

- JavaScript
- Python
- Java
- Go

Algorand Standard Assets

- Fungible Assets
- Non-Fungible Assets
- Increased security
- Inherent compatibility
- High ease of use

Tutorial - ASA

- Step 1: Create Accounts and Setup code
- Step 2: Create a New Asset
- Step 3: Configure Asset Manager
- Step 4: Opt-in to Receive Asset
- Step 5: Transfer an Asset
- Step 6: Freeze an Asset
- Step 7: Revoke an Asset
- Step 8: Destroy an Asset

Step 1A: Create Accounts

Time estimate 20 min

- Install SDK
- File New in VS Code or editor of choice
 - CreateNewAccounts.js
 - CreateNewAccounts.py
 - CreateNewAccount.java
 - CreateNewAccount.go
- Create Three Accounts
- Copy off the Account Address and Mnemonic
- Load with Algos from the TestNet Dispenser

Step 1B: Setup Accounts, Utility Functions and Tools

Time Estimate 20 minutes

- Use GoalSeeker and AlgoExplorer to view accounts
- File New in VS Code or editor of choice
 - TutorialASA.js
 - TutorialASA.py
 - TutorialASA.java
 - TutorialASA.go
- Paste in your account mnemonics
- Make Client
- SuggestedParms
- WaitForTransaction
- Submit Transaction
- Add your Address and Token
- RESTORE ACCOUNT from mnemonic

Step 2 Create a New Asset Transaction

Time estimate – 20 min

Parameters for create new asset

- AssetTotal
- defaultFrozen
- Decimals
- UnitName //8 characters
- assetName //friendly name
- url
- Asset managers
 - Manage
 - Reserve
 - Freeze
 - Clawback

Step 3 Configure Asset Manager

Time Estimate 20 min

- Assign roles for:
 - Manage
 - Reserve
 - Freeze
 - Clawback
- Address must be re-specified in each new configuration transaction.
- Supplying an empty address is the same as turning the associated feature off for this asset.
- `clawback=""` means the associated asset could never be revoked from asset holders
- The `strictEmptyAddressChecking` argument will enforce that these address must not be empty.

Step 4 Opt-in to Receive Asset

Time estimate 10 min

- Special case for Asset transaction
- The user sends 0 assets to their own account
- Once accepted that account can now perform transactions for that asset.

Step 5 Transfer an Asset

Time Estimate 20 minutes

- Once the account has Opt-in, it can receive assets
- The optional closeRemainderTo argument can be used to stop transacting with a particular asset.
- Let's Say I have 1000 asset units
- I send 10 from Account 1 to Account 3, with close remainder to Account 2
- 10 will go to Account 3
- 990 will go to Account 2

Step 6 Freeze an Asset

Time Estimates 10 minutes

- Used to freeze or unfreeze an asset.
- This transaction must be sent from the account specified as the freeze manager for the asset.

Step 7 Revoke an Asset

Time Estimates

- Revoking an asset allows an asset's revocation manager to transfer assets on behalf of another user.
- It will only work, when issued by the asset's revocation manager.

Step 8 Destroy an Asset

Time Estimate 10 minutes

- Asset destruction allows the creator to remove the asset from the ledger, if all outstanding assets are held by the creator.

- What Can Algorand 2.0 Do?
- Algorand Foundation
- Algorand BetaNet, TestNet, MainNet
- Local Nodes and Standup Instances
- APIs, SDKs and Command Line tools
- Algorand 2.0 Features
- Resources
- ASA Workshop



Sign up at
developer.algorand.org
community.algorand.org

twitter.com/algorand | facebook.com/algorand | youtube.com/algorand | linkedin.com/company/algorand

©2019 Algorand, All rights reserved



Questions?