
FABTOKEN

Contributors

Name	Organization
Nitin Gaur	IBM
Bobbie Cochrane	IBM
Angelo De Caro1	IBM
Elli Androulaki	IBM
Jim Tomonto	IBM

Taxonomy Formula: $tF'\{\sim d, f, u, r, e\}$

Token Specification Summary

Token Classification

Template Type:	SingleToken	This token has no sub or child tokens.
Token Type:	Fungible	Tokens have interchangeable value with one another, where any quantity of them has the same value as another equal quantity if they are in the same class or series.
Token Unit:	Whole	There can be many instances of this token, but they cannot be subdivided.
Value Type:	Reference	This token is a receipt or title to a material item, property or right. The token represents a reference to the value, can be owned or used digitally via its token. Sometimes referred to as a digital twin.
Representation Type:	Unique	Token instances are unique having their own identities and can be individually traced. Each unique token can carry unique properties that cannot be changed in one place and their balances must be summed. These are like bank notes, paper bills and metal coins, they are interchangeable but have unique properties like a serial number.

FabToken is a token management system that allows you to issue, transfer, and redeem tokens. These are Unique Whole Fungible Token with Variable Supply where supply is set when the tokens are fabricated, and supply can be added to new owners and removed by owners by redeeming. It is Whole by setting the Decimals property on the sub-dividable behavior = 0.

Example

Loyalty points are a common use of this type of token. Representing a loyalty point using fractional amounts like `.081231` does not make sense, so a point is just that a single whole unit. Redemption of these is easy for users to understand using whole numbers. New points can be fabricated or issued based on customer activity and points can be removed when they are redeemed. This formula supports transferable points as well.

Analogies

Name	Description
Airline Points	A customer can earn a point for each mile travelled and then redeem these points for upgrades or new tickets. Points awarded for each flight can be fabricated in a single token issue for the customer.

FabToken is:

- Non-Subdividable
- Unique-Transferable
- Roles
- Redeemable
- Fabricate

FabToken Details

Unique Whole Fungible

Type:	Base
Name:	Unique Whole Fungible
Id:	2d291501-4cca-43cf-8330-e2440e58d7df
Visual:	$\tau;F'_{d}$

Tooling: tF'{'~d}

Version: 1.0

Definition

Unique, Whole Fungible tokens have interchangeable value with each other, where any owned sum of them from a class has the same value as another owned sum from the same class. A whole token cannot be sub-divided, so it doesn't support the notion of 'making change'. Because this token is unique, it will have its own identity and can have unique properties like a serial number. Implementations should support a GetBalance or List for owners to see their balances or tokens they own.

Example

An inventory item or SKU, where an item is treated as a whole, because it makes no sense to own a fraction of a SKU or loyalty point.

Analogies

Name	Description
Loyalty Points	Most credit card or retail loyalty point programs deal with whole numbers so that redeeming points is easy to understand for their customers.
General Admission Movie Ticket	Purchasing a general admission ticket to a movie only allows for you to have a seat, but the seat that you get depends on factors like when you arrive. You're not likely to want to share a seat with another adult.

Dependencies

Artifact Type	Symbol	Description
Base	t	Base Token Definition

Incompatible With

Artifact Type	Symbol	Id
Behavior	~d	d5807a8e-879b-4885-95fa-f09ba2a22172

Influenced By

Description	Symbol	Applies To
-------------	--------	------------

Artifact Files

Content Type	File Name	File Content
Control	whole-fungible.proto	
Uml	whole-fungible.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	FabToken Alpha	ChaincodeGo	https://github.com/hyperledger/fabric/tree/v2.0.0-alpha/token

Implementation Map

Map Type	Name	Platform	Location
Implementation	FabToken Alpha	ChaincodeGo	https://github.com/hyperledger/fabric/blob/v2.0.0-alpha/docs/source/token/FabToken.md

Resource Map

Map Type	Name	Location	Description
----------	------	----------	-------------

Draft
Base Details

Token Name:	
Token Type:	Fungible
Representation Type:	Unique
Value Type:	Reference

Token Unit: Whole

Symbol:

Owner:

Quantity: 0

Decimals: 0

Constructor Name: Constructor

Behaviors

Non-Subdividable

Type: Behavior

Name: Non-Subdividable

Id: d5807a8e-879b-4885-95fa-f09ba2a22172

Visual: <i>~d</i>

Tooling: ~d

Version: 1.0

Definition

An ability or restriction on the token where it cannot be subdivided from a single whole token into fractions. Sets the base token Decimals property to 0 which will make the token non-sub-dividable and a whole token is the smallest ownable unit of the token.

Example

Non-subdividable is common for items where subdivision does not make sense, like a property title, inventory item or invoice.

Analogies

Name	Description
Non-Fractional	It is not possible to own a fraction of this token.
Barrel of Oil	Barrels of Oil don't make sense to subdivide.

Dependencies

Artifact Type	Symbol	Description
---------------	--------	-------------

Incompatible With

Artifact Type	Symbol	Id
Behavior	d	6e3501dc-5800-4c71-b59e-ad11418a998c

Influenced By

Description	Symbol	Applies To
-------------	--------	------------

Artifact Files

Content Type	File Name	File Content
Control	non-subdividable.proto	
Uml	non-subdividable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	Code 1	Daml	

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
----------	------	----------	-------------

Resource	Regulation Reference 1		
-----------------	---------------------------	--	--

Specification Behavior

Non-Subdividable

Taxonomy Symbol: ~d

An ability or restriction on the token where it cannot be subdivided from a single whole token into fractions. Sets the base token Decimals property to 0 which will make the token non-sub-dividable and a whole token is the smallest ownable unit of the token.

Example

Non-subdividable is common for items where subdivision does not make sense, like a property title, inventory item or invoice.

Analogies

Name	Description
Non-Fractional	It is not possible to own a fraction of this token.
Barrel of Oil	Barrels of Oil don't make sense to subdivide.

Is External:	True
Constructor:	

Non-Subdividable responds to these Invocations

Properties

Name: Decimals

Value Description: Set to Zero, not allowing any subdivision, usually this is applied to the base token.

Template Value: 0

Invocations

GetDecimals

Id: 2ca7fbb2-ce98-4dda-a6ae-e4ac2527bb33

Description: Should return 0

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value
------	-------

Response

Control Message: GetDecimalsResponse

Description: Return 0

Parameters

Name	Value
Decimals	0

GetDecimals

Id: 2ca7fbb2-ce98-4dda-a6ae-e4ac2527bb33

Description: Should return 0

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value
------	-------

Response

Control Message: GetDecimalsResponse

Description: Return 0

Parameters

Name	Value
Decimals	0

Properties

Unique-Transferable

Type:	Behavior
Name:	Unique-Transferable
Id:	f324a190-2f17-454a-b5b3-e1bcc50ee2d2
Visual:	<i>u</i>
Tooling:	u
Version:	1.0

Definition

The unique transferable behavior provides the owner the ability to transfer the ownership to another party or account of one or more unique tokens owned. This behavior and does not transfer the tokens themselves. Rather, new tokens are created by the transfer transaction. Because this behavior works with unique tokens, the invocation request can take multiple tokens as inputs to be transferred. The quantity of the assets being transferred to the recipients of the transaction needs to be the same quantity as the input tokens. If you do not want to transfer the entire quantity of the asset represented by the token, you can transfer a portion of the asset and the transaction will automatically make you the owner of the remaining balance. Using the example above, if only spend 50 dollars of the 100 dollar token, the transfer transaction will automatically create a new token worth 50 dollars with you as the owner. All input tokens of the transaction need to be of the same type and the tokens being transferred need to belong to the transaction initiator and are unspent.

Example

For example, if you own a token that is worth 100 dollars, you can spend 50 dollars using that token. The transfer transaction will create two new tokens as output. One token worth 50 dollars will belong to you, and another token worth 50 dollars will belong to the recipient.

Analogies

Name	Description
------	-------------

Comments

This is for unique fungible and non-fungible tokens only.

Dependencies

Artifact Type	Symbol	Description
---------------	--------	-------------

Incompatible With

Artifact Type	Symbol	Id
Behavior	~t	a4fa4ca8-6afd-452b-91f5-7103b6fee5e5
Base	tF{d}	89ca6daf-5585-469e-abd1-19bc44e7a012
Base	tN{d}	8314a797-df3c-409b-835c-0e80af92714f
Base	tF{~d}	b1eacdf8-35d8-454a-b1af-92eb0b6f45d4
Base	tN{~d}	3c05a856-c901-4c30-917e-df9feed1c8de

Influenced By

Description	Symbol	Applies To
If Compliance is present, a CheckTransferAllowed request has to be made and verified before a Transfer request or a TransferFrom request.	c	[]
If issuable is present, an AcceptTokenRequest from the token issuer, in response to a RequestTokens, has to be made and verified before a Transfer request.	i	[]

Artifact Files

Content Type	File Name	File Content
Control	uniqueTransferable.proto	
Uml	unitqueTransferable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	FabCoin Alpha	ChaincodeGo	https://github.com/hyperledger/fabric/blob/v2.0.0-alpha/token/cmd/transfer.go

Implementation Map

Map Type	Name	Platform	Location
----------	------	----------	----------

Resource Map

Map Type	Name	Location	Description
----------	------	----------	-------------

Specification Behavior

Unique-Transferable

Taxonomy Symbol: u

The unique transferable behavior provides the owner the ability to transfer the ownership to another party or account of one or more unique tokens owned. This behavior and does not transfer the tokens themselves. Rather, new tokens are created by the transfer transaction. Because this behavior works with unique tokens, the invocation request can take multiple tokens as inputs to be transferred. The quantity of the assets being transferred to the recipients of the transaction needs to be the same quantity as the input tokens. If you do not want to transfer the entire quantity of the asset represented by the token, you can

transfer a portion of the asset and the transaction will automatically make you the owner of the remaining balance. Using the example above, if only spend 50 dollars of the 100 dollar token, the transfer transaction will automatically create a new token worth 50 dollars with you as the owner. All input tokens of the transaction need to be of the same type and the tokens being transferred need to belong to the transaction initiator and are unspent.

Example

For example, if you own a token that is worth 100 dollars, you can spend 50 dollars using that token. The transfer transaction will create two new tokens as output. One token worth 50 dollars will belong to you, and another token worth 50 dollars will belong to the recipient.

Comments

This is for unique fungible and non-fungible tokens only.

Is External: True

Constructor:

Unique-Transferable responds to these Invocations

Transfer

Id: a1861aea-d2f3-47a9-bbfd-7c8ee3cf9070

Description: A transfer request will invoke a quantity transfer from tokens provided to the recipient as long as tokens being transferred need to belong to the transaction initiator and are unspent. Any remaining quantity will be returned as a new unspent quantity to the invoker/owner.

Request Message:

TransferRequest

Description: The request includes an asset quantity to transfer to a recipient and a list of tokens that have quantities that equal or exceed the quantity to transfer. Any remaining balance will be deposited into a new token with the invoker as the owner.

Request Parameters

Name	Value
TokenIds	List of Token Identifiers submitted to draw the quantity to transfer from. This data type is defined at implementation.
Quantity	Number of tokens to transfer.

Recipient	The Id to transfer ownership to.
------------------	----------------------------------

Response Message

TransferResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the owner based on the outcome of the transfer request.

Properties

Roles

Type:	Behavior
Name:	Roles
Id:	c32726da-9787-4dd8-8de3-d07d1733d0f6
Visual:	<i>r</i>
Tooling:	r
Version:	1.0

Definition

A token can have behaviors that the class will restrict invocations to a select set of parties or accounts that are members of a role or group. This is a generic behavior that can apply to a token many times to represent many role definitions within the template. This behavior will allow you to define what role(s) to create and what behavior(s) to apply the role to in the TemplateDefinition.

Example

Analogies

Name	Description
Minters	A role called 'Minters' for a token can have accounts in the role. The MintTo behavior invocation will be bound to the role check to ensure only account in the 'Minters' role are allowed to mint new instances in the class.

Comments

Roles has a constructor control that creates roles and applies them to certain behaviors of the token at creation of the class from the template.

Dependencies

Artifact Type	Symbol	Description
---------------	--------	-------------

Incompatible With

Artifact Type	Symbol	Id
---------------	--------	----

Influenced By

Description	Symbol	Applies To
-------------	--------	------------

Artifact Files

Content Type	File Name	File Content
Control	roles.proto	
Uml	roles.md	Draft

Code Map

Map Type	Name	Platform	Location
SourceCode	Code 1	Daml	

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
Resource	Regulation Reference 1		

Specification Behavior

Roles

Taxonomy Symbol: r

A token can have behaviors that the class will restrict invocations to a select set of parties or accounts that are members of a role or group. This is a generic behavior that can apply to a token many times to represent many role definitions within the template. This behavior will allow you to define what role(s) to create and what behavior(s) to apply the role to in the TemplateDefinition.

Example

Analogies

Name	Description
Minters	A role called 'Minters' for a token can have accounts in the role. The MintTo behavior invocation will be bound to the role check to ensure only account in the 'Minters' role are allowed to mint new instances in the class.

Comments

Roles has a constructor control that creates roles and applies them to certain behaviors of the token at creation of the class from the template.

Is External:	True
---------------------	------

Constructor:

Roles responds to these Invocations

RoleCheck

Id: 00a665e3-1dda-441e-8262-5750435c153c

Description: Internal invocation when the applied behavior is called to check if the requestor is a member of the role.

Request Message:

IsInRole

Description: The request

Request Parameters

Name	Value
AccountId	AccountId of the requestor.

Response Message

True/False

Description: The response

Response Parameters

Name	Value
IsInRole	True/False

Properties

Name: Role

Value Description: A group or list an account can be a member or be in.

Template Value:

Invocations

GetRoleMembers

Id:

Description: Request the the list of member accounts in the role.

Request

Control Message: GetRoleMembersRequest

Description: The request

Parameters

Name	Value
------	-------

Response

Control Message: GetRoleMembersResponse

Description: The response

Parameters

Name	Value
Members	Returning the list of accounts in the role.

AddRoleMember

Id: 600357f8-0499-47f8-87a5-eedf4ad034af

Description: Add a member to the group or role property.

Request

Control Message: AddRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if

	there is only one role.
AccountAddress	Address, name or identifier of the account to be added to the role.

Response

Control Message: AddRoleMemberResponse

Description: The response

Parameters

Name	Value
Added	True or False.

RemoveRoleMember

Id: 97e160bb-6c60-4f1d-923b-813b07b89638

Description: Remove a member to the group or role property.

Request

Control Message: RemoveRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be removed from the role.

Response

Control Message: RemoveRoleMemberResponse

Description: The response

Parameters

Name	Value
------	-------

Added	True or False.
--------------	----------------

IsInRole

Id: e42b1b16-074a-4d7d-b9f9-f69a2397a21b

Description: Check to see if an account is in the role.

Request

Control Message: IsInRoleRequest

Description: The request may be internal only and not exposed externally.

Parameters

Name	Value
RoleName	Name of the role you are checking membership of. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be checked.

Response

Control Message: IsInRoleRequestResponse

Description: The response

Parameters

Name	Value
InRole	True or False.

GetRoleMembers

Id:

Description: Request the the list of member accounts in the role.

Request

Control Message: GetRoleMembersRequest

Description: The request

Parameters

Name	Value
------	-------

Response

Control Message: GetRoleMembersResponse

Description: The response

Parameters

Name	Value
Members	Returning the list of accounts in the role.

AddRoleMember

Id: 600357f8-0499-47f8-87a5-eedf4ad034af

Description: Add a member to the group or role property.

Request

Control Message: AddRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be added to the role.

Response

Control Message: AddRoleMemberResponse

Description: The response

Parameters

Name	Value
Added	True or False.

RemoveRoleMember

Id: 97e160bb-6c60-4f1d-923b-813b07b89638

Description: Remove a member to the group or role property.

Request

Control Message: RemoveRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be removed from the role.

Response

Control Message: RemoveRoleMemberResponse

Description: The response

Parameters

Name	Value
Added	True or False.

IsInRole

Id: e42b1b16-074a-4d7d-b9f9-f69a2397a21b

Description: Check to see if an account is in the role.

Request

Control Message: IsInRoleRequest

Description: The request may be internal only and not exposed externally.

Parameters

Name	Value
RoleName	Name of the role you are checking membership of. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be checked.

Response

Control Message: IsInRoleRequestResponse

Description: The response

Parameters

Name	Value
InRole	True or False.

Properties

Redeemable

Type:	Behavior
Name:	Redeemable
Id:	51a62eba-0e84-433a-a3f1-9b3e7f72d928
Visual:	<i>e</i>
Tooling:	e
Version:	1.0

Definition

Draft

This behavior only applies to unique tokens. Redeemed tokens can no longer be spent. Redeeming a token removes an asset from the business network and guarantees that it can no longer be transferred or changed. You redeem a quantity represented in a token or tokens you own. If the redemption amount is less than the quantity represented in your token submitted, the remaining quantity after redemption is deposited into a new token and returned to you as

the owner. For example, if you have a token representing 100 dollars, and want to redeem 50, the redeem transaction will create a new token worth 50 dollars, and transfer another 50 to a restricted account without an owner.

Example

If an item in a supply chain reaches its final destination, or a financial asset reaches its term, the token representing the asset can be redeemed since the asset no longer needs to be used.

Analogies

Name	Description
Oil Barrels	If you receive a token for each barrel of oil as a refiner, you will redeem the barrel when it is refined to remove it from circulation.
Admission Ticket	A token that is a coupon or single use ticket, should be marked or torn when it is redeemed so it cannot be used again.

Dependencies

Artifact Type	Symbol	Description
---------------	--------	-------------

Incompatible With

Artifact Type	Symbol	Id
Base	tF{d}	89ca6daf-5585-469e-abd1-19bc44e7a012
Base	tN{d}	8314a797-df3c-409b-835c-0e80af92714f
Base	tF{~d}	b1eacdf8-35d8-454a-b1af-92eb0b6f45d4
Base	tN{~d}	3c05a856-c901-4c30-917e-df9feed1c8de

Influenced By

Description	Symbol	Applies To
If Compliance is present, a CheckBurnAllowed request has to be made and verified before a Burn request or a BurnFrom request.	c	[]

Artifact Files

Content Type	File Name	File Content
Control	redeemable.proto	
Uml	burnable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	FabToken	EthereumSolidity	https://github.com/hyperledger/fabric/blob/v2.0.0-alpha/token/cmd/redeem.go

Implementation Map

Map Type	Name	Platform	Location
----------	------	----------	----------

Resource Map

Map Type	Name	Location	Description
----------	------	----------	-------------

Specification Behavior

Redeemable

Taxonomy Symbol: e

This behavior only applies to unique tokens. Redeemed tokens can no longer be spent. Redeeming a token removes an asset from the business network and guarantees that it can no longer be transferred or changed. You redeem a quantity represented in a token or tokens you own. If the redemption amount is less than the quantity represented in your token submitted, the remaining quantity after redemption is deposited into a new token and returned to you as the owner. For example, if you have a token representing 100 dollars, and want

to redeem 50, the redeem transaction will create a new token worth 50 dollars, and transfer another 50 to a restricted account without an owner.

Example

If an item in a supply chain reaches its final destination, or a financial asset reaches its term, the token representing the asset can be redeemed since the asset no longer needs to be used.

Analogies

Name	Description
Oil Barrels	If you receive a token for each barrel of oil as a refiner, you will redeem the barrel when it is refined to remove it from circulation.
Admission Ticket	A token that is a coupon or single use ticket, should be marked or torn when it is redeemed so it cannot be used again.

Is External:	True
Constructor:	

Redeemable responds to these Invocations

Redeem

Id: c6a0ef93-0d7d-4c68-a3b4-8f5d0bbadcbe

Description: A redeem request will redeem a quantity transfer from tokens provided to an account that has no owner, removing them permanently from circulation, as long as tokens being redeemed need to belong to the transaction initiator and are unspent. Any remaining quantity will be returned as a new unspent quantity to the invoker/owner.

Request Message:

RedeemRequest

Description: The request includes an asset quantity to redeem and a list of tokens that have quantities that equal or exceed the quantity to redeem. Any remaining balance will be deposited into a new token with the invoker as the owner.

Request Parameters

Name	Value
TokenIds	List of Token Identifiers submitted to draw the quantity to redeem

	from. This data type is defined at implementation.
Quantity	Number of tokens to transfer.
Recipient	Always null, empty or to a know account that has no owner.

Response Message

RedeemResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the invoker based on the outcome of the fabricate request.

Properties

Fabricate

Type:	Behavior
Name:	Fabricate
Id:	76c6ebb0-8ab8-457b-b6e3-29c38102ed22
Visual:	<i>f</i>
Tooling:	f
Version:	1.0

Definition

Unique tokens can be fabricated from a base tokenization capability present in the platform being used. Similar to minting or issuing of new token instances, tokens are given a type, which can be a simple string, and a quantity. The result will be a single token of some quantity of the type. You can fabricate multiple tokens in the same issue request to different new owners of the fabricated tokens.

Example

Fabricated tokens have a type and quantity, like a check or IOU, a single token can represent a distinct quantity of a value identified by its type.

Analogies

Name	Description
Writing a Check	Like writing a check, the amount of the check is like the quantity on a token and the denomination is like the type of token.

Dependencies

Artifact Type	Symbol	Description
---------------	--------	-------------

Incompatible With

Artifact Type	Symbol	Id
Base	tF{d}	89ca6daf-5585-469e-abd1-19bc44e7a012
Base	tN{d}	8314a797-df3c-409b-835c-0e80af92714f
Base	tF{~d}	b1eacdf8-35d8-454a-b1af-92eb0b6f45d4
Base	tN{~d}	3c05a856-c901-4c30-917e-df9feed1c8de

Influenced By

Description	Symbol	Applies To
Roles is common to implement to provide authorization checks for invoking the behavior.	r	[]
If Compliance is present, a CheckMintAllowed request has to be made and verified before a Mint request or a MintTo request.	c	[]

Artifact Files

Content Type	File Name	File Content
--------------	-----------	--------------

Control	fabricate.proto	
Uml	fabricate.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	FabToken Alpha	ChaincodeGo	https://github.com/hyperledger/fabric/blob/v2.0.0-alpha/token/cmd/issue.go

Implementation Map

Map Type	Name	Platform	Location
----------	------	----------	----------

Resource Map

Map Type	Name	Location	Description
----------	------	----------	-------------

Specification Behavior

Fabricate

Taxonomy Symbol: f

Unique tokens can be fabricated from a base tokenization capability present in the platform being used. Similar to minting or issuing of new token instances, tokens are given a type, which can be a simple string, and a quantity. The result will be a single token of some quantity of the type. You can fabricate multiple tokens in the same issue request to different new owners of the fabricated tokens.

Example

Fabricated tokens have a type and quantity, like a check or IOU, a single token can represent a distinct quantity of a value identified by its type.

Analogies

Name	Description
Writing a Check	Like writing a check, the amount of the check is like the quantity on a token and the denomination is like the type of token.

Is External:	True
Constructor:	

Fabricate responds to these Invocations

Binding Is Influenced by Roles's Invocation RoleCheckRoles's Invocation RoleCheck Intercepts this behavior's invocation.'

RoleCheck

Id: 00a665e3-1dda-441e-8262-5750435c153c

Description: Check to see if the account is in the Role called 'Issuers'

Request Message:

IsInRole

Description: Checking the 'Issuers' role.

Request Parameters

Name	Value
AccountId	AccountId of the requestor.

Response Message

True/False

Description: Respond true if the account is in the 'Issuers' role.

Response Parameters

Name	Value
IsInRole	True/False

Fabricate

Id: 39181196-12e0-4fd4-b8fa-4e965a6fef66

Description: A request to create new token instances in the class by the owner or a party or account in a role that is granted this permission. Can contain multiple issues to more than one new owner.

Request Message:

FabricateRequest

Description: A list of Issue properties defining the new token Owner, Token Type and Quantity.

Request Parameters

Name	Value
Issues	A list of Issue properties with values for the new Token Owner, Type and Quantity.

Response Message

FabricateResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the invoker based on the outcome of the fabricate request.

Properties

Name: Issue

Value Description: Values for Owner, Type and Quantity for issuing a Fabricated token. This can be repeated

Template Value:

Invocations

Properties

Name: Owner Id

Value Description: identifies the owner that can transfer or redeem the new token.

Template Value:

Invocations

Properties

Name: TokenType

Value Description: Name or type name of the asset that the token represents.

Template Value:

Invocations

Properties

Name: Quantity

Value Description: Quantity of the type the token represents.

Template Value:

Invocations

Properties

