
STUDENTSKILLS

Contributors

Taxonomy Formula: $tN\{\sim d, \sim t, SC\}$

Token Specification Summary

Token Classification

This is a Whole Non-Transferable and Non-Fungible Token that uses the behavior group Supply Control to manage the circulating supply, so at creation can be set an initial supply that can be increased or removed as needed. As it is non-fungible, it is meant to be unique for the student. It is whole by setting the decimals property on the dividable behavior to 0, that way it cannot be fractioned. It's Non-Transferable, which means that the owner cannot transfer the ownership of the tokens to another party.

Example

This token definition's purpose is to register and certify skills acquisition by a student after completing a course successfully. It will represent a certificate of completion of a course, while it will persist on the blockchain forever. Every instance of token definition minted will be stored on the student wallet if a course is completed, this wallet will be used as the student profile to be shown with every learning token to potential employers, organizations, and institutions.

Analogies

Earned Skill Credits/Certificates. A student can earn a non-fungible point/token for each course completed to build up a profile for potential employers.

StudentSkills is:

- Indivisible
- Non-Transferable
- Mintable
- Burnable
- Roles

StudentSkills Details

Whole Non-Fungible

Type:	Base
Whole Non-Fungible	
Id:	3c05a856-c901-4c30-917e-df9feed1c8de
&tau_N{<i>~d</i>}	
Tooling:	tN{~d}
1.0	

Definition

Every non-fungible token is unique. A non-fungible token is not interchangeable with other tokens of the same class but have some shared properties while also having unique property values between instances. These tokens are whole tokens and can have quantities greater than 1 and also could support variable supply.

Example

CryptoKitties, Art, Reserved Seat for an event.

Analogies

Name:	Description
Property Title	The physical property title, land for example, have the identical look and feel from the paper, colors and seal. The difference between them are the values like property address, plot numbers, etc. These values make the title unique. There are some properties on a class of titles that are the same, like the county or jurisdiction the property is in. For titles that have some shared values and unique values, it may make more sense to define them in the same class.
Art	The valuable painting or other unique piece of art may not share any property values with other paintings, unless the artist is extremely prolific in generating tens of thousands of pieces of art, it would make sense to define each piece of art as its own class. Meaning there would be only a single piece of art represented by the token class. If the art cannot be sub-divided, meaning there can be no fractional owners, this token class can be a singleton if the quantity in the class is set to 1. A singleton has only one instance in the class, essentially meaning the class is the instance, and not be sub-dividable and no new tokens can be minted in the class.

Dependencies

Artifact Type	Symbol	Description
Base	t	Base Token Definition

Incompatible With

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

Influenced By

Artifact Files

Code Map

Implementation Map

Resource Map

Base Details

Token Name:	
NonFungible	
Representation Type:	Common
Intrinsic	
Token Unit:	Whole
CET	
Owner:	
0	
Decimals:	0
Constructor	

Behaviors

Indivisible

Type:	Behavior
Indivisible	
Id:	d5807a8e-879b-4885-95fa-f09ba2a22172
<i>d</i>	
Tooling:	~d
1.0	

Definition

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

Example

Indivisible is common for items where division does not make sense, like a property title, inventory item or invoice.

Non-Transferable

Type:	Behavior
Non-Transferable	
Id:	a4fa4ca8-6afd-452b-91f5-7103b6fee5e5
<i>~t</i>	
Tooling:	~t
1.0	

Definition

Every token instance has an owner. The Non-transferable behavior prevents the owner of a token from changing.

Example

A vote token, for a citizen in a public election would be non-transferable.

Analogies

Name:	Description
Diploma	A diploma from an educational institution is not transferable to another party that can claim to have earned the diploma.
Airline Ticket	Due to security restrictions at airports and airlines, tickets can only be used by the person they were issued to.

Dependencies

Incompatible With

Artifact Type	Symbol	Id
Behavior	t	af119e58-6d84-4ca6-9656-75e8d312f038

Influenced By

Artifact Files

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

Code Map

Implementation Map

Resource Map

Specification Behavior

Mintable

Type:	Behavior
Mintable	
Id:	f9224e90-3cab-45bf-b5dc-0175121e2ead

<i>m</i>

Tooling: m

1.0

Definition

A token class that implements this behavior will support the minting or issuing of new token instances in the class. These new tokens can be minted and belong to the owner or minted to another account. This behavior may be invalidated by a restrictive behavior like Singleton, where only a single instance of the token can exist. Mintable is technically delegable, but it's delegation should be controlled by a behavior like Roles.

Example

A consortium of oil producers needs to create tokens for each barrel of oil they are putting on the market to trade. There are separate classes of tokens for each grade of oil. Producers of barrels will need to have the ability to mint new tokens in order to facilitate the trading of them in the supply chain.

Analogies

Name:	Description
SKU	A token class can represent a particular item SKU, where the manufacturer of the item has the ability to mint or issue new inventory of the SKU into the supply chain.

Dependencies

Incompatible With

Influenced By

Description	Symbol	Applies To
Roles is common to implement to provide authorization checks for invoking the behavior. Highly Recommended that Role restrictions be applied to MintTo invocations.	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	mintable.proto	
Uml	mintable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	Openzeppelin	EthereumSolidity	https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Mintable.sol

Implementation Map

Resource Map

Specification Behavior

Burnable

Type:	Behavior
Burnable	
Id:	803297a1-c0f9-4898-9d44-29c9d41cca97
<i>b</i>	
Tooling:	b
1.0	

Definition

A token class that implements this behavior will support the burning or decommissioning of token instances of the class. This does not delete a token, but rather places it in a permanent non-use state. Burning is a one way operation and cannot be reversed. This behavior is Delegation. If the token definition is Delegation, BurnFrom will be available.

Example

When a token is used in a certain way, you may want to remove it from circulation or from being used again. Since the ledger doesn't allow for deletions, burning a token essentially 'deletes' the token from being used, but not from history.

Analogies

Name:	Description
Oil Barrels	If you mint a new token for each barrel of oil created, you may transfer ownership several times until the barrel is refined. The refining process should burn the barrel of oil to remove it from circulation.
Redeem	A token that is a coupon or single use ticket, should be burned when it is redeemed.

Dependencies

Incompatible With

Influenced By

Description	Symbol	Applies To
Delegable or not, will determine if the BurnFrom Control will be available in the implementation.	g	[]
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	burnable.proto	
Uml	burnable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	Openzeppelin	EthereumSolidity	https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Burnable.sol

Implementation Map

Resource Map

Specification Behavior

Roles

Type:	Behavior
Roles	
Id:	c32726da-9787-4dd8-8de3-d07d1733d0f6
<i>r</i>	
Tooling:	r
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.

Dependencies

Incompatible With

Influenced By

Artifact Files

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

Code Map

Implementation Map

Resource Map

Specification Behavior