

---

# STUDENTGRADE

---

## Contributors

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

## Token Specification Summary

### Token Classification

*This is a Transferable Whole 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. It is whole by setting the decimals property on the dividable behavior to 0, that way it cannot be fractioned. It's Transferable, which means that the owner can transfer the ownership of the tokens to another party.*

### Example

This token definition's purpose is to register and reflect the grade obtained by a student after completing a course. This will help to determine how successful was the course for the student, and also a way to register student's final grade of the course, that way if obtained a satisfactory grade the student will receive a StudentSkills non-fungible token as certificate.

### Analogies

A student can earn a point/token based on the grade obtained after completing each test/assignment of the course, and then redeem a certificate (NFT) with these points/tokens once the course is completed.

### StudentGrade is:

- Indivisible
- Transferable
- Delegable
- Mintable
- Burnable
- Roles

# StudentGrade Details

## Whole Fungible

Type:	Base
Whole Fungible	
Id:	b1eacdf8-35d8-454a-b1af-92eb0b6f45d4
&tau;<sub>F</sub>{<i>~d</i>}	
Tooling:	tF{~d}
1.0	

## Definition

*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'.*

## 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	Purchasing a general admission ticket to a movie only allows for you to have a seat, but the seat that you actually get depends on actors like when you arrive.
Movie Ticket	Your not likely to want to share a seat with another adult.

## Definition

## 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

Artifact Files

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

Code Map

Implementation Map

Resource Map

Base Details

Token Name:	
Fungible	
Representation Type:	Common
Intrinsic	
Token Unit:	Whole
CET	
Owner:	
0	
Decimals:	8
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.

### Transferable

Type:	Behavior
Transferable	
Id:	af119e58-6d84-4ca6-9656-75e8d312f038
<i>t</i>	
Tooling:	t
1.0	

### Definition

*Every token instance has an owner. The Transferable behavior provides the owner the ability to transfer the ownership to another party or account. This behavior is often inferred by other behaviors that might exist like Redeem, Sell, etc. This behavior is Delegable. If the token definition is Delegable, TransferFrom will be available.*

## Analogies

## Dependencies

## Incompatible With

Artifact Type	Symbol	Id
Behavior	~t	a4fa4ca8-6afd-452b-91f5-7103b6fee5e5

## Influenced By

Description	Symbol	Applies To
If the token is Delegable, TransferFrom should be enabled.	g	[ ]
If Compliance is present, a CheckTransferAllowed request has to be made and verified before a Transfer request or a TransferFrom request.	c	[ ]

## Artifact Files

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

## Code Map

## Implementation Map

## Resource Map

## Specification Behavior

## Delegable

Type:	Behavior
Delegable	
Id:	a3d02076-6009-4a65-9ed4-2deffe5291e1
<i>g</i>	
Tooling:	g

## Definition

*A token class that implements this behavior will support the delegation of certain behaviors to another party or account to invoke them on the behalf of the owner. When applied to a token, behaviors that are Delegable will enable delegated request invocations. This is useful to provide another party to automatically be able to perform the behaviors that can be delegated without seeking permission up to a certain allowance.*

## Analogies

Name:	Description
<b>Broker</b>	You may allow a broker to transfer your tokens as a part of an investment strategy. Setting an allowance can cap the total number of tokens the broker is allowed to perform delegated behaviors, when exceeded a new allowance request will need to be granted.

## Dependencies

## Incompatible With

## Influenced By

## Artifact Files

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

## Code Map

## Implementation Map

## Resource Map

## Specification Behavior

## Mintable

Type:	Behavior
<b>Mintable</b>	
<b>Id:</b>	f9224e90-3cab-45bf-b5dc-0175121e2ead
<i>m</i>	
<b>Tooling:</b>	m
<b>1.0</b>	

## 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
<b>SKU</b>	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
<b>Roles is common to implement to provide authorization checks for r invoking the behavior. Highly Recommended that Role restrictions be applied to MintTo invocations.</b>		[ ]
<b>If Compliance is present, a CheckMintAllowed request has to be made c and verified before a Mint request or a MintTo request.</b>		[ ]

## Artifact Files

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

## Code Map

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

## 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 Delegable. If the token definition is Delegable, 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
<b>Oil Barrels</b>	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.
<b>Redeem</b>	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
<b>Delegable or not, will determine if the BurnFrom Control will be available in the implementation.</b>	<b>g</b>	[ ]
<b>If Compliance is present, a CheckBurnAllowed request has to be made and verified before a Burn request or a BurnFrom request.</b>	<b>c</b>	[ ]

## Artifact Files

Content Type	File Name	File Content
<b>Control</b>	<b>burnable.proto</b>	
<b>Uml</b>	<b>burnable.md</b>	

## Code Map

Map Type	Name	Platform	Location
<b>SourceCode</b>	<b>Openzeppelin</b>	<b>EthereumSolidity</b>	<a href="https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Burnable.sol">https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Burnable.sol</a>

## 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