

Spending

Because Token types do not have the store ability, it is impossible to store them in another object. Hence, Coin-like approaches to spending are not possible - an application that takes Token as a payment won't be able to add it to its balance. To address this issue, Token has a spend method, which allows spending it in one application and then delivering it as a spent_balance to the [TokenPolicy](#) or burning right away with a TreasuryCap .

Tokens can be spent by calling the spend method. It takes the following arguments:

As the signature shows, the Token object is consumed. Its balance becomes the spent_balance in the [ActionRequest](#) .

The ActionRequest for the spend action contains the Balance of the spent Token , and it can either be confirmed [with a](#) TreasuryCap or [delivered to the](#) TokenPolicy . In the first case, the balance is burned directly in the TreasuryCap , and in the second case, it's delivered to the TokenPolicy spent_balance .

Spent balance cannot be used in any way, and it is not possible to withdraw it. The only available action is [flushing](#) - burning the spent_balance by bringing a TreasuryCap .

Normally, the spend action should have at least one rule assigned to it to prevent aimless spending, and the recommended way of authorizing the spend in an application that accepts the token is to stamp it right in the function where a spend is performed. For example:

Spend action

Tokens can be spent by calling the spend method. It takes the following arguments:

As the signature shows, the Token object is consumed. Its balance becomes the spent_balance in the [ActionRequest](#) .

The ActionRequest for the spend action contains the Balance of the spent Token , and it can either be confirmed [with a](#) TreasuryCap or [delivered to the](#) TokenPolicy . In the first case, the balance is burned directly in the TreasuryCap , and in the second case, it's delivered to the TokenPolicy spent_balance .

Spent balance cannot be used in any way, and it is not possible to withdraw it. The only available action is [flushing](#) - burning the spent_balance by bringing a TreasuryCap .

Normally, the spend action should have at least one rule assigned to it to prevent aimless spending, and the recommended way of authorizing the spend in an application that accepts the token is to stamp it right in the function where a spend is performed. For example:

Spent token

The ActionRequest for the spend action contains the Balance of the spent Token , and it can either be confirmed [with a](#) TreasuryCap or [delivered to the](#) TokenPolicy . In the first case, the balance is burned directly in the TreasuryCap , and in the second case, it's delivered to the TokenPolicy spent_balance .

Spent balance cannot be used in any way, and it is not possible to withdraw it. The only available action is [flushing](#) - burning the spent_balance by bringing a TreasuryCap .

Normally, the spend action should have at least one rule assigned to it to prevent aimless spending, and the recommended way of authorizing the spend in an application that accepts the token is to stamp it right in the function where a spend is performed. For example:

Gating the spend action

Normally, the spend action should have at least one rule assigned to it to prevent aimless spending, and the recommended way of authorizing the spend in an application that accepts the token is to stamp it right in the function where a spend is performed. For example: