Module sui::token

The Token module which implements a Closed Loop Token with a configurable policy. The policy is defined by a set of rules that must be satisfied for an action to be performed on the token.

The module is designed to be used with a TreasuryCap to allow for minting and burning of the Token s. And can act as a replacement / extension or a companion to existing open-loop (Coin) systems.

The Token system allows for fine-grained control over the actions performed on the token. And hence it is highly suitable for applications that require control over the currency which a simple open-loop system can't provide.

A single Token with Balance inside. Can only be owned by an address, and actions performed on it must be confirmed in a matching TokenPolicy .

A Capability that manages a single TokenPolicy specified in the for field. Created together with TokenPolicy in the new function.

TokenPolicy represents a set of rules that define what actions can be performed on a Token and which Rules must be satisfied for the action to succeed.

A request to perform an "Action" on a token. Stores the information about the action to be performed and must be consumed by the confirm_request or confirm_request_mut functions when the Rules are satisfied.

Dynamic field key for the TokenPolicy to store the Config for a specific action Rule. There can be only one configuration per Rule per TokenPolicy .

An event emitted when a TokenPolicy is created and shared. Because TokenPolicy can only be shared (and potentially frozen in the future), we emit this event in the share_policy function and mark it as mutable.

The action is not allowed (defined) in the policy.

The rule was not approved.

Trying to perform an admin action with a wrong cap.

The balance is too low to perform the action.

The balance is not zero.

The balance is not zero when trying to confirm with TransferPolicyCap.

Rule is trying to access a missing config (with type).

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

A Tag for the spend action.

A Tag for the transfer action.

A Tag for the to_coin action.

A Tag for the from_coin action.

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy .

spent_balance .

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

Join two Token s into one, always available.

Split a Token with amount . Aborts if the Token . balance is lower than amount .

Create a zero Token .

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

Transfer the Token to the transaction sender.

Create a new ActionRequest . Publicly available method to allow for custom actions.

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

Burn a [Token](#) using the TreasuryCap.

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

Check whether an action is present in the rules VecMap.

Returns the rules required for a specific action.

Returns the [spent_balance](#) of the [TokenPolicy](#) .

Returns the [balance](#) of the [Token](#) .

Name of the Transfer action.

Name of the Spend action.

Name of the ToCoin action.

Name of the FromCoin action.

The Action in the [ActionRequest](#) .

Amount of the [ActionRequest](#) .

Sender of the [ActionRequest](#) .

Recipient of the [ActionRequest](#) .

Approvals of the [ActionRequest](#) .

Burned balance of the [ActionRequest](#) .

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

Struct

A single [Token](#) with Balance inside. Can only be owned by an address, and actions performed on it must be confirmed in a matching [TokenPolicy](#) .

```bash

```
```

A Capability that manages a single TokenPolicy specified in the for field. Created together with TokenPolicy in the new function.

```bash
```

TokenPolicy represents a set of rules that define what actions can be performed on a Token and which Rules must be satisfied for the action to succeed.

```bash
```

A request to perform an "Action" on a token. Stores the information about the action to be performed and must be consumed by the confirm_request or confirm_request_mut functions when the Rules are satisfied.

```bash
```

Dynamic field key for the TokenPolicy to store the Config for a specific action Rule. There can be only one configuration per Rule per TokenPolicy .

```bash
```

An event emitted when a TokenPolicy is created and shared. Because TokenPolicy can only be shared (and potentially frozen in the future), we emit this event in the share_policy function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

```bash
```

```
```

```bash
```

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```bash
```

Split a Token with amount . Aborts if the Token . balance is lower than amount .

```bash
```

```bash
```

Create a zero Token .

```bash
```

```bash
```

```

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

```bash
```

```bash
```

Transfer the Token to the transaction sender.

```bash
```

```bash
```

Create a new ActionRequest . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash

```
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash

```
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Struct

A Capability that manages a single TokenPolicy specified in the for field. Created together with TokenPolicy in the new function.

```bash
```

TokenPolicy represents a set of rules that define what actions can be performed on a Token and which Rules must be satisfied for the action to succeed.

```bash
```

A request to perform an "Action" on a token. Stores the information about the action to be performed and must be consumed by the confirm_request or confirm_request_mut functions when the Rules are satisfied.

```bash
```

Dynamic field key for the TokenPolicy to store the Config for a specific action Rule. There can be only one configuration per Rule per TokenPolicy .

```bash
```

An event emitted when a TokenPolicy is created and shared. Because TokenPolicy can only be shared (and potentially frozen in the future), we emit this event in the share_policy function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

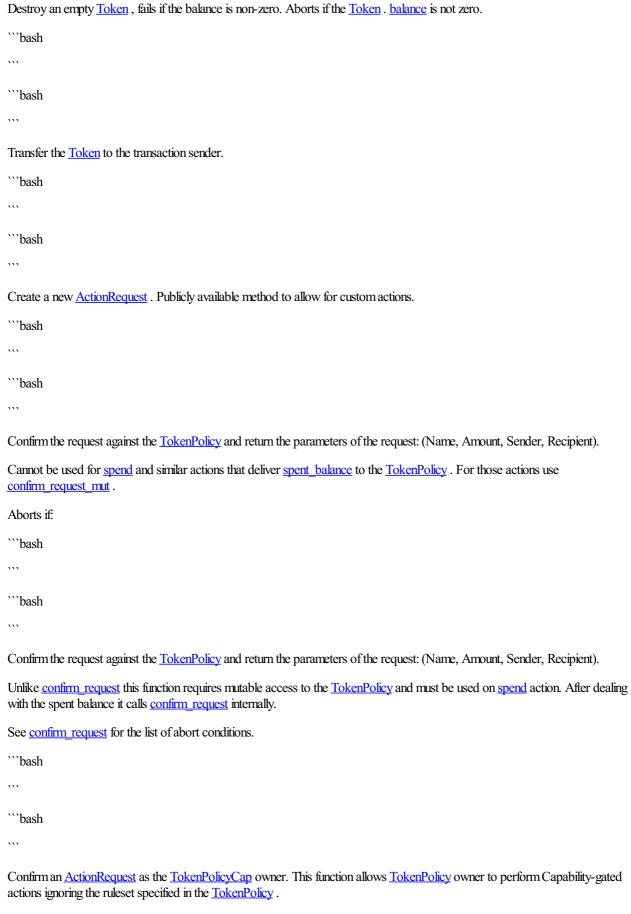The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

```
```

Rule is trying to access a missing config (with type).

```bash
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

```bash
```

```bash
```

Spend a [Token](#) by unwrapping it and storing the Balance in the [ActionRequest](#) for the "spend" action. The [ActionRequest](#) contains the [spent_balance](#) field to be used in verification.

Spend action requires [confirm_request_mut](#) to be called to confirm the request and join the spent balance with the [TokenPolicy](#) . [spent_balance](#) .

```bash
```

```bash
```

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a [Token](#) . Creates an [ActionRequest](#) for the "from_coin" action.

```bash
```

```bash
```

Join two [Token](#) s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Struct

[TokenPolicy] represents a set of rules that define what actions can be performed on a [Token] and which Rules must be satisfied for

the action to succeed.

```bash
```

A request to perform an "Action" on a token. Stores the information about the action to be performed and must be consumed by the confirm_request or confirm_request_mut functions when the Rules are satisfied.

```bash
```

Dynamic field key for the TokenPolicy to store the Config for a specific action Rule. There can be only one configuration per Rule per TokenPolicy .

```bash
```

An event emitted when a TokenPolicy is created and shared. Because TokenPolicy can only be shared (and potentially frozen in the future), we emit this event in the share_policy function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

```
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

```bash
```

```bash
```

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```bash
```

Split a Token with amount . Aborts if the Token . balance is lower than amount .

```bash
```

```bash
```

Create a zero Token .

```bash
```

```bash
```

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

```
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash
```

Amount of the ActionRequest .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Struct

A request to perform an "Action" on a token. Stores the information about the action to be performed and must be consumed by the [confirm_request] or [confirm_request_mut] functions when the Rules are satisfied.

```bash
```

Dynamic field key for the [TokenPolicy](#) to store the Config for a specific action Rule. There can be only one configuration per Rule per [TokenPolicy](#) .

```bash
```

An event emitted when a [TokenPolicy](#) is created and shared. Because [TokenPolicy](#) can only be shared (and potentially frozen in the future), we emit this event in the [share_policy](#) function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

Using [confirm_request_mut](#) without [spent_balance](#) . Immutable version of the function must be used instead.

```bash
```

A Tag for the [spend](#) action.

```bash
```

A Tag for the [transfer](#) action.

```bash
```

A Tag for the [to_coin](#) action.

```bash
```

A Tag for the [from_coin](#) action.

```bash
```

Create a new [TokenPolicy](#) and a matching [TokenPolicyCap](#) . The [TokenPolicy](#) must then be shared using the [share_policy](#) method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the [TokenPolicy](#) . Due to [key](#) -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a [Token](#) to a [recipient](#) . Creates an [ActionRequest](#) for the "transfer" action. The [ActionRequest](#) contains the [recipient](#) field to be used in verification.

```bash
```

```bash
```

Spend a [Token](#) by unwrapping it and storing the Balance in the [ActionRequest](#) for the "spend" action. The [ActionRequest](#) contains the [spent_balance](#) field to be used in verification.

Spend action requires [confirm_request_mut](#) to be called to confirm the request and join the spent balance with the [TokenPolicy](#) . [spent_balance](#) .

```bash
```

```bash
```

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a [Token](). Creates an [ActionRequest]() for the "from_coin" action.

```bash
```

```bash
```

Join two [Token]()s into one, always available.

```bash
```

```bash
```

Split a [Token]() with [amount](). Aborts if the [Token](). [balance]() is lower than [amount]().

```bash
```

```bash
```

Create a zero [Token]().

```bash
```

```bash
```

Destroy an empty [Token](), fails if the balance is non-zero. Aborts if the [Token](). [balance]() is not zero.

```bash
```

```bash
```

Transfer the [Token]() to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```

```bash

```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash

```

```bash

```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash

```
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Struct

Dynamic field key for the [TokenPolicy] to store the Config for a specific action Rule. There can be only one configuration per Rule per [TokenPolicy] .

```bash
```

An event emitted when a [TokenPolicy] is created and shared. Because [TokenPolicy] can only be shared (and potentially frozen in the future), we emit this event in the [share_policy] function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

```bash
```

```bash
```

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](). Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](). Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap]() to guarantee that the [TokenPolicy]() owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](). Unlike the [add_rule_config](), this function does not require a Rule witness, hence can be performed by the [TokenPolicy]() owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest]().

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy]() without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy]() and that it matches the type provided.

```bash
```

```bash
```

Allows an [action]() to be performed on the [Token]() freely by adding an empty set of Rules for the [action]().

Aborts if the [TokenPolicyCap]() is not matching the [TokenPolicy]().

```bash

```
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest] .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Struct

An event emitted when a [TokenPolicy](#) is created and shared. Because [TokenPolicy](#) can only be shared (and potentially frozen in the future), we emit this event in the [share_policy](#) function and mark it as mutable.

```bash
```

The action is not allowed (defined) in the policy.

```bash
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new TokenPolicy and a matching TokenPolicyCap . The TokenPolicy must then be shared using the share_policy method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the TokenPolicy . Due to key -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a Token to a recipient . Creates an ActionRequest for the "transfer" action. The ActionRequest contains the recipient field to be used in verification.

```bash
```

```bash

```

```

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```bash
```

Split a Token with amount . Aborts if the Token . balance is lower than amount .

```bash
```

```bash
```

Create a zero Token .

```bash
```

```bash
```

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash

```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash

```

```bash

```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash

```

```bash

```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash

```

```bash

```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash

```

```bash

```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash

```

```bash

```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence

can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

```

```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash

```

```bash

```

Burn a [Token](#) using the TreasuryCap.

```bash

```

```bash

```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash

```

```bash

```

Check whether an action is present in the rules VecMap.

```bash

```

```bash

```

Returns the rules required for a specific action.

```bash

```

```bash

```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash

```

```
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Constants

The action is not allowed (defined) in the policy.

```bash
```

```
```

The rule was not approved.

```bash
```

Trying to perform an admin action with a wrong cap.

```bash
```

The balance is too low to perform the action.

```bash
```

The balance is not zero.

```bash
```

The balance is not zero when trying to confirm with TransferPolicyCap.

```bash
```

Rule is trying to access a missing config (with type).

```bash
```

Using confirm_request_mut without spent_balance . Immutable version of the function must be used instead.

```bash
```

A Tag for the spend action.

```bash
```

A Tag for the transfer action.

```bash
```

A Tag for the to_coin action.

```bash
```

A Tag for the from_coin action.

```bash
```

Create a new [TokenPolicy](#) and a matching [TokenPolicyCap](#) . The [TokenPolicy](#) must then be shared using the [share_policy](#) method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the [TokenPolicy](#) . Due to [key](#) -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a [Token](#) to a [recipient](#) . Creates an [ActionRequest](#) for the "transfer" action. The [ActionRequest](#) contains the [recipient](#) field to be used in verification.

```bash
```

```bash
```

Spend a [Token](#) by unwrapping it and storing the Balance in the [ActionRequest](#) for the "spend" action. The [ActionRequest](#) contains the [spent_balance](#) field to be used in verification.

Spend action requires [confirm_request_mut](#) to be called to confirm the request and join the spent balance with the [TokenPolicy](#) . [spent_balance](#) .

```bash
```

```bash
```

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a [Token](#) . Creates an [ActionRequest](#) for the "from_coin" action.

```bash
```

```bash
```

Join two [Token](#)s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy] . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy] . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap] to guarantee that the [TokenPolicy] owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy] . Unlike the [add_rule_config] , this function does not require a Rule witness, hence can be performed by the [TokenPolicy] owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest] .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy] without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy] and that it matches the type provided.

```bash
```

```bash
```

Allows an [action] to be performed on the [Token] freely by adding an empty set of Rules for the [action] .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

```
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Create a new [TokenPolicy] and a matching [TokenPolicyCap] . The [TokenPolicy] must then be shared using the [share_policy] method.

TreasuryCap guarantees full ownership over the currency, and is unique, hence it is safe to use it for authorization.

```bash
```

```bash
```

Share the [TokenPolicy] . Due to [key] -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a [Token] to a [recipient] . Creates an [ActionRequest] for the "transfer" action. The [ActionRequest] contains the [recipient] field to be used in verification.

```bash
```

```bash
```

Spend a [Token] by unwrapping it and storing the Balance in the [ActionRequest] for the "spend" action. The [ActionRequest] contains the [spent_balance] field to be used in verification.

Spend action requires [confirm_request_mut] to be called to confirm the request and join the spent balance with the [TokenPolicy] . [spent_balance] .

```bash
```

```bash
```

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a [Token](#) . Creates an [ActionRequest](#) for the "from_coin" action.

```bash
```

```bash
```

Join two [Token](#) s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the Token to the transaction sender.

```bash
```

```bash
```

Create a new ActionRequest . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Share the [TokenPolicy] . Due to [key] -only restriction, it must be shared after initialization.

```bash
```

```bash
```

Transfer a [Token] to a [recipient] . Creates an [ActionRequest] for the "transfer" action. The [ActionRequest] contains the [recipient] field

to be used in verification.

```bash
```

```bash
```

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```bash
```

Split a Token with amount . Aborts if the Token . balance is lower than amount .

```bash
```

```bash
```

Create a zero Token .

```bash
```

```

```bash
```

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

```bash
```

```bash
```

Transfer the Token to the transaction sender.

```bash
```

```bash
```

Create a new ActionRequest . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated

actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash

```

Function

Transfer a [Token](#) to a [recipient](#) . Creates an [ActionRequest](#) for the "transfer" action. The [ActionRequest](#) contains the [recipient](#) field to be used in verification.

```bash

```

```bash

```

Spend a [Token](#) by unwrapping it and storing the Balance in the [ActionRequest](#) for the "spend" action. The [ActionRequest](#) contains the [spent_balance](#) field to be used in verification.

Spend action requires [confirm_request_mut](#) to be called to confirm the request and join the spent balance with the [TokenPolicy](#) . [spent_balance](#) .

```bash

```

```bash

```

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash

```

```bash

```

Convert an open Coin into a [Token](#) . Creates an [ActionRequest](#) for the "from_coin" action.

```bash

```

```bash

```

Join two [Token](#) s into one, always available.

```bash

```

```bash

```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash

```

```bash
```

Create a zero Token .

```bash
```

```bash
```

Destroy an empty Token , fails if the balance is non-zero. Aborts if the Token . balance is not zero.

```bash
```

```bash
```

Transfer the Token to the transaction sender.

```bash
```

```bash
```

Create a new ActionRequest . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash

```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash

```

```bash

```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash

```

```bash

```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash

```

```bash

```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash

```

```bash

```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash

```

```bash

```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash

```
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

```
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Spend a Token by unwrapping it and storing the Balance in the ActionRequest for the "spend" action. The ActionRequest contains the spent_balance field to be used in verification.

Spend action requires confirm_request_mut to be called to confirm the request and join the spent balance with the TokenPolicy . spent_balance .

```bash
```

```bash
```

Convert Token into an open Coin. Creates an ActionRequest for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a Token . Creates an ActionRequest for the "from_coin" action.

```bash
```

```bash
```

Join two Token s into one, always available.

```bash
```

```bash
```

Split a Token with amount . Aborts if the Token . balance is lower than amount .

```bash
```

```bash
```

Create a zero [Token] .

```bash
```

```bash
```

Destroy an empty [Token] , fails if the balance is non-zero. Aborts if the [Token] . [balance] is not zero.

```bash
```

```bash
```

Transfer the [Token] to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest] . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy] and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend] and similar actions that deliver [spent_balance] to the [TokenPolicy] . For those actions use [confirm_request_mut] .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy] and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request] this function requires mutable access to the [TokenPolicy] and must be used on [spend] action. After dealing with the spent balance it calls [confirm_request] internally.

See [confirm_request] for the list of abort conditions.

```bash
```

```bash
```

```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap ).

```bash

```

```bash

```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash

```

```bash

```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash

```

```bash

```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash

```

```bash

```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash

```

```bash

```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

```
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```
```

```bash
```

Function

Convert [Token](#) into an open Coin. Creates an [ActionRequest](#) for the "to_coin" action.

```bash
```

```bash
```

Convert an open Coin into a [Token](#) . Creates an [ActionRequest](#) for the "from_coin" action.

```bash
```

```bash
```

Join two [Token](#) s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Convert an open Coin into a [Token] . Creates an [ActionRequest] for the "from_coin" action.

```bash
```

```bash
```

Join two [Token] s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```

```bash

```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash

```

```bash

```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#).

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash

```

```bash

```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#).

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#).

```bash

```

```bash

```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash

```

```bash

```

Add a Config for a Rule in the [TokenPolicy](#). Rule configuration is independent from the [TokenPolicy](#). [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash

```

```bash

```

```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash

```

```bash

```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash

```

```bash

```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash

```

```bash

```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash

```

```bash

```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash

```

```bash

```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash
```

Amount of the ActionRequest .

```bash
```

```bash
```

Sender of the ActionRequest .

```bash
```

```bash
```

Recipient of the ActionRequest .

```bash
```

```bash
```

Approvals of the ActionRequest .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Join two [Token](#) s into one, always available.

```bash
```

```bash
```

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```bash
```

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new ActionRequest . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for spend and similar actions that deliver spent_balance to the TokenPolicy . For those actions use confirm_request_mut .

Aborts if:

```bash
```

```bash
```

Confirm the request against the TokenPolicy and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash

```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash

```

```bash

```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash

```

```bash

```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash

```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash

```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Split a [Token](#) with [amount](#) . Aborts if the [Token](#) . [balance](#) is lower than [amount](#) .

```bash
```

```bash
```

Create a zero [Token](#) .

```bash
```

```

```bash
```

Destroy an empty [Token](), fails if the balance is non-zero. Aborts if the [Token]() . [balance]() is not zero.

```bash
```

```bash
```

Transfer the [Token]() to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest]() . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy]() and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend]() and similar actions that deliver [spent_balance]() to the [TokenPolicy]() . For those actions use [confirm_request_mut]() .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy]() and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request]() this function requires mutable access to the [TokenPolicy]() and must be used on [spend]() action. After dealing with the spent balance it calls [confirm_request]() internally.

See [confirm_request]() for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest]() as the [TokenPolicyCap]() owner. This function allows [TokenPolicy]() owner to perform Capability-gated

actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Create a zero [Token] .

```bash
```

```bash
```

Destroy an empty [Token] , fails if the balance is non-zero. Aborts if the [Token] . [balance] is not zero.

```bash
```

```bash
```

Transfer the [Token] to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest] . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy] and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend] and similar actions that deliver [spent_balance] to the [TokenPolicy] . For those actions use [confirm_request_mut] .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy] and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request] this function requires mutable access to the [TokenPolicy] and must be used on [spend] action. After dealing with the spent balance it calls [confirm_request] internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash
```

Amount of the ActionRequest .

```bash
```

```bash
```

Sender of the ActionRequest .

```bash
```

```bash
```

Recipient of the ActionRequest .

```bash
```

```bash
```

Approvals of the ActionRequest .

```bash
```

```bash
```

Burned balance of the ActionRequest .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Destroy an empty [Token](#) , fails if the balance is non-zero. Aborts if the [Token](#) . [balance](#) is not zero.

```bash
```

```bash
```

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash

```

Get mutable access to the Config for a Rule in the [TokenPolicy] . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap] to guarantee that the [TokenPolicy] owner is the one who initiated the Config modification.

Aborts if:

```bash

```

```bash

```

Remove a Config for a Rule in the [TokenPolicy] . Unlike the [add_rule_config] , this function does not require a Rule witness, hence can be performed by the [TokenPolicy] owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest] .

Aborts if:

```bash

```

```bash

```

Check if a config for a Rule is set in the [TokenPolicy] without checking the type of the Config.

```bash

```

```bash

```

Check if a Config for a Rule is set in the [TokenPolicy] and that it matches the type provided.

```bash

```

```bash

```

Allows an [action] to be performed on the [Token] freely by adding an empty set of Rules for the [action] .

Aborts if the [TokenPolicyCap] is not matching the [TokenPolicy] .

```bash

```

```bash

```

Completely disallows an [action] on the [Token] by removing the record from the [TokenPolicy] . [rules] .

Aborts if the [TokenPolicyCap] is not matching the [TokenPolicy] .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable

storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Transfer the [Token](#) to the transaction sender.

```bash
```

```bash
```

Create a new [ActionRequest](#) . Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash

```

Function

Create a new [ActionRequest](). Publicly available method to allow for custom actions.

```bash
```

```bash
```

Confirm the request against the [TokenPolicy]() and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend]() and similar actions that deliver [spent_balance]() to the [TokenPolicy](). For those actions use [confirm_request_mut]().

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy]() and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request]() this function requires mutable access to the [TokenPolicy]() and must be used on [spend]() action. After dealing with the spent balance it calls [confirm_request]() internally.

See [confirm_request]() for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest]() as the [TokenPolicyCap]() owner. This function allows [TokenPolicy]() owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy]().

Aborts if request contains [spent_balance]() due to inability of the [TokenPolicyCap]() to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap]()).

```bash
```

```bash
```

Confirm an [ActionRequest]() as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy]().

Unlike [confirm_with_policy_cap]() this function allows [spent_balance]() to be consumed, decreasing the total_supply of the [Token]().

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

```
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
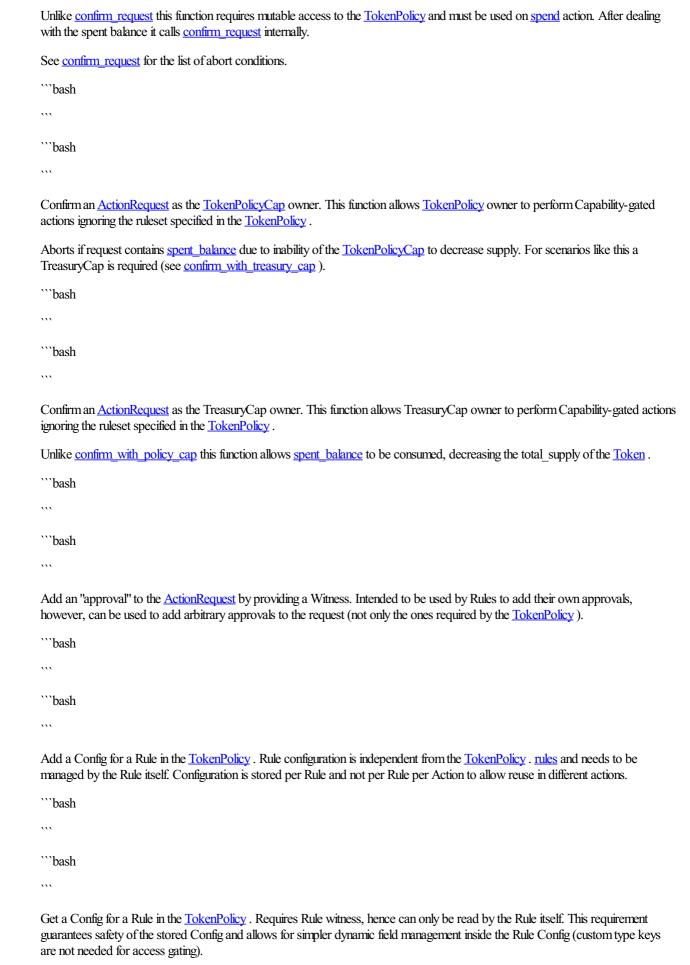```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Cannot be used for [spend](#) and similar actions that deliver [spent_balance](#) to the [TokenPolicy](#) . For those actions use [confirm_request_mut](#) .

Aborts if:

```bash
```

```bash
```

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike confirm_request this function requires mutable access to the TokenPolicy and must be used on spend action. After dealing with the spent balance it calls confirm_request internally.

See confirm_request for the list of abort conditions.

```bash
```

```bash
```

Confirm an ActionRequest as the TokenPolicyCap owner. This function allows TokenPolicy owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Aborts if request contains spent_balance due to inability of the TokenPolicyCap to decrease supply. For scenarios like this a TreasuryCap is required (see confirm_with_treasury_cap).

```bash
```

```bash
```

Confirm an ActionRequest as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the TokenPolicy .

Unlike confirm_with_policy_cap this function allows spent_balance to be consumed, decreasing the total_supply of the Token .

```bash
```

```bash
```

Add an "approval" to the ActionRequest by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the TokenPolicy).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](). Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap]() to guarantee that the [TokenPolicy]() owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](). Unlike the [add_rule_config](), this function does not require a Rule witness, hence can be performed by the [TokenPolicy]() owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest]().

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy]() without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy]() and that it matches the type provided.

```bash
```

```bash
```

Allows an [action]() to be performed on the [Token]() freely by adding an empty set of Rules for the [action]().

Aborts if the [TokenPolicyCap]() is not matching the [TokenPolicy]().

```bash
```

```bash

```
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

```
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Confirm the request against the [TokenPolicy](#) and return the parameters of the request: (Name, Amount, Sender, Recipient).

Unlike [confirm_request](#) this function requires mutable access to the [TokenPolicy](#) and must be used on [spend](#) action. After dealing with the spent balance it calls [confirm_request](#) internally.

See [confirm_request](#) for the list of abort conditions.

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the [TokenPolicyCap](#) owner. This function allows [TokenPolicy](#) owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Aborts if request contains [spent_balance](#) due to inability of the [TokenPolicyCap](#) to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap](#)).

```bash
```

```bash
```

Confirm an [ActionRequest](#) as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy](#) .

Unlike [confirm_with_policy_cap](#) this function allows [spent_balance](#) to be consumed, decreasing the total_supply of the [Token](#) .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash
```

```
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

```
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

```
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```

```bash

```

Approvals of the [ActionRequest] .

```bash

```

```bash

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Confirm an [ActionRequest] as the [TokenPolicyCap] owner. This function allows [TokenPolicy] owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy] .

Aborts if request contains [spent_balance] due to inability of the [TokenPolicyCap] to decrease supply. For scenarios like this a TreasuryCap is required (see [confirm_with_treasury_cap]).

```bash

```

```bash

```

Confirm an [ActionRequest] as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy] .

Unlike [confirm_with_policy_cap] this function allows [spent_balance] to be consumed, decreasing the total_supply of the [Token] .

```bash

```

```bash

```
```

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#) ).

```bash
```

```bash
```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](#) . Unlike the [add_rule_config](#) , this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#) .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

```
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Confirm an [ActionRequest] as the TreasuryCap owner. This function allows TreasuryCap owner to perform Capability-gated actions ignoring the ruleset specified in the [TokenPolicy] .

Unlike [confirm_with_policy_cap] this function allows [spent_balance] to be consumed, decreasing the total_supply of the [Token] .

```bash
```

```bash
```

Add an "approval" to the [ActionRequest] by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy]).

```bash
```

```bash
```

Add a Config for a Rule in the TokenPolicy . Rule configuration is independent from the TokenPolicy . rules and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash
```

```bash
```

Get a Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

```
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash

```

Approvals of the [ActionRequest](#) .

```bash

```

```bash

```

Burned balance of the [ActionRequest](#) .

```bash

```

```bash

```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Add an "approval" to the [ActionRequest](#) by providing a Witness. Intended to be used by Rules to add their own approvals, however, can be used to add arbitrary approvals to the request (not only the ones required by the [TokenPolicy](#)).

```bash

```

```bash

```

Add a Config for a Rule in the [TokenPolicy](#) . Rule configuration is independent from the [TokenPolicy](#) . [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash

```

```bash

```

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash
```

```bash
```

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

Burn a Token using the TreasuryCap.

```bash
```

```bash
```

Flush the TokenPolicy . spent_balance into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest] .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

```

```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Add a Config for a Rule in the [TokenPolicy](#). Rule configuration is independent from the [TokenPolicy](#). [rules](#) and needs to be managed by the Rule itself. Configuration is stored per Rule and not per Rule per Action to allow reuse in different actions.

```bash

```

```bash

```

Get a Config for a Rule in the [TokenPolicy](#). Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash

```

```bash

```

Get mutable access to the Config for a Rule in the [TokenPolicy](#). Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap](#) to guarantee that the [TokenPolicy](#) owner is the one who initiated the Config modification.

Aborts if:

```bash

```

```bash

```

Remove a Config for a Rule in the [TokenPolicy](#). Unlike the [add_rule_config](#), this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#).

Aborts if:

```bash

```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action.

Aborts if the TokenPolicyCap is not matching the TokenPolicy.

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy. rules.

Aborts if the TokenPolicyCap is not matching the TokenPolicy.

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy.

Aborts if the TokenPolicyCap is not matching the TokenPolicy.

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy. Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy.

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```

```

```bash

```

Name of the Transfer action.

```bash

```

```bash

```

Name of the Spend action.

```bash

```

```bash

```

Name of the ToCoin action.

```bash

```

```bash

```

Name of the FromCoin action.

```bash

```

```bash

```

The Action in the [ActionRequest](#) .

```bash

```

```bash

```

Amount of the [ActionRequest](#) .

```bash

```

```bash

```

Sender of the [ActionRequest](#) .

```bash

```

```bash

```

Recipient of the [ActionRequest](#) .

```bash

```

```bash

```

Approvals of the [ActionRequest](#) .

```bash

```

```bash

```

Burned balance of the [ActionRequest](#) .

```bash

```

```bash

```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Get a Config for a Rule in the [TokenPolicy](#) . Requires Rule witness, hence can only be read by the Rule itself. This requirement guarantees safety of the stored Config and allows for simpler dynamic field management inside the Rule Config (custom type keys are not needed for access gating).

Aborts if the Config is not present.

```bash

```

```bash

```

Get mutable access to the Config for a Rule in the [TokenPolicy](). Requires Rule witness, hence can only be read by the Rule itself, as well as [TokenPolicyCap]() to guarantee that the [TokenPolicy]() owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the [TokenPolicy](). Unlike the [add_rule_config](), this function does not require a Rule witness, hence can be performed by the [TokenPolicy]() owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest]().

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy]() without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy]() and that it matches the type provided.

```bash
```

```bash
```

Allows an [action]() to be performed on the [Token]() freely by adding an empty set of Rules for the [action]().

Aborts if the [TokenPolicyCap]() is not matching the [TokenPolicy]().

```bash
```

```bash
```

Completely disallows an [action]() on the [Token]() by removing the record from the [TokenPolicy]() . [rules]() .

Aborts if the [TokenPolicyCap]() is not matching the [TokenPolicy]().

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

```
```

The Action in the [ActionRequest] .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Get mutable access to the Config for a Rule in the TokenPolicy . Requires Rule witness, hence can only be read by the Rule itself, as well as TokenPolicyCap to guarantee that the TokenPolicy owner is the one who initiated the Config modification.

Aborts if:

```bash
```

```bash
```

Remove a Config for a Rule in the TokenPolicy . Unlike the add_rule_config , this function does not require a Rule witness, hence can be performed by the TokenPolicy owner on their own.

Rules need to make sure that the Config is present when performing verification of the ActionRequest .

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the TokenPolicy without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

```
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Remove a Config for a Rule in the [TokenPolicy](#). Unlike the [add_rule_config](#), this function does not require a Rule witness, hence can be performed by the [TokenPolicy](#) owner on their own.

Rules need to make sure that the Config is present when performing verification of the [ActionRequest](#).

Aborts if:

```bash
```

```bash
```

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#).

```bash
```

```bash
```

```
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

```
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Check if a config for a Rule is set in the [TokenPolicy](#) without checking the type of the Config.

```bash
```

```bash
```

Check if a Config for a Rule is set in the [TokenPolicy](#) and that it matches the type provided.

```bash
```

```bash
```

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#).

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#). [rules](#).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#).

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#).

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

```
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Check if a Config for a Rule is set in the TokenPolicy and that it matches the type provided.

```bash
```

```bash
```

Allows an action to be performed on the Token freely by adding an empty set of Rules for the action .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Completely disallows an action on the Token by removing the record from the TokenPolicy . rules .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Adds a Rule for an action with name in the TokenPolicy .

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Removes a rule for an action with name in the TokenPolicy . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the TokenPolicyCap is not matching the TokenPolicy .

```bash
```

```bash
```

Mint a Token with a given amount using the TreasuryCap.

```bash
```

```bash
```

```
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Allows an [action](#) to be performed on the [Token](#) freely by adding an empty set of Rules for the [action](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```
```

```bash
```

Amount of the ActionRequest .

```bash
```

```bash
```

Sender of the ActionRequest .

```bash
```

```bash
```

Recipient of the ActionRequest .

```bash
```

```bash
```

Approvals of the ActionRequest .

```bash
```

```bash
```

Burned balance of the ActionRequest .

```bash
```

```bash
```

Create a new RuleKey for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

```

## Function

Completely disallows an [action](#) on the [Token](#) by removing the record from the [TokenPolicy](#) . [rules](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash

```
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest] .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Adds a Rule for an action with name in the [TokenPolicy](#) .

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Removes a rule for an action with name in the [TokenPolicy](#) . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap](#) is not matching the [TokenPolicy](#) .

```bash
```

```bash
```

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

```

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Removes a rule for an action with name in the [TokenPolicy] . Returns the config object to be handled by the sender (or a Rule itself).

Aborts if the [TokenPolicyCap] is not matching the [TokenPolicy] .

```bash

```

```bash

```

Mint a [Token] with a given [amount] using the TreasuryCap.

```bash

```

```bash

```

Burn a [Token] using the TreasuryCap.

```bash

```

```bash

```

Flush the [TokenPolicy] . [spent_balance] into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash

```

```bash

```
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Mint a [Token](#) with a given [amount](#) using the TreasuryCap.

```bash
```

```bash
```

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the spent_balance of the TokenPolicy .

```bash
```

```bash
```

Returns the balance of the Token .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the ActionRequest .

```bash
```

```
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

```
```

Function

Burn a [Token](#) using the TreasuryCap.

```bash
```

```bash
```

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash
```

```bash
```

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash

```

```bash

```

Approvals of the [ActionRequest](#) .

```bash

```

```bash

```

Burned balance of the [ActionRequest](#) .

```bash

```

```bash

```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Flush the [TokenPolicy](#) . [spent_balance](#) into the TreasuryCap. This action is only available to the TreasuryCap owner.

```bash

```

```bash

```

Check whether an action is present in the rules VecMap.

```bash

```

```bash

```

Returns the rules required for a specific action.

```bash
```

```
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Check whether an action is present in the rules VecMap.

```bash
```

```bash
```

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Returns the rules required for a specific action.

```bash
```

```bash
```

Returns the [spent_balance] of the [TokenPolicy] .

```bash
```

```bash
```

Returns the [balance] of the [Token] .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Returns the [spent_balance](#) of the [TokenPolicy](#) .

```bash
```

```bash
```

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Returns the [balance](#) of the [Token](#) .

```bash
```

```bash
```

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Name of the Transfer action.

```bash
```

```bash
```

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest] .

```bash
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Name of the Spend action.

```bash
```

```bash
```

Name of the ToCoin action.

```bash
```

```bash
```

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash

```

```bash

```

Approvals of the [ActionRequest] .

```bash

```

```bash

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Name of the ToCoin action.

```bash

```

```bash

```

Name of the FromCoin action.

```bash

```

```bash

```

The Action in the [ActionRequest] .

```bash

```
```

```bash
```

Amount of the [ActionRequest] .

```bash
```

```bash
```

Sender of the [ActionRequest] .

```bash
```

```bash
```

Recipient of the [ActionRequest] .

```bash
```

```bash
```

Approvals of the [ActionRequest] .

```bash
```

```bash
```

Burned balance of the [ActionRequest] .

```bash
```

```bash
```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

```
```

Function

Name of the FromCoin action.

```bash
```

```bash
```

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

The Action in the [ActionRequest](#) .

```bash
```

```bash
```

Amount of the [ActionRequest](#) .

```bash
```

```bash
```

Sender of the [ActionRequest](#) .

```bash
```

```bash
```

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Amount of the [ActionRequest] .

```bash

```

```bash

```

Sender of the [ActionRequest] .

```bash

```

```bash

```

Recipient of the [ActionRequest] .

```bash

```

```bash

```

Approvals of the [ActionRequest] .

```bash

```

```bash
```

```
```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Sender of the [ActionRequest] .

```bash

```

```bash

```

Recipient of the [ActionRequest] .

```bash

```

```bash

```

Approvals of the [ActionRequest] .

```bash

```

```bash

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Recipient of the [ActionRequest](#) .

```bash
```

```bash
```

Approvals of the [ActionRequest](#) .

```bash
```

```bash
```

Burned balance of the [ActionRequest](#) .

```bash
```

```bash
```

Create a new [RuleKey](#) for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash
```

```bash
```

Function

Approvals of the [ActionRequest](#) .

```bash
```

```bash

```

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Burned balance of the [ActionRequest] .

```bash

```

```bash

```

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```

Function

Create a new [RuleKey] for a Rule. The is_protected field is kept for potential future use, if Rules were to have a freely modifiable storage as addition / replacement for the Config system.

The goal of is_protected is to potentially allow Rules store a mutable version of their configuration and mutate state on user action.

```bash

```

```bash

```