

Module `sui::kiosk_extension`

This module implements the Kiosk Extensions functionality. It allows exposing previously protected (only-owner) methods to third-party apps.

A Kiosk Extension is a module that implements any functionality on top of the Kiosk without discarding nor blocking the base. Given that Kiosk itself is a trading primitive, most of the extensions are expected to be related to trading. However, there's no limit to what can be built using the [kiosk_extension](#) module, as it gives certain benefits such as using Kiosk as the storage for any type of data / assets.

Trading functionality can utilize the PurchaseCap to build a custom logic around the purchase flow. However, it should be carefully managed to prevent asset locking.

[kiosk_extension](#) is a friend module to [kiosk](#) and has access to its internal functions (such as `place_internal` and `lock_internal` to implement custom authorization scheme for [place](#) and [lock](#) respectively).

[Struct](#) `Extension`

[Struct](#) `ExtensionKey`

[Constants](#)

[Function](#) `add`

[Function](#) `disable`

[Function](#) `enable`

[Function](#) `remove`

[Function](#) `storage`

[Function](#) `storage_mut`

[Function](#) `place`

[Function](#) `lock`

[Function](#) `is_installed`

[Function](#) `is_enabled`

[Function](#) `can_place`

[Function](#) `can_lock`

[Function](#) `extension`

[Function](#) `extension_mut`

The `Extension` struct contains the data used by the extension and the configuration for this extension. Stored under the [ExtensionKey](#) dynamic field.

The [ExtensionKey](#) is a typed dynamic field key used to store the extension configuration and data. `Ext` is a phantom type that is used to identify the extension witness.

Trying to add an extension while not being the owner of the Kiosk.

Extension is trying to access a permissioned action while not having the required permission.

Extension is not installed in the Kiosk.

Value that represents the [place](#) permission in the permissions bitmap.

Value that represents the [lock](#) and [place](#) permission in the permissions bitmap.

Add an extension to the Kiosk. Can only be performed by the owner. The extension witness is required to allow extensions define their set of permissions in the custom [add](#) call.

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

Check whether an extension of type Ext is installed.

Check whether an extension of type Ext is enabled.

Check whether an extension of type Ext can [place](#) into Kiosk.

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

Internal: get a read-only access to the Extension.

Internal: get a mutable access to the Extension.

Struct

The Extension struct contains the data used by the extension and the configuration for this extension. Stored under the [ExtensionKey](#) dynamic field.

```
```bash
```

```
```
```

The [ExtensionKey](#) is a typed dynamic field key used to store the extension configuration and data. Ext is a phantom type that is used to identify the extension witness.

```
```bash
```

```
```
```

Trying to add an extension while not being the owner of the Kiosk.

```
```bash
```

```
```
```

Extension is trying to access a permissioned action while not having the required permission.

```
```bash
```

```
```
```

Extension is not installed in the Kiosk.

```
```bash
```

```
```
```

Value that represents the [place](#) permission in the permissions bitmap.

```
```bash
```

```
```
```

Value that represents the [lock](#) and [place](#) permission in the permissions bitmap.

```
```bash
```

```
```
```

Add an extension to the Kiosk. Can only be performed by the owner. The extension witness is required to allow extensions define their set of permissions in the custom [add](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext is installed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext is enabled.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Internal: get a read-only access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Internal: get a mutable access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Struct

The [ExtensionKey](#) is a typed dynamic field key used to store the extension configuration and data. Ext is a phantom type that is used to identify the extension witness.

```
```bash
```

```
```
```

Trying to add an extension while not being the owner of the Kiosk.

```
```bash
```

```
```
```

Extension is trying to access a permissioned action while not having the required permission.

```
```bash
```

```
```
```

Extension is not installed in the Kiosk.

```
'''bash
```

```
'''
```

Value that represents the [place](#) permission in the permissions bitmap.

```
'''bash
```

```
'''
```

Value that represents the [lock](#) and [place](#) permission in the permissions bitmap.

```
'''bash
```

```
'''
```

Add an extension to the Kiosk. Can only be performed by the owner. The extension witness is required to allow extensions define their set of permissions in the custom [add](#) call.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Constants

Trying to add an extension while not being the owner of the Kiosk.

```
'''bash
```

```
'''
```

Extension is trying to access a permissioned action while not having the required permission.

```
'''bash
```

```
'''
```

Extension is not installed in the Kiosk.

```
'''bash
```

```
'''
```

Value that represents the [place](#) permission in the permissions bitmap.

```
'''bash
```

```
'''
```

Value that represents the [lock](#) and [place](#) permission in the permissions bitmap.

```
'''bash
```

```
'''
```


Add an extension to the Kiosk. Can only be performed by the owner. The extension witness is required to allow extensions define their set of permissions in the custom [add](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
```bash
```

```
```
```

```
'''bash
```

```
'''
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Internal: get a mutable access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Add an extension to the Kiosk. Can only be performed by the owner. The extension witness is required to allow extensions define their set of permissions in the custom [add](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext is installed.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext is enabled.

```
```bash
```

```
```
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Revoke permissions from the extension. While it does not remove the extension completely, it keeps it from performing any protected actions. The storage is still available to the extension (until it's removed).

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

'''bash

'''

'''bash

'''

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

'''bash

'''

'''bash

'''

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

'''bash

'''

'''bash

'''

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

'''bash

'''

'''bash

'''

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is installed.

'''bash

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Re-enable the extension allowing it to call protected actions (eg [place](#) , [lock](#)). By default, all added extensions are enabled. Kiosk owner can disable them via [disable](#) call.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

'''bash

'''

'''bash

'''

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

'''bash

'''

'''bash

'''

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

'''bash

'''

'''bash

'''

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

'''bash

'''

'''bash

'''

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is installed.

'''bash


```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Remove an extension from the Kiosk. Can only be performed by the owner, the extension storage must be empty for the transaction to succeed.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

'''bash

'''

'''bash

'''

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

'''bash

'''

'''bash

'''

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

'''bash

'''

'''bash

'''

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is installed.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is enabled.

'''bash

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Get immutable access to the extension storage. Can only be performed by the extension as long as the extension is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
```bash
```

```
```
```

Internal: get a read-only access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Internal: get a mutable access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Get mutable access to the extension storage. Can only be performed by the extension as long as the extension is installed. Disabling the extension does not prevent it from accessing the storage.

Potentially dangerous: extension developer can keep data in a Bag therefore never really allowing the KioskOwner to remove the extension. However, it is the case with any other solution (1) and this way we prevent intentional extension freeze when the owner wants to ruin a trade (2) - eg locking extension while an auction is in progress.

Extensions should be crafted carefully, and the KioskOwner should be aware of the risks.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Check whether an extension of type Ext is installed.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Check whether an extension of type Ext is enabled.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Internal: get a read-only access to the Extension.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Internal: get a mutable access to the Extension.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Function

Protected action: place an item into the Kiosk. Can be performed by an authorized extension. The extension must have the [place](#) permission or a [lock](#) permission.

To prevent non-tradable items from being placed into Kiosk the method requires a TransferPolicy for the placed type to exist.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Internal: get a mutable access to the Extension.

'''bash

'''

'''bash

'''

Function

Protected action: lock an item in the Kiosk. Can be performed by an authorized extension. The extension must have the [lock](#) permission.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is installed.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext is enabled.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext can [place](#) into Kiosk.

'''bash

'''

'''bash

'''

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

'''bash

'''

'''bash

'''

Internal: get a read-only access to the Extension.


```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Check whether an extension of type Ext is installed.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Check whether an extension of type Ext is enabled.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Check whether an extension of type Ext can [place](#) into Kiosk.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Check whether an extension of type Ext can [lock](#) items in Kiosk. Locking also enables [place](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a read-only access to the Extension.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Internal: get a mutable access to the Extension.

```
'''bash
```

```
'''
```

```
```bash
```

```
```
```

Function

Internal: get a read-only access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Internal: get a mutable access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Internal: get a mutable access to the Extension.

```
```bash
```

```
```
```

```
```bash
```

```
```
```