

Module sui::object_bag

Similar to [sui::bag](#), an [ObjectBag](#) is a heterogeneous map-like collection. But unlike [sui::bag](#), the values bound to these dynamic fields must be objects themselves. This allows for the objects to still exist in storage, which may be important for external tools. The difference is otherwise not observable from within Move.

Creates a new, empty bag

Adds a key-value pair to the bag [bag](#) : & mut [ObjectBag](#) Aborts with [sui::dynamic_field::EFieldAlreadyExists](#) if the bag already has an entry with that key k: K.

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#). Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#). Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

Returns the size of the bag, the number of key-value pairs

Returns true iff the bag is empty (if [length](#) returns 0)

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

Struct

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Creates a new, empty bag

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Adds a key-value pair to the bag [bag](#) : & mut [ObjectBag](#) Aborts with [sui::dynamic_field::EFieldAlreadyExists](#) if the bag already has an entry with that key k: K.

```
```bash
```

```
...
```

```
```bash
```

```
'''
```

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the size of the bag, the number of key-value pairs

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Constants

```
```bash
```

```
...
```

Creates a new, empty bag

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Adds a key-value pair to the bag [bag](#) : & mut [ObjectBag](#) Aborts with [sui::dynamic\\_field::EFieldAlreadyExists](#) if the bag already has an entry with that key k: K.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
...
```

```
```bash
```

'''

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

'''bash

'''

'''bash

'''

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

'''bash

'''

'''bash

'''

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

'''bash

'''

'''bash

'''

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

'''bash

'''

'''bash

'''

Returns the size of the bag, the number of key-value pairs

'''bash

'''

'''bash

'''

Returns true iff the bag is empty (if [length](#) returns 0)

'''bash

'''

'''bash

'''

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Creates a new, empty bag

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Adds a key-value pair to the bag [bag](#) : & mut [ObjectBag](#) Aborts with [sui::dynamic\\_field::EFieldAlreadyExists](#) if the bag already has an entry with that key k: K.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with

[sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the size of the bag, the number of key-value pairs

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Adds a key-value pair to the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldAlreadyExists](#) if the bag already has an entry with that key k: K.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic\\_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic\\_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the size of the bag, the number of key-value pairs

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Function

Immutably borrows the value associated with the key in the bag [bag](#) : & [ObjectBag](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
'''bash
```


'''

'''bash

'''

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

'''bash

'''

'''bash

'''

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

'''bash

'''

'''bash

'''

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

'''bash

'''

'''bash

'''

Returns the size of the bag, the number of key-value pairs

'''bash

'''

'''bash

'''

Returns true iff the bag is empty (if [length](#) returns 0)

'''bash

'''

'''bash

'''

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

'''bash

'''

'''bash

'''

Returns the ID of the object associated with the key if the bag has an entry with key k: K. Returns none otherwise

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Mutably borrows the value associated with the key in the bag [bag](#) : & mut [ObjectBag](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the size of the bag, the number of key-value pairs

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Function

Mutably borrows the key-value pair in the bag [bag](#) : & mut [ObjectBag](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the bag does not have an entry with that key k: K. Aborts with [sui::dynamic_field::EFieldTypeMismatch](#) if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Returns true iff there is a value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Returns the size of the bag, the number of key-value pairs

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Function

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#)

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Returns the size of the bag, the number of key-value pairs

```
```bash
```

```
```
```

```
```bash
```

```
'''
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Returns true iff there is an value associated with the key k: K in the bag [bag](#) : & [ObjectBag](#) with an assigned value of type V

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the size of the bag, the number of key-value pairs

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Returns the size of the bag, the number of key-value pairs

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns true iff the bag is empty (if [length](#) returns 0)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Returns true iff the bag is empty (if [length](#) returns 0)

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Destroys an empty bag Aborts with [EBagNotEmpty](#) if the bag still contains values

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```