Module sui::bag

A bag is a heterogeneous map-like collection. The collection is similar to sui::table in that its keys and values are not stored within the Bag value, but instead are stored using Sui's object system. The Bag struct acts only as a handle into the object system to retrieve those keys and values. Note that this means that Bag values with exactly the same key-value mapping will not be equal, with ==, at runtime. For example

At it's core, sui::bag is a wrapper around UID that allows for access to sui::dynamic_field while preventing accidentally stranding field values. A UID can be deleted, even if it has dynamic fields associated with it, but a bag, on the other hand, must be empty to be destroyed.

Creates a new, empty bag

Adds a key-value pair to the bag bag : & mut Bag Aborts with sui::dynamic_field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 an value associated with the key k: K in the bag bag : & Bag

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

Struct

```bash

```

```bash

```

Creates a new, empty bag

```bash

```

```bash

```

Adds a key-value pair to the bag bag : & mut Bag Aborts with sui::dynamic_field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

```bash

```

```bash
```

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 : & Bag

```bash
```

```bash
```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Constants

```bash

```

Creates a new, empty bag

```bash

```

```bash

```

Adds a key-value pair to the bag bag : & mut Bag Aborts with sui::dynamic_field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

```bash

```

```bash

```

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Creates a new, empty bag

```bash

```

```bash

```

Adds a key-value pair to the bag bag : & mut Bag Aborts with sui::dynamic_field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

```bash

```

```bash

```

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 : & Bag

```bash

```

```bash
```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Adds a key-value pair to the bag bag : & mut Bag Aborts with sui::dynamic_field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

```bash

```

```bash

```

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Immutable borrows the value associated with the key in the bag bag : & Bag . 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 Bag . 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 Bag 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 : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Mutably borrows the value associated with the key in the bag bag : & mut Bag . 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 Bag 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 : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Mutably borrows the key-value pair in the bag bag : & mut Bag 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 : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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

```

Function

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag

```bash

```

```bash

```

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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
```

Function

Returns true iff there is an value associated with the key k: K in the bag bag : & Bag 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
```

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
```

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
```

Function

Destroys an empty bag Aborts with EBagNotEmpty if the bag still contains values

```bash
```

```bash
```