Module sui::object_table

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

Creates a new, empty table

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

Immutable borrows the value associated with the key in the table table : & ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

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

Returns true if the table is empty (if length returns 0)

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

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

Struct

```bash
```

```bash
```

Creates a new, empty table

```bash
```

```bash
```

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

```bash
```

```bash
```

Immutable borrows the value associated with the key in the table table : & ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```

```bash

```

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash

```

```bash

```

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash

```

```bash

```

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash

```

```bash

```

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

```bash

```

```bash

```

Returns true if the table is empty (if length returns 0)

```bash

```

```bash

```

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

```bash

```

```bash

```

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

```bash
```

```bash
```

Constants

```bash
```

Creates a new, empty table

```bash
```

```bash
```

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

```bash
```

```bash
```

Immutable borrows the value associated with the key in the table table : & ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Returns true if there is a value associated with the key k: K in table [table](#) : & [ObjectTable](#)

```bash
```

```bash
```

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

```bash
```

```bash
```

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

```bash
```

```bash
```

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

Creates a new, empty table

```bash
```

```bash
```

Adds a key-value pair to the table [table](#) : & mut [ObjectTable](#) Aborts with [sui::dynamic_field::EFieldAlreadyExists](#) if the table already

has an entry with that key k: K.

```bash
```

```bash
```

Immutable borrows the value associated with the key in the table table : & ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash
```

```bash
```

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

```bash
```

```bash
```

Returns true if the table is empty (if length returns 0)

```bash
```

```

```bash

```

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash

```

```bash

```

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

```bash

```

```bash

```

Function

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

```bash

```

```bash

```

Immutable borrows the value associated with the key in the table [table](#) : & [ObjectTable](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the table does not have an entry with that key k: K.

```bash

```

```bash

```

Mutably borrows the value associated with the key in the table [table](#) : & mut [ObjectTable](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the table does not have an entry with that key k: K.

```bash

```

```bash

```

Removes the key-value pair in the table [table](#) : & mut [ObjectTable](#) and returns the value. Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the table does not have an entry with that key k: K.

```bash

```
```

```bash
```

Returns true if there is a value associated with the key k: K in table [table](#) : & [ObjectTable](#)

```bash
```

```bash
```

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

```bash
```

```bash
```

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

```bash
```

```bash
```

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

Immutable borrows the value associated with the key in the table [table](#) : & [ObjectTable](#) . Aborts with [sui::dynamic_field::EFieldDoesNotExist](#) if the table does not have an entry with that key k: K.

```bash
```

```bash
```

```
```

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash
```

```bash
```

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

```bash
```

```bash
```

Returns true if the table is empty (if length returns 0)

```bash
```

```bash
```

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

```bash
```

```bash
```

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

```bash
```

```
```

```bash
```

Function

Mutably borrows the value associated with the key in the table table : & mut ObjectTable . Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash
```

```bash
```

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

```bash
```

```bash
```

Returns true if the table is empty (if length returns 0)

```bash
```

```bash
```

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

Removes the key-value pair in the table table : & mut ObjectTable and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the table does not have an entry with that key k: K.

```bash
```

```bash
```

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash
```

```bash
```

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

```bash
```

```bash
```

Returns true if the table is empty (if length returns 0)

```bash
```

```bash
```

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

Returns true if there is a value associated with the key k: K in table table : & ObjectTable

```bash
```

```bash
```

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

```bash
```

```bash
```

Returns true if the table is empty (if length returns 0)

```bash
```

```bash
```

Destroys an empty table Aborts with ETableNotEmpty if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

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

```bash
```

```bash
```

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

```bash
```

```bash
```

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

Function

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

```bash
```

```bash
```

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```

```

## Function

Destroys an empty table Aborts with [ETableNotEmpty](#) if the table still contains values

```bash
```

```bash
```

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

```bash
```

```bash
```