

HashMap

Generated by Doxygen 1.9.8

| | |
|--|----------|
| 1 File Index | 1 |
| 1.1 File List | 1 |
| 2 File Documentation | 3 |
| 2.1 hashmap.h File Reference | 3 |
| 2.1.1 Typedef Documentation | 4 |
| 2.1.1.1 hash | 4 |
| 2.1.1.2 HashMap | 4 |
| 2.1.2 Function Documentation | 4 |
| 2.1.2.1 hm_create() | 4 |
| 2.1.2.2 hm_destroy() | 4 |
| 2.1.2.3 hm_get() | 5 |
| 2.1.2.4 hm_put() | 5 |
| 2.1.2.5 hm_remove() | 5 |
| 2.1.2.6 hm_set() | 6 |
| 2.1.2.7 hm_size() | 6 |
| 2.2 hashmap.h | 6 |
| Index | 9 |

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

| | |
|-------------------------------------|---|
| hashmap.h | 3 |
|-------------------------------------|---|

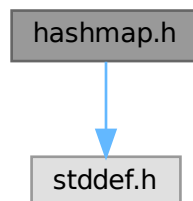
Chapter 2

File Documentation

2.1 hashmap.h File Reference

```
#include <stddef.h>
```

Include dependency graph for hashmap.h:



Typedefs

- typedef struct [HashMap](#) * [HashMap](#)
- typedef size_t(* [hash](#)) (const void *key, size_t key_size)

Functions

- [HashMap hm_create](#) (size_t hm_capacity, size_t key_size, size_t value_size, [hash](#) hash_func)
Creates a new HashMap with a hash function.
- int [hm_destroy](#) ([HashMap](#) hm)
Destroys the HashMap.
- void * [hm_get](#) ([HashMap](#) hm, const void *key)
Retrieves the value at the specified key.
- int [hm_set](#) ([HashMap](#) hm, const void *key, const void *value)
Updates the value at the specified key.
- int [hm_put](#) ([HashMap](#) hm, const void *key, const void *value)
Adds a new key-value pair to the HashMap.
- size_t [hm_size](#) ([HashMap](#) hm)
Returns the size of the HashMap.
- int [hm_remove](#) ([HashMap](#) hm, const void *key)
Removes the key-value pair from the HashMap.

2.1.1 Typedef Documentation

2.1.1.1 hash

```
typedef size_t(* hash) (const void *key, size_t key_size)
```

2.1.1.2 HashMap

```
typedef struct HashMap* HashMap
```

2.1.2 Function Documentation

2.1.2.1 hm_create()

```
HashMap hm_create (
    size_t hm_capacity,
    size_t key_size,
    size_t value_size,
    hash hash_func )
```

Creates a new HashMap with a hash function.

Parameters

| | |
|--------------------|--|
| <i>hm_capacity</i> | The initial capacity of the HashMap |
| <i>key_size</i> | The sizeof value of the key |
| <i>value_size</i> | The sizeof value of the value |
| <i>hash_func</i> | a custom hash function. Pass NULL for generic hashing. |

Returns

HashMap

2.1.2.2 hm_destroy()

```
int hm_destroy (
    HashMap hm )
```

Destroys the HashMap.

Parameters

| | |
|-----------|-------------|
| <i>hm</i> | The HashMap |
|-----------|-------------|

Returns

Success code

2.1.2.3 hm_get()

```
void * hm_get (
    HashMap hm,
    const void * key )
```

Retrieves the value at the specified key.

Parameters

| | |
|------------|-------------|
| <i>hm</i> | The HashMap |
| <i>key</i> | The key |

Returns

The value at the key

2.1.2.4 hm_put()

```
int hm_put (
    HashMap hm,
    const void * key,
    const void * value )
```

Adds a new key-value pair to the HashMap.

Parameters

| | |
|--------------|-------------|
| <i>hm</i> | The HashMap |
| <i>key</i> | The key |
| <i>value</i> | The value |

Returns

Success code

2.1.2.5 hm_remove()

```
int hm_remove (
    HashMap hm,
    const void * key )
```

Removes the key-value pair from the HashMap.

Parameters

| | |
|------------|-------------|
| <i>hm</i> | The HashMap |
| <i>key</i> | The key |

Returns

Success code

2.1.2.6 hm_set()

```
int hm_set (
    HashMap hm,
    const void * key,
    const void * value )
```

Updates the value at the specified key.

Parameters

| | |
|--------------|-------------|
| <i>hm</i> | The HashMap |
| <i>key</i> | The key |
| <i>value</i> | The value |

Returns

Success code

2.1.2.7 hm_size()

```
size_t hm_size (
    HashMap hm )
```

Returns the size of the HashMap.

Parameters

| | |
|-----------|-------------|
| <i>hm</i> | The HashMap |
|-----------|-------------|

Returns

The size

2.2 hashmap.h

[Go to the documentation of this file.](#)

```
00001 //
00002 // Created by Emanuel on 02.09.2024.
00003 //
00004
00005 #ifndef HASHMAP_H
00006 #define HASHMAP_H
00007
00008 #include <stddef.h>
00009
00010 typedef struct HashMap *HashMap;
00011 typedef size_t (*hash)(const void *key, size_t key_size);
00012
00022 HashMap hm_create(size_t hm_capacity, size_t key_size, size_t value_size, hash hash_func);
00023
00030 int hm_destroy(HashMap hm);
00031
00039 void *hm_get(HashMap hm, const void *key);
00040
00049 int hm_set(HashMap hm, const void *key, const void *value);
00050
00059 int hm_put(HashMap hm, const void *key, const void *value);
00060
00067 size_t hm_size(HashMap hm);
00068
00076 int hm_remove(HashMap hm, const void *key);
00077
00078 #endif //HASHMAP_H
```


Index

- hash
 - hashmap.h, [4](#)
- HashMap
 - hashmap.h, [4](#)
- hashmap.h, [3](#)
 - hash, [4](#)
 - HashMap, [4](#)
 - hm_create, [4](#)
 - hm_destroy, [4](#)
 - hm_get, [5](#)
 - hm_put, [5](#)
 - hm_remove, [5](#)
 - hm_set, [6](#)
 - hm_size, [6](#)
- hm_create
 - hashmap.h, [4](#)
- hm_destroy
 - hashmap.h, [4](#)
- hm_get
 - hashmap.h, [5](#)
- hm_put
 - hashmap.h, [5](#)
- hm_remove
 - hashmap.h, [5](#)
- hm_set
 - hashmap.h, [6](#)
- hm_size
 - hashmap.h, [6](#)