# hash-ring-ctypes Documentation

***Release 1.0.0***

**Matt Dennewitz**

October 02, 2013

# CONTENTS

`hash-ring-ctypes` is a fast ctypes-based wrapper around libhashring.

This library is available on GitHub under the MIT license.

Contents:

CHAPTER

ONE

# INSTALLATION AND USAGE

Before using this library, please make sure you have *libhashring* installed. *libhashring* is available here.

## 1.1 Installation

For now, clone from this repo:

```
$ pip install -e git+https://github.com/mattdennewitz/hash-ring-ctypes#egg=hash_ring
```

## 1.2 Usage

Creating a hash ring is simple: create an instance of the *HashRing* class, and populate with nodes.

```
import hash_ring

# create a ring with nodes with 10 replicas and four nodes
nodes = [’cdn1’, ’cdn2’, ’cdn3’, ’cdn4’]
ring = hash_ring.HashRing(replicas=10, nodes=nodes)

# or create an empty ring with the default number of replicas (five)
# and add nodes by hand
ring = hash_ring.HashRing()
map(ring.add_node, [’cdn1’, ’cdn2’, ’cdn3’, ’cdn4’])
```

Once you have an instance of *HashRing*, you can start finding nodes for values with `lookup()`

```
# look up a node for a certain value
fn = ’artwork/1/header.jpg’
node = ring.lookup(fn)
```

If you need to remove a node from the ring – for example, you might need to remove a feisty CDN endpoint – you can do so with `remove_node()`

CHAPTER

TWO

# GUTS

## 2.1 HashRing

## 2.2 Return codes

**HASH_RING_OK**

Returned when a hash ring operation is successful.

**HASH_RING_ERR**

Returned when a hash ring operation is successful.

## 2.3 Hash functions

**HASH_FUNCTION_MD5**

Used when hashing should be done via MD5. Default.

**HASH_FUNCTION_SHA1**

Used when hashing should be done via SHA1.

CHAPTER

THREE

# INDICES AND TABLES

- *genindex*
- *search*