

---

# Chippr Robotics

*Release 1.1.48*

**Aug 18, 2023**



---

## Contents

---

<b>1</b>	<b>BridgetteDB</b>	<b>1</b>
<b>2</b>	<b>About</b>	<b>3</b>
2.1	Use . . . . .	3
2.2	Data flow . . . . .	3
2.3	Install . . . . .	3
2.4	Enable . . . . .	4
<b>3</b>	<b>Functions</b>	<b>5</b>
3.1	Get . . . . .	5
3.2	Set . . . . .	5
3.3	Remove . . . . .	5
3.4	Unlock . . . . .	5
<b>4</b>	<b>Indices and tables</b>	<b>7</b>



# CHAPTER 1

---

BridgetteDB

---



BridgetteDB is a lightweight, key value store database built on top of the ethereum blockchain built for nodeJS. It sets and retrives values to a key.

## 2.1 Use

To use the system, you will need an ethereum node or RPC endpoint for communication. To set a kvs you will need to have an account which can be unlocked

## 2.2 Data flow

All data is saved within the BridgetteDB smart contract located on the public blockchain. #DO NOT SAVE ANYTHING THAT IS NOT PUBLIC INFORMATION!#

When interacting with data on chain, a DBKEY is appended to the data to index within the storage space.

The format of a database entry is: DBKEY + key : value The DBKEY is stored when initilizing the db and will be used for all keys within the session without a need to explicitly append it.

## 2.3 Install

Install using a packge manager

```
$ yarn add @chipprbots/bridgetteDB
```

```
$ npm install -s @chipprbots/bridgetteDB
```

## 2.4 Enable

To use BridgetteDB, require it within a script and create a new instance of the DB agent. All values are strings.

```
var bdb = require('@chipprbots/bridgetteDB');
var db = new bdb({
  "nodeAddr": url of the ethereum node,
  "accountAddress": ethereum account to use for transactions,
  "accountPasswd" : ethereum account password,
  "kvsAddr" : address the kvs is deployed to,
  → "0x57EEB5d4D3E1Ac75D51067AE2dCF78922CF3F189",
  "DBKEY": user assigned unique key for storage
})
```



### 3.1 Get

Given a key, retrieve any data stored within the DB

```
db.get ( _Key)
```

### 3.2 Set

Given a key and Value, stores the value in the DB using the Key

```
db.set ( _key, _value)
```

### 3.3 Remove

Given a key, remove a DB entry

```
db.rem( _key )
```

### 3.4 Unlock

Unlock the DB for writing

```
db.unlock()
```



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`