
MerkavaDB Documentation

Release 0.2.0

Adam Hopkins

Sep 25, 2017

Contents

1	What is MerkavaDB?	3
2	So, why would I use it?	5
3	How do I use it?	7
3.1	What kind of data?	7
3.2	What is a channel?	7
4	Supported Operations	9
5	Roadmap	11
6	Current Version	13
6.1	Installation	13
6.2	Usage	13
6.3	Contributing	13
6.4	Credits	15
6.5	History	15

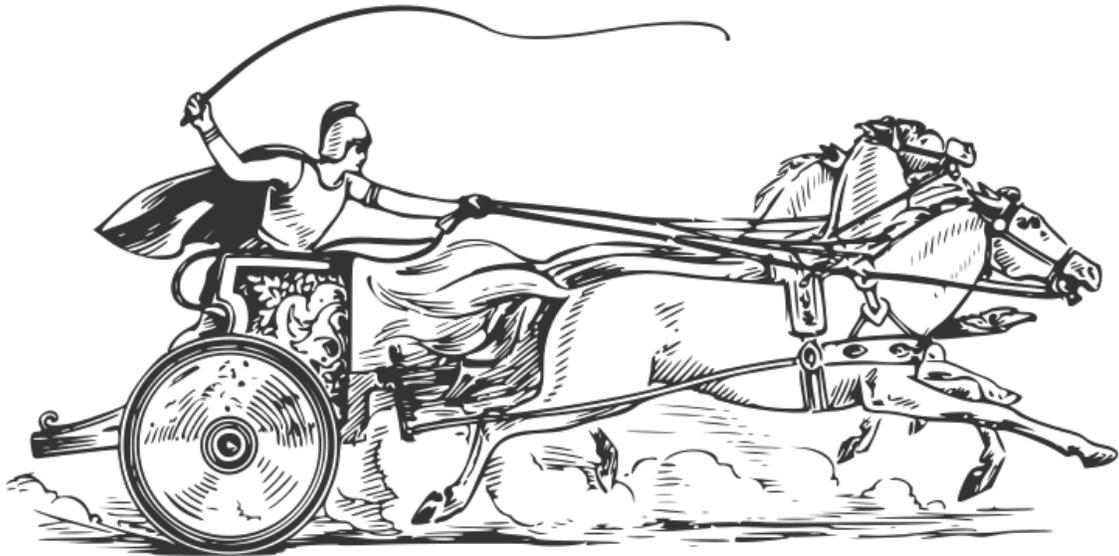
pypi package 0.2.0

build passing

A fast ordered NoSQL database.

[Documentation](#) | [GitHub](#)

Note: This is still in **active** development. Things will change. If you are interested in helping out, or would like to see any particular features added, let me know.



MERKAVADB

CHAPTER 1

What is MerkavaDB?

A very fast, and lightweight DB for storing ordered data. The order it comes out is the order that it went in. The API is meant to be small and without bloated queries. Consequently, there is a limited set of queries to be made since the primary tool is getting objects in and out of storage in a specific order.

CHAPTER 2

So, why would I use it?

Because it is fast. And it is simple.

Let's say, for example, you are building an application. As a part of your application, you want to have a chat or a news feed. The data will ALWAYS be displayed in the same order. Well, you can persist your data objects and feel confident that they will always be in the same order, no matter what.

MerkavaDB stores data in a similar format to JSON. So, it is schemaless and will allow you to store data in whatever format you need.

By making HTTP calls to the database server. All you need to do is specify a “channel” and some data.

3.1 What kind of data?

- nulls
- booleans
- integers
- floats
- strings
- arrays/lists
- maps/dicts

Basically anything you would pass by JSON.

3.2 What is a channel?

A channel is a division of data. All data is stored in a sequential order given the channel that it is in. For example, it could be a single chat room or news feed.

CHAPTER 4

Supported Operations

- HTTP POST /<channel>/ - create
- HTTP GET /<channel>/<id>/ - retrieve a single record
- HTTP PATCH /<channel>/<id>/ - update a record
- HTTP DELETE /<channel>/<id>/ - delete a record
- HTTP PUT /<channel>/<id>/ - restore a deleted record
- HTTP GET /<channel>/recent/<X> - retrieve an array of the X most recent records

CHAPTER 5

Roadmap

- Drivers for: Python, NodeJS, Java
- Test coverage
- Documentation
- Clean up utilities
- User interface
- Debian installer
- Single script installer
- Configuration options
- Examples

version 0.2.0

6.1 Installation

At the command line either via `easy_install` or `pip`:

```
$ easy_install merkava
$ pip install merkava
```

Or, if you have `virtualenvwrapper` installed:

```
$ mkvirtualenv merkava
$ pip install merkava
```

6.2 Usage

To use MerkavaDB in a project:

```
import merkava
```

6.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

6.3.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/ahopkins/merkava/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

MerkavaDB could always use more documentation, whether as part of the official MerkavaDB docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/ahopkins/merkava/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.3.2 Get Started!

Ready to contribute? Here’s how to set up *merkava* for local development.

1. Fork the *merkava* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/merkava.git
```

3. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you're done making changes, check that your changes pass style and unit tests, including testing other Python versions with tox:

```
$ tox
```

To get tox, just pip install it.

5. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

6.3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check <https://travis-ci.org/ahopkins/merkava> under pull requests for active pull requests or run the `tox` command and make sure that the tests pass for all supported Python versions.

6.3.4 Tips

To run a subset of tests:

```
$ py.test test/test_merkava.py
```

6.4 Credits

6.4.1 Development Lead

- Adam Hopkins <admhpks@gmail.com>

6.4.2 Contributors

None yet. Why not be the first?

6.5 History

6.5.1 0.2.0 (2017-09-25)

- History and tooling scripts

- Functioning API with Sanic server
- Versioning tools

6.5.2 0.1.0 (2017-09-24)

- First release on PyPI.

If you have any suggestions or questions about **MerkavaDB** feel free to email me at admhpkins@gmail.com.

If you encounter any errors or problems with **MerkavaDB**, please let me know! Open an Issue at the GitHub <https://github.com/ahopkins/merkava> main repository.