# cryptbase

## *Release 1.0.0*

**Alexandr Mansurov**

**Oct 17, 2021**

# CONTENTS

CHAPTER

ONE

# OVERVIEW

Protect yourself and your customers with database encryption.

- Free software: MIT license

## 1.1 Installation

pip install cryptbase

## 1.2 Usage

You can use cryptbase with django ORM or witg SQLAlchemy.

To use with SQLAlchemy:

from cryptbase import EncryptedText sensitive = Column(EncryptedText(key=DB_KEY))

To use with django:

from cryptbase import EncryptedTextField sensitive = EncryptedTextField(key=DB_KEY)

DB_KEY is 32 bytes encryption key as hex encoded string. Fields behave as TEXT fields, data are transparently encrypted when storing into the database and decrypted on retrieval.

## 1.3 Development

To run all the tests run:

```
tox
```

CHAPTER

TWO

# INSTALLATION

At the command line:

```
pip install cryptbase
```

## CHAPTER THREE

# USAGE

To use cryptbase in a project:

```
import cryptbase
```

CHAPTER

FOUR

# REFERENCE

## 4.1 cryptbase

CHAPTER

FIVE

# CONTRIBUTING

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

## 5.1 Bug reports

When 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.

## 5.2 Documentation improvements

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

## 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://code.eghuro.com/cryptbase/python-cryptbase/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 code contributions are welcome :)

## 5.4 Development

To set up *python-cryptbase* for local development:

1. Fork python-cryptbase (look for the "Fork" button).
2. Clone your fork locally:

   ```
   git clone git@code.eghuro.com:YOURGITHUBNAME/python-cryptbase.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 run all the checks and docs builder with tox one command:

   ```
   tox
   ```

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.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`).
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

# CHAPTER SIX

# AUTHORS

- Alexandr Mansurov - https://eghuro.cz

# CHAPTER SEVEN

# CHANGELOG

## 7.1 1.0.0 (2021-10-17)

- Encrypted text field for Django and SQLAlchemy with AES-256-CTR

CHAPTER

EIGHT

# INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

# INDEX