
Mantis Core Django Application Documentation

Release 0.2.0

Siemens

February 26, 2014

1	Django Mantis Core	3
1.1	Acknowledgments	3
2	Installation	5
3	Contributing	7
3.1	Types of Contributions	7
3.2	Get Started!	8
3.3	Pull Request Guidelines	8
4	Credits	9
4.1	Development Lead	9
4.2	Contributors	9
5	History	11
5.1	0.1.0 (2013-12-19)	11
5.2	0.0.9 (2013-12-17)	11

Contents:

Django Mantis Core

A wrapper around the Django Dingos application used by the Mantis Cyber Threat Intelligence Management Framework. The rationale for this wrapper is to provide a location for code specific to cyber threat intelligence management that is to be added to the underlying models. Such code should not be added to Dingos, because Dingos is a generic framework.

Please refer to the documentation and developer guidelines of [Django Dingos](#) and [MANTIS](#).

The full documentation of the Mantis core module is at

<http://django-mantis-core.readthedocs.org/en/latest/>

1.1 Acknowledgments

The basic layout for this Django app with out-of-the-box configuration of `setup.py` for easy build, submission to PyPi, etc., and Sphinx documentation tree was generated with Audrey Roy's excellent [Cookiecutter](#) and Daniel Greenfield's [cookiecutter-djangopackage](#) template.

Installation

1. Follow the Installation instructions for django-dingos
2. Do:

```
$ pip install django-mantis-core
```

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:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at <https://github.com/siemens/django-mantis-core/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Write Documentation

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

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/siemens/django-mantis-core/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 :)

3.2 Get Started!

Ready to contribute? Here's how to set up *django-mantis-core* for local development.

1. Fork the *django-mantis-core* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-mantis-core.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-mantis-core
$ cd django-mantis-core/
$ python setup.py develop
```

4. Create a branch for local development:

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

Now you can make your changes locally.

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
```

7. Submit a pull request through the GitHub website.

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 2.7.

Credits

4.1 Development Lead

- Siemens <mantis.cert@siemens.com>

4.2 Contributors

None yet. Why not be the first?

History

5.1 0.2.0 (2014-02-26)

Added corresponding abstract model classes for models introduced in DINGOS 0.2.0.

5.2 0.1.0 (2013-12-19)

- Bump dependencies

5.3 0.0.9 (2013-12-17)

- First release on PyPI.