MQLAlchemy Documentation

Release 0.2.0

Nicholas Repole

Jul 31, 2020

Contents

1 Why? 3

2 Supported Operators 5

3 Examples 7

4 Contributing 9

5 License 11

5.1 Changes 11

5.2 Installation 12

5.3 Support 12

6 Indices and tables 13

Query SQLAlchemy models using MongoDB style syntax.

CHAPTER 1

Why?

The need arose for me to be able to pass complex database filters from client side JavaScript to a Python server. I started building some JSON style syntax to do so, then realized such a thing already existed. I've never seriously used MongoDB, but the syntax for querying lends itself pretty perfectly to this use case.

That sounds pretty dangerous...

It can be. When using this with any sort of user input, you'll want to pass in a whitelist of attributes that are ok to query, otherwise you'll open the possibility of leaked passwords and all sorts of other scary stuff.

So, can I actually use this for a serious project?

Maybe? There's some decent test coverage, but this certainly isn't a very mature project yet.

I'll be pretty active in supporting this, so if you are using this and run into problems, I should be pretty quick to fix them.

How fast is it?

I'm sure my actual syntax parsing is inefficient and has loads of room for improvement, but the time it takes to parse should be minimal compared to the actual database query, so this shouldn't slow your queries down too much.

CHAPTER 2

Supported Operators

- $and
- $or
- $not
- $nor
- $in
- $nin
- $gt
- $gte
- $lt
- $lte
- $ne
- $mod

Custom operators added for convenience:

- $eq - Explicit equality check.
- $like - Search a text field for the given value.

Not yet supported, but would like to add:

- Index based relation queries. Album.tracks.0.track_id won't work.
- $regex

CHAPTER 3

Examples

```
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from mqlalchemy import apply_mql_filters
from myapp.mymodels import Album

# get your sqlalchemy db session here
db_engine = create_engine("sqlite+pysqlite:///mydb.sqlite")
DBSession = sessionmaker(bind=db_engine)
db_session = DBSession()

# define which fields of Album are ok to query
whitelist = ["album_id", "artist.name", "tracks.playlists.name"]
# Find all albums that are either by Led Zeppelin or have a track
# that can be found on the "Grunge" playlist.
filters = {
    "$or": [
        {"tracks.playlists.name": "Grunge"},
        {"artist.name": "Led Zeppelin"}
    ]
}
query = apply_mql_filters(db_session, Album, filters, whitelist)
matching_records = query.all()
```

For more, please see the included tests, as they're probably the easiest way to get an idea of how the library can be used.

CHAPTER 4

Contributing

Submit a pull request and make sure to include an updated AUTHORS with your name along with an updated CHANGES.rst.

CHAPTER 5

License

MIT

5.1 Changes

5.1.1 Release 0.2.0

Incompatible changes

- convert_key_names parameter for apply_mql_filters removed.
- convert_key_names_func parameter for apply_mql_filters added.
- All RecordClass parameters names changed to model_class.

Features added

- May also now pass in a function instead of a whitelist.

Documentation

- Changed docstring format to match Google's style guide.

5.1.2 Release 0.1.4

Features added

- Ability to convert camelCase search parameters to underscore.
- Internationalization support for error messages.

Documentation

- Removed now broken badges.

5.1.3 Release 0.1.3

Incompatible changes

- Changed license from BSD to MIT.

Documentation

- Added more badges.
- Included install instructions for installing from source.

5.1.4 Release 0.1.2

Features added

- readthedocs support

Bugs fixed

- Model relationship attributes were causing documentation issues. Changed a few imports to work around the issue.

Documentation

- Added basic Sphinx documentation.

5.2 Installation

Install the package with pip:

```
$ pip install mqlalchemy
```

Install from source:

```
$ python setup.py install
```

5.3 Support

File an issue on github, or feel free to email me at n.repole+mqlalchemy@gmail.com.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search