aiohttp-sqlalchemy

Release 0.34.0

Ruslan Ilyasovich Gilfanov

Dec 11, 2021

CONTENTS

1 Installation 3

2 Quickstart 5

3 Advansed usage 9

4 Reference 13

5 Releases 19

Index 29

SQLAlchemy 1.4 / 2.0 support for AIOHTTP.

The library provides the next features:

- initializing asynchronous sessions through a middlewares;
- initializing asynchronous sessions through a decorators;
- simple access to one asynchronous session by default key;
- preventing attributes from being expired after commit by default;
- support different types of request handlers;
- support nested applications.

CHAPTER

ONE

INSTALLATION

Installing `aiohttp-sqlalchemy` with pip:

```
pip install aiohttp-sqlalchemy
```

1.1 Optional requirements

For PostgreSQL support, you also need install `asyncpg`:

```
pip install asyncpg
```

For MySQL support, you also need install `aiomysql`:

```
pip install aiomysql
```

For SQLite3 support, you also need install `aiosqlite`:

```
pip install aiosqlite
```

CHAPTER

TWO

QUICKSTART

2.1 Simple example

Install `aiosqlite` for work with sqlite3:

```
pip install aiosqlite
```

Copy and paste this code in a file and run:

```
from datetime import datetime

import sqlalchemy as sa
from aiohttp import web
from sqlalchemy import orm

import aiohttp_sqlalchemy as ahsa

metadata = sa.MetaData()
Base = orm.declarative_base(metadata=metadata)


class MyModel(Base):
    __tablename__ = 'my_table'

    pk = sa.Column(sa.Integer, primary_key=True)
    timestamp = sa.Column(sa.DateTime(), default=datetime.now)


async def main(request):
    sa_session = ahsa.get_session(request)

    async with sa_session.begin():
        sa_session.add(MyModel())
        result = await sa_session.execute(sa.select(MyModel))
        result = result.scalars()

    data = {
        instance.pk: instance.timestamp.isoformat()
        for instance in result
    }
    return web.json_response(data)
```

(continues on next page)

(continued from previous page)

```
async def app_factory():
    app = web.Application()

    ahsa.setup(app, [
        ahsa.bind('sqlite+aiosqlite:///'),
    ])
    await ahsa.init_db(app, metadata)

    app.add_routes([web.get('/', main)])
    return app


if __name__ == '__main__':
    web.run_app(app_factory())
```

2.2 SQLAlchemy and Asyncio

See Asynchronous I/O (asyncio) section in SQLAlchemy 1.4 documentation.

2.3 More control in configuration

```
import aiohttp_sqlalchemy as ahsa
from sqlalchemy import orm

url = 'sqlite+aiosqlite:///'
engine = create_async_engine(url, echo=True)
Session = orm.sessionmaker(main_engine, AsyncSession, expire_on_commit=False)

ahsa.setup(app, [
    ahsa.bind(Session),
])
```

2.4 Class based views

```
from aiohttp import web
import aiohttp_sqlalchemy as ahsa


class MyClassBasedView(web.View, ahsa.SAMixin):
    async def get(self):
        db_session = self.get_sa_session()

        async with db_session.begin():
```

(continues on next page)

(continued from previous page)

```
            # some your code


ahsa.setup(app, [
    ahsa.bind(MainSession),
])
app.add_routes([web.view('/', MyClassBasedView)])
```

CHAPTER

THREE

ADVANSED USAGE

3.1 Multiple database backends per session

See Partitioning Strategies (e.g. multiple database backends per Session) section in SQLAlchemy 1.4 documentation.

3.2 Multiple session factories in application

```
import aiohttp_sqlalchemy as ahsa

postgresql_url = 'postgresql+asyncpg://user:password@host/database'
mysql_url = 'mysql+aiomysql://user:password@host/database'
sqlite_url = 'sqlite+aiosqlite:///path/to/file.sqlite3'

ahsa.setup(app, [
    ahsa.bind(postgresql_url),
    ahsa.bind(mysql_url, 'sa_second'),
    ahsa.bind(sqlite_url, 'sa_third'),
])
```

3.3 Decorating handlers

Warning: For use a some session factory in decorators, you must set a `middleware` argument to `False` in `bind()` call. Else will raise an exception `DuplicateRequestKeyError`.

If access to one or more databases is needed only in some request handlers, then you can use a `sa_decorator(key)`. For example:

```
import aiohttp_sqlalchemy as ahsa

@ahsa.sa_decorator('sa_specific')
async def specific_handler(request):
    specific_db_session = ahsa.get_session(request, 'sa_specific')

    async with specific_db_session.begin():
        # some your code
```

(continues on next page)

(continued from previous page)

```
ahsa.setup(app, [
    ahsa.bind(specific_db_url, 'sa_specific', middleware=False),
])
```

You can combine the use of decorators with the use of middlewares. For example:

```
import aiohttp_sqlalchemy as ahsa

async def simple_handler(request):
    main_db_session = ahsa.get_session(request)

    async with main_db_session.begin():
        # some your code


@ahsa.sa_decorator('sa_specific')
async def specific_handler(request):
    main_db_session = ahsa.get_session(request)
    specific_db_session = ahsa.get_session(request, 'sa_specific')

    async with main_db_session.begin():
        # some your code

    async with specific_db_session.begin():
        # some your code


ahsa.setup(app, [
    ahsa.bind(main_db_url),
    ahsa.bind(specific_db_url, 'sa_specific', middleware=False),
])
app.add_routes([
    web.get('/simple', simple_handler),
    web.get('/specific', specific_handler),
])
```

You can apply `sa_decorator(key)` with class based views. For example:

```
from aiohttp import web
import aiohttp_sqlalchemy as ahsa


SPECIFIC_DB_KEY = 'sa_specific'
SPECIFIC_DB_URL = 'sqlite+aiosqlite:///'


class SpecificHandler(web.View, ahsa.SAMixin):
    @property
    def specific_session(self):
        return self.get_sa_session(SPECIFIC_DB_KEY)
```

(continues on next page)

(continued from previous page)

```
    @ahsa.sa_decorator(SPECIFIC_DB_KEY)
    async def get(self):
        async with self.specific_session.begin():
            # some your code

    @ahsa.sa_decorator(SPECIFIC_DB_KEY)
    async def post(self):
        async with self.specific_session.begin():
            # some your code


ahsa.setup(app, [
    ahsa.bind(SPECIFIC_DB_URL, SPECIFIC_DB_KEY, middleware=False),
])
app.add_routes([web.view('/', SpecificHandler)])
```

CHAPTER

FOUR

REFERENCE

4.1 Main user functionality

`aiohttp_sqlalchemy.setup(app, binds)`

Setup function for SQLAlchemy binding to AIOHTTP application.

Parameters

- **app** (*aiohttp.web_app.Application*) – your AIOHTTP application.
- **binds** (*Iterable[Tuple[sqlalchemy.orm.session.sessionmaker, str, bool]]*) – iterable of *aiohttp_sqlalchemy.bind()* calls.

Return type None

`aiohttp_sqlalchemy.bind(target, key='sa_main', *, middleware=True)`

Function wrapper for binding.

Parameters

- **target** (*Union[str, sqlalchemy.ext.asyncio.engine.AsyncEngine, sqlalchemy.orm.session.sessionmaker]*) – argument can be database connection url, asynchronous engine or asynchronous session factory.
- **key** (*str*) – key of SQLAlchemy binding.
- **middleware** (*bool*) – *bool* for enable middleware. True by default.

Return type Tuple[sqlalchemy.orm.session.sessionmaker, str, bool]

`async aiohttp_sqlalchemy.init_db(app, metadata, key='sa_main')`

Create all tables, indexes and etc.

Parameters

- **app** (*aiohttp.web_app.Application*) – your AIOHTTP application.
- **metadata** (*sqlalchemy.sql.schema.MetaData*) – ...
- **key** (*str*) – key of SQLAlchemy binding.

Return type None

`aiohttp_sqlalchemy.get_session(request, key='sa_main')`

Return *AsyncSession* instance.

Parameters

- **request** (*aiohttp.web_request.Request*) – AIOHTTP request object.
- **key** (*str*) – key of SQLAlchemy binding.

Return type sqlalchemy.ext.asyncio.session.AsyncSession

4.2 Class based views

Warning: The API of class based views is experimental and unstable.

class aiohttp_sqlalchemy.**SAMixin**(*request*)

Bases: aiohttp_things.web_handlers.ContextMixin

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: **None**

Request instance.

class aiohttp_sqlalchemy.**SAModelMixin**(*request*)

Bases: *aiohttp_sqlalchemy.web_handlers.SAMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

4.2.1 Instance mixins

class aiohttp_sqlalchemy.**PrimaryKeyMixin**(*request*)

Bases: aiohttp_things.web_handlers.PrimaryKeyMixin, *aiohttp_sqlalchemy.web_handlers.SAModelMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

class **UnitAddMixin**(*request*)

Bases: *aiohttp_sqlalchemy.web_handlers.SAModelMixin*, aiohttp_things.web_handlers.ItemMixin

property UnitAddMixin.**request:** **None**

Request instance.

property request: **None**

Request instance.

class aiohttp_sqlalchemy.**UnitDeleteMixin**(*request*)

Bases: aiohttp_sqlalchemy.web_handlers.DeleteStatementMixin, *aiohttp_sqlalchemy.web_handlers.PrimaryKeyMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: **None**

Request instance.

class aiohttp_sqlalchemy.**UnitEditMixin**(*request*)

Bases: aiohttp_things.web_handlers.ItemMixin, aiohttp_sqlalchemy.web_handlers.UpdateStatementMixin, *aiohttp_sqlalchemy.web_handlers.PrimaryKeyMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

class aiohttp_sqlalchemy.**UnitViewMixin**(*request*)
Bases: aiohttp_things.web_handlers.ItemMixin, aiohttp_sqlalchemy.web_handlers.SelectStatementMixin, *aiohttp_sqlalchemy.web_handlers.PrimaryKeyMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

4.2.2 List mixins

class aiohttp_sqlalchemy.**OffsetPaginationMixin**(*request*)
Bases: aiohttp_things.web_handlers.PaginationMixin, aiohttp_sqlalchemy.web_handlers.SelectStatementMixin

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

page_key_adapter
alias of int

property request: None
Request instance.

class aiohttp_sqlalchemy.**ListAddMixin**(*request*)
Bases: aiohttp_things.web_handlers.ListMixin, *aiohttp_sqlalchemy.web_handlers.SAModelMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

class aiohttp_sqlalchemy.**ListDeleteMixin**(*request*)
Bases: aiohttp_things.web_handlers.ListMixin, aiohttp_sqlalchemy.web_handlers.DeleteStatementMixin

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

class aiohttp_sqlalchemy.**ListEditMixin**(*request*)
Bases: aiohttp_things.web_handlers.ListMixin, aiohttp_sqlalchemy.web_handlers.UpdateStatementMixin

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

class aiohttp_sqlalchemy.**ListViewMixin**(*request*)
Bases: aiohttp_things.web_handlers.ListMixin, aiohttp_sqlalchemy.web_handlers.SelectStatementMixin

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

property request: None
Request instance.

4.2.3 Views

class aiohttp_sqlalchemy.**SABaseView**(*request*)
Bases: aiohttp.web_urldispatcher.View, *aiohttp_sqlalchemy.web_handlers.SAMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

class aiohttp_sqlalchemy.**SAModelView**(*request*)
Bases: aiohttp.web_urldispatcher.View, *aiohttp_sqlalchemy.web_handlers.SAModelMixin*

Parameters **request** (*aiohttp.web_request.Request*) –

Return type None

4.3 Additional functionality

aiohttp_sqlalchemy.**sa_decorator**(*key='sa_main'*)
SQLAlchemy asynchronous handler decorator.

Parameters **key** (*str*) – key of SQLAlchemy binding.

Return type Callable[[...], Callable[[...], Awaitable[aiohttp.web_response.StreamResponse]]]

aiohttp_sqlalchemy.**sa_middleware**(*key='sa_main'*)
SQLAlchemy asynchronous middleware factory.

Parameters **key** (*str*) – key of SQLAlchemy binding. Has default.

Return type Callable[[...], Awaitable[aiohttp.web_response.StreamResponse]]

async aiohttp_sqlalchemy.**get_engine**(*app*, *key='sa_main'*)
Return *AsyncEngine* instance.

Parameters

- **app** (*aiohttp.web_app.Application*) – your AIOHTTP application.
- **key** (*str*) – key of SQLAlchemy binding.

Return type sqlalchemy.ext.asyncio.engine.AsyncEngine

aiohttp_sqlalchemy.**get_session_factory**(*source*, *key='sa_main'*)
Return callable object which returns an *AsyncSession* instance.

Parameters

- **source** (*Union[aiohttp.web_request.Request, aiohttp.web_app.Application]*) – AIOHTTP request object or your AIOHTTP application.
- **key** (*str*) – key of SQLAlchemy binding.

Return type sqlalchemy.orm.session.sessionmaker

CHAPTER

FIVE

RELEASES

5.1 Version 0.34

Added

- Add official support Python 3.10;
- Add `get_engine` util.

Changed

- Argument `target` in signature `bind(target ,key=SA_DEFAULT_KEY, *, middleware=True)` can be of the following types `str`, `sqlalchemy.ext.asyncio.AsyncEngine`, `sqlalchemy.orm.sessionmaker` or their compatible descendants. Custom session factory is no longer supported.

Removed

- Removed `views` synonym for `web_handlers`;
- Removed `ItemAddMixin` synonym for `UnitAddMixin`;
- Removed `ItemDeleteMixin` synonym for `UnitDeleteMixin`;
- Removed `ItemEditMixin` synonym for `UnitEditMixin`;
- Removed `ItemViewMixin` synonym for `UnitViewMixin`;
- Removed `OffsetPagination` synonym for `OffsetPaginationMixin`;
- Removed `SAItemAddMixin` synonym for `UnitAddMixin`;
- Removed `SAItemDeleteMixin` synonym for `UnitDeleteMixin`;
- Removed `SAItemEditMixin` synonym for `UnitEditMixin`;
- Removed `SAItemViewMixin` synonym for `UnitViewMixin`;
- Removed `SAListAddMixin'` synonym for `ListAddMixin`;
- Removed `SAListDeleteMixin` synonym for `ListDeleteMixin`;
- Removed `SAListEditMixin` synonym for `ListEditMixin`;
- Removed `SAListViewMixin` synonym for `ListViewMixin`;
- Removed `SAModelDeleteMixin` synonym for `DeleteStatementMixin`;
- Removed `SAModelEditMixin` synonym for `UpdateStatementMixin`;
- Removed `SAModelViewMixin` synonym for `SelectStatementMixin`;
- Removed `SAPrimaryKeyMixin` synonym for `PrimaryKeyMixin`.

5.2 Version 0.33

- `ItemAddMixin` is deprecated. Use `UnitAddMixin`;
- `ItemDeleteMixin` is deprecated. Use `UnitDeleteMixin`;
- `ItemEditMixin` is deprecated. Use `UnitEditMixin`;
- `ItemViewMixin` is deprecated. Use `UnitViewMixin`.

5.3 Version 0.32

- Added `prepare_context()` method for `OffsetPaginationMixin`.

5.4 Version 0.31

Added

- Added default `page_key` and `page_key_adapter` attributes for `OffsetPaginationMixin`;
- Added `execute_select_stmt()` method for `OffsetPaginationMixin`.

Deprecated

- `OffsetPagination` is deprecated. Use `OffsetPaginationMixin`;
- `SAModelDeleteMixin` is deprecated. Use `DeleteStatementMixin`;
- `SAModelEditMixin` is deprecated. Use `UpdateStatementMixin`;
- `SAModelViewMixin` is deprecated. Use `SelectStatementMixin`.

5.5 Version 0.30

Deprecated

- `views` module is deprecated. Use import from `aiohttp_sqlalchemy` or `aiohttp_sqlalchemy.web_handlers`;
- `SAItemAddMixin` synonym is deprecated. Use `ItemAddMixin` class;
- `SAItemDeleteMixin` is deprecated. Use `ItemDeleteMixin` class;
- `SAItemEditMixin` synonym is deprecated. Use `ItemEditMixin` class;
- `SAItemViewMixin` synonym is deprecated. Use `ItemViewMixin` class;
- `SAListAddMixin` synonym is deprecated. Use `ListAddMixin` class;
- `SAListDeleteMixin` synonym is deprecated. Use `ListDeleteMixin` class;
- `SAListEditMixin` synonym is deprecated. Use `ListEditMixin` class;
- `SAListViewMixin` synonym is deprecated. Use `ListViewMixin` class;
- `SAPrimaryKeyMixin` synonym is deprecated. Use `PrimaryKeyMixin` class.

5.6 Version 0.29

Added

- `OffsetPagination`.

Changed

- `SAListViewMixin` class is no longer inherited from `aiohttp_things.PaginationMixin`.

5.7 Version 0.28

Changed

- Renamed `handlers` module to `web_handlers`;
- Renamed `sa_session` method to `get_sa_session` in `SAMixin`;
- Renamed `get_sa_delete_stmt` method to `get_delete_stmt` in `SAModelDeleteMixin` and `ItemDeleteMixin`;
- Renamed `get_sa_update_stmt` method to `get_update_stmt` in `SAModelEditMixin` and `ItemEditMixin`;
- Renamed `get_sa_view_stmt` method to `get_select_stmt` in `SAModelViewMixin` and `ItemViewMixin`.

5.8 Version 0.27

Added

- `SAItemAddMixin` as a synonym for `ItemAddMixin`;
- `SAItemDeleteMixin` as a synonym for `ItemDeleteMixin`;
- `SAItemEditMixin` as a synonym for `ItemEditMixin`;
- `SAItemViewMixin` as a synonym for `ItemViewMixin`;
- `SAListAddMixin` as a synonym for `ListAddMixin`;
- `SAListDeleteMixin` as a synonym for `ListDeleteMixin`;
- `SAListEditMixin` as a synonym for `ListEditMixin`;
- `SAListViewMixin` as a synonym for `ListViewMixin`;
- `SAPrimaryKeyMixin` as a synonym for `PrimaryKeyMixin`.

Changed

- `views` module renamed to `handlers`;
- classes from `handlers` temporarily imported to empty `views` module for backward compatibility;
- `SAItemAddMixin` renamed to `ItemAddMixin`;
- `SAItemDeleteMixin` renamed to `ItemDeleteMixin`;
- `SAItemEditMixin` renamed to `ItemEditMixin`;
- `SAItemViewMixin` renamed to `ItemViewMixin`;
- `SAListAddMixin` renamed to `ListAddMixin`;

- `SAListDeleteMixin` renamed to `ListDeleteMixin`;
- `SAListEditMixin` renamed to `ListEditMixin`;
- `SAListViewMixin` renamed to `ListViewMixin`;
- `SAPrimaryKeyMixin` renamed to `PrimaryKeyMixin`.

Removed

- `SAView` a synonym for `SAModelView`.

5.9 Version 0.26

Changed

- `SAListViewMixin` inherited by `aiohttp_things.PaginationMixin`.

Removed

- `SAListMixin` class removed, use `aiohttp_things.ListMixin`;
- `SAItemMixin` class removed, use `aiohttp_things.ItemMixin`.

5.10 Version 0.25

Changed

- Attribute `instance` renamed to `item` in `SAItemAddMixin`, `SAItemEditMixin`, `SAItemViewMixin`.

Removed

- `SAInstanceMixin` removed, use `SAItemMixin`.

5.11 Version 0.24

Removed

- `SAInstanceView` class;
- `SAItemView` synonym for `SAInstanceView`;
- `SAAbstractView` synonym for `SAMixin`;
- `SAOneModelMixin` synonym for `SAModelMixin`;
- `SAInstanceAddMixin` synonym for `SAItemAddMixin`;
- `SAInstanceDeleteMixin` synonym for `SAItemDeleteMixin`;
- `SAInstanceEditMixin` synonym for `SAItemEditMixin`;
- `SAInstanceViewMixin` synonym for `SAItemViewMixin`.

5.12 Version 0.23

Add

- `SAItemMixin` as a separate mixin;
- `SAInstanceAddMixin` as a synonym for `SAItemAddMixin`;
- `SAInstanceDeleteMixin` as a synonym for `SAItemDeleteMixin`;
- `SAInstanceEditMixin` as a synonym for `SAItemEditMixin`;
- `SAInstanceViewMixin` as a synonym for `SAItemViewMixin`;

Changed

- `SAInstanceAddMixin` renamed to `SAItemAddMixin`;
- `SAInstanceDeleteMixin` renamed to `SAItemDeleteMixin`;
- `SAInstanceEditMixin` renamed to `SAItemEditMixin`;
- `SAInstanceViewMixin` renamed to `SAItemViewMixin`;
- `SAItemMixin` is no longer a synonym for `SAInstanceMixin`.

5.13 Version 0.22

Added

- `SAModelDeleteMixin`;
- `SAModelEditMixin`;
- `SAModelViewMixin`;
- `SAPrimaryKeyMixin`;
- `SAInstanceAddMixin`;
- `SAInstanceDeleteMixin`;
- `SAInstanceEditMixin`;
- `SAInstanceViewMixin`;
- `SAListMixin`;
- `SAListAddMixin`;
- `SAListDeleteMixin`;
- `SAListEditMixin`;
- `SAListViewMixin`.

5.14 Version 0.21

Changed

- Rename `SAItemMixin` to `SAInstanceMixin`;
- Rename `SAItemView` to `SAInstanceView`.

Added

- `SAItemMixin` as a synonym for `SAInstanceMixin`;
- `SAItemView` as a synonym for `SAInstanceView`.

5.15 Version 0.20

Added

- Added `SAItemMixin`;
- Added `SAItemView`.

5.16 Version 0.19

Added

- `sa_session` as a synonym for `get_session`;
- `sa_session_factory` as a synonym for `get_session_factory`.

Changed

- Rename `sa_session` to `get_session`;
- Rename `sa_session_factory` to `get_session_factory`.

5.17 Version 0.18.1

Added

- `SAView` as a synonym for `SAModelView`.

Changed

- Rename `SAView` to `SAModelView`.

5.18 Version 0.18

Changed

- First argument of function `aiohttp_sqlalchemy.bind()` renamed from `bind_to` to `target`;
- Type hint alias `TBinding` renamed to `TBind`;
- Type hint alias `TBindings` renamed to `TBinds`;
- Type hint alias `TBindTo` renamed to `TTarget`.

5.19 Version 0.17

Added

- `views.SAAbstractView` as a synonym for `views.SAMixin`;
- `views.SAOneModelMixin` as a synonym for `views.SAModelMixin`;

Changed

- type checks in `aiohttp_sqlalchemy.bind()``including replacing from ``ValueError` to `TypeError`;
- `views.SAAbstractView` renamed `views.SAMixin`;
- `views.SAOneModelMixin` renamed `views.SAModelMixin`.

Removed

- Removed type check of result of call session factory.

5.20 Version 0.16

Added

- Added utility `sa_session_factory(source, key = SA_DEFAULT_KEY)`, when `source` can be instance of `aiohttp.web.Request` or `aiohttp.web.Application`.

5.21 Version 0.15.4

Changed

- Changed `DEFAULT_KEY` from deprecated to synonym.

5.22 Version 0.15

Added

- Added synonym `bind` for `sa_bind`;
- Added synonym `init_db` for `sa_init_db`.

5.23 Version 0.14

Added

- Added utility `sa_init_db(app, metadata, key = SA_DEFAULT_KEY)`;
- Added constant `SA_DEFAULT_KEY` instead `DEFAULT_KEY`.

Deprecated

- `DEFAULT_KEY` is deprecated. Use `SA_DEFAULT_KEY`.

5.24 Version 0.13

Changed

- Argument `expire_on_commit` of `sessionmaker` set to `False` by default.

5.25 Version 0.12

Added

- Added `sa_session_key` attribute in `SAAbstractView` class;
- Added support url and `AssyncEngine` instance as first argument in `sa_bind()`.

Changed

- Rename first argument from `factory` to `bind_to` in `sa_bind()` signature.

5.26 Version 0.11

Added

- Added `sa_session(request, key='sa_main')` utility.

5.27 Version 0.10

Added

- Added support Python 3.7.

5.28 Version 0.9

Added

- Support of organized handlers in class added to `sa_decorator(key)`.

Removed

- Removed support of `AsyncEngine` type in `sa_bind()` signature. Use `sessionmaker(engine, AsyncSession)` or custom session factory returning `AsyncSession` instance.

5.29 Version 0.8

Changed

- Rename first argument from `arg` to `factory` in `sa_bind()` signature.

Deprecated

- `AsyncEngine` type is deprecated in `sa_bind()` signature. Use `sessionmaker(engine, AsyncSession)` or custom session factory returning `AsyncSession` instance.

5.30 Version 0.7

Changed

- Usage `sqlalchemy.orm.sessionmaker` instance is recomended as a first argument for `aiohttp_sqlalchemy.sa_bind()` signature. See examples in documetation.

Removed

- Removed support of `request.config_dict.get('sa_main')` and `request.app['sa_main']` expressions. Use a `request['sa_main'].bind` expression.

5.31 Version 0.6

Added

- Add support `sqlalchemy.orm.sessionmaker` as a first argument in function `sa_bind(arg, key, middleware)`.

Changed

- Argument `engine: AsyncEngine` changed to `arg: Union[AsyncEngine, sessionmaker]` in `sa_bind()` signature.

Deprecated

- Deprecated support of `request.config_dict.get('sa_main')` and `request.app['sa_main']` expressions. Use a `request['sa_main'].bind` expression.

Removed

- Deprecated class `views.SAViewMixin` is removed. Use `views.SAAbstractView`;
- Deprecated attribute `SAView.sa_main_session` is removed. Use method `SAView.sa_session(key: str = 'sa_main')`.

5.32 Version 0.5

Removed

- Deprecated function `aiohttp_sqlalchemy.sa_engine()` is removed. Use `aiohttp_sqlalchemy.sa_bind()`.

Deprecated

- Undocumented class `views.SAViewMixin` is deprecated. Use `views.SAAbstractView`.

5.33 Version 0.4

Added

- `SAView.sa_session(key: str = 'sa_main')` function is added instead `SAView.sa_main_session`.

Deprecated

- `SAView.sa_main_session` is deprecated. Use `SAView.sa_session(key: str = 'sa_main')`.

5.34 Version 0.3

Added

- `aiohttp_sqlalchemy.sa_bind()` function is added instead `aiohttp_sqlalchemy.sa_engine()`.

Deprecated

- `aiohttp_sqlalchemy.sa_engine()` function is deprecated. Use `aiohttp_sqlalchemy.sa_bind()`.

INDEX

B

bind() (*in module aiohttp_sqlalchemy*), 13

G

get_engine() (*in module aiohttp_sqlalchemy*), 16
get_session() (*in module aiohttp_sqlalchemy*), 13
get_session_factory() (*in module aiohttp_sqlalchemy*), 16

I

init_db() (*in module aiohttp_sqlalchemy*), 13

L

ListAddMixin (*class in aiohttp_sqlalchemy*), 15
ListDeleteMixin (*class in aiohttp_sqlalchemy*), 15
ListEditMixin (*class in aiohttp_sqlalchemy*), 15
ListViewMixin (*class in aiohttp_sqlalchemy*), 16

O

OffsetPaginationMixin (*class in aiohttp_sqlalchemy*), 15

P

page_key_adapter (*aiohttp_sqlalchemy.OffsetPaginationMixin attribute*), 15
PrimaryKeyMixin (*class in aiohttp_sqlalchemy*), 14
PrimaryKeyMixin.UnitAddMixin (*class in aiohttp_sqlalchemy*), 14

R

request (*aiohttp_sqlalchemy.ListAddMixin property*), 15
request (*aiohttp_sqlalchemy.ListDeleteMixin property*), 15
request (*aiohttp_sqlalchemy.ListEditMixin property*), 15
request (*aiohttp_sqlalchemy.ListViewMixin property*), 16
request (*aiohttp_sqlalchemy.OffsetPaginationMixin property*), 15
request (*aiohttp_sqlalchemy.PrimaryKeyMixin property*), 14
request (*aiohttp_sqlalchemy.PrimaryKeyMixin.UnitAddMixin.UnitAddMi property*), 14
request (*aiohttp_sqlalchemy.SAMixin property*), 14
request (*aiohttp_sqlalchemy.UnitDeleteMixin property*), 14
request (*aiohttp_sqlalchemy.UnitEditMixin property*), 15
request (*aiohttp_sqlalchemy.UnitViewMixin property*), 15

S

sa_decorator() (*in module aiohttp_sqlalchemy*), 16
sa_middleware() (*in module aiohttp_sqlalchemy*), 16
SABaseView (*class in aiohttp_sqlalchemy*), 16
SAMixin (*class in aiohttp_sqlalchemy*), 14
SAModelMixin (*class in aiohttp_sqlalchemy*), 14
SAModelView (*class in aiohttp_sqlalchemy*), 16
setup() (*in module aiohttp_sqlalchemy*), 13

U

UnitDeleteMixin (*class in aiohttp_sqlalchemy*), 14
UnitEditMixin (*class in aiohttp_sqlalchemy*), 14
UnitViewMixin (*class in aiohttp_sqlalchemy*), 15