# msaSDK Module

`.db.crud._sqlmodel`

## Attributes

### __version__ `module-attribute`

```
__version__ = '0.1.0'
```

### sql_operator_map `module-attribute`

```
sql_operator_map: Dict[str, str] = {
    "=": "__eq__",
    "<=": "__le__",
    "<": "__lt__",
    ">": "__gt__",
    ">=": "__ge__",
    "!": "__ne__",
    "!=": "__ne__",
    "<>": "__ne__",
    "*": "in_",
    "!*": "not_in",
    "~": "like",
    "!~": "not_like",
    "-": "between",
}
```

### sql_operator_pattern `module-attribute`

```
sql_operator_pattern: Pattern = re.compile(
    "^\\[(=|<=|<|>|>=|!|!=|<>|\\*|!\\*|~|!~|-)]"
)
```

## Classes

### MSASQLModelCrud

Bases: `MSABaseCrud` , `MSASQLModelSelector`

**Attributes**

## create_fields `class-attribute`

```
create_fields: List[SQLModelField] = []
```

## db `instance-attribute`

```
db = (
    AsyncDatabase(self.engine)
    if isinstance(self.engine, AsyncEngine)
    else Database(self.engine)
)
```

## engine `instance-attribute`

```
engine = engine or self.engine
```

## list_filter `class-attribute`

```
list_filter: List[SQLModelListField] = []
```

## readonly_fields `class-attribute`

```
readonly_fields: List[SQLModelListField] = []
```

## update_fields `class-attribute`

```
update_fields: List[SQLModelListField] = []
```

**Functions**

## __init__

```
__init__(
    model: Type[SQLModel],
    engine: Union[Engine, AsyncEngine],
    fields: List[SQLModelListField] = None,
    router: APIRouter = None,
) -> None
```

### on_create_pre `async`

```
on_create_pre(
    request: Request, obj: BaseModel, **kwargs
) -> Dict[str, Any]
```

### on_filter_pre `async`

```
on_filter_pre(
    request: Request, obj: BaseModel, **kwargs
) -> Dict[str, Any]
```

### on_update_pre `async`

```
on_update_pre(
    request: Request,
    obj: BaseModel,
    item_id: Union[List[str], List[int]],
    **kwargs
) -> Dict[str, Any]
```

### route_create `property`

```
route_create() -> Callable
```

### route_delete `property`

```
route_delete() -> Callable
```

### route_list `property`

```
route_list() -> Callable
```

### route_read `property`

```
route_read() -> Callable
```

### route_update `property`

```
route_update() -> Callable
```

### schema_name_prefix `property`

```
schema_name_prefix()
```

# MSASQLModelSelector

SQLModel Selector

| PARAMETER | DESCRIPTION | |
|-----------|-------------|---|
| `model` | The SQLModel to use<br>**TYPE:** `Type[SQLModel]` | **DEFAULT:** `None` |
| `fields` | List of the SQlModelListFields<br>**TYPE:** `List[SQLModelListField]` | **DEFAULT:** `None` |

**Attributes**

## exclude `class-attribute`

```
exclude: List[SQLModelListField] = []
```

Excluded fields list. A list of fields to exclude from the current model.

Supports current SQLModel model fields, current model database table field names Default: []

## fields `instance-attribute`

```
fields = list(
    filter(
        lambda x: x
        not in self.parser.filter_insfield(self.exclude),
        self.parser.filter_insfield(self.fields),
    )
)
```

## link_models `class-attribute`

```
link_models: Dict[
    str, Tuple[Type[Table], Column, Column]
] = {}
```

Dictionary of link models. More complex, detailed parsing to be done.

## model `instance-attribute`

```
model = model or self.model
```

## ordering `class-attribute`

```
ordering: List[
    Union[SQLModelListField, UnaryExpression]
] = []
```

> List of fields sorted by selector. Default: []

## parser `instance-attribute`

```
parser = MSASQLModelFieldParser(self.model)
```

## pk `instance-attribute`

```
pk: InstrumentedAttribute = self.model.__dict__[
    self.pk_name
]
```

## pk_name `instance-attribute`

```
pk_name: str = (
    self.pk_name
    or self.model.__table__.primary_key.columns.keys()[0]
)
```

### Functions

### __init__

```
__init__(
    model: Type[SQLModel] = None,
    fields: List[SQLModelListField] = None,
) -> None
```

### calc_filter_clause

```
calc_filter_clause(
    data: Dict[str, Any]
) -> List[BinaryExpression]
```

### get_link_clause `async`

```
get_link_clause(
    request: Request,
    link_model: str = None,
    link_item_id: Union[int, str] = Query(
        None,
        title="pk",
        example="1,2,3",
        description="Link Model Primary key or list of primary keys",
    ),
) -> Optional[Any]
```

### get_select `async`

```
get_select(request: Request) -> Select
```

## Functions

---

Last update: September 13, 2022

Created: September 13, 2022