u2d_msa_sdk Module

`.db.crud.parser`

Attributes

SQLModelField `module-attribute`

```
SQLModelField = Union[str, InstrumentedAttribute]
```

SQLModelListField `module-attribute`

```
SQLModelListField = Union[Type[SQLModel], SQLModelField]
```

__version__ `module-attribute`

```
__version__ = '0.0.3'
```

Classes

MSASQLModelFieldParser

Attributes

default_model `instance-attribute`

```
default_model = default_model
```

Functions

__init__

```
__init__(default_model: Type[SQLModel])
```

conv_row_to_dict

```
conv_row_to_dict(
    rows: Union[Row, List[Row]]
) -> Union[None, Dict[str, Any], List[Dict[str, Any]]]
```

sqlalchemy row to dict

filter_insfield

```
filter_insfield(
    fields: Iterable[Union[SQLModelListField, Any]],
    save_class: Tuple[type] = None,
) -> List[Union[InstrumentedAttribute, Any]]
```

get_alias

```
get_alias(
    field: Union[Column, SQLModelField, Label]
) -> str
```

get_column

```
get_column(field: SQLModelField) -> Optional[Column]
```

sqlalchemy Column

get_insfield

```
get_insfield(
    field: SQLModelField,
) -> Optional[InstrumentedAttribute]
```

get_modelfield

```
get_modelfield(
    field: Union[ModelField, SQLModelField],
    deepcopy: bool = False,
) -> Optional[ModelField]
```

pydantic ModelField

get_name

```
get_name(field: InstrumentedAttribute) -> str
```

get_row_keys

```
get_row_keys(row: Row) -> List[str]
```

> sqlalchemy row keys

get_select_keys

```
get_select_keys(stmt: Select) -> List[str]
```

> sqlalchemy select keys

get_sqlmodel_insfield

```
get_sqlmodel_insfield(
    model: Type[SQLModel],
) -> List[InstrumentedAttribute]
```

Functions

get_python_type_parse cached

```
get_python_type_parse(
    field: Union[InstrumentedAttribute, Column]
) -> Callable
```

> This function returns a cached instance of the Python Type Callable object.
>
> > ✏️ **Note**   ⌄
> >
> > Caching is used to prevent re-reading the environment every time the object is used.

Last update: September 13, 2022

Created: September 13, 2022