# msaSDK Module

`.admin.frontend.types`

## Attributes

### MSAOptionsNode `module-attribute`

```
MSAOptionsNode = Union[List[Dict[str, Any]], List[str]]
```

### MSAUIExpression `module-attribute`

```
MSAUIExpression = str
```

### MSAUISchemaNode `module-attribute`

```
MSAUISchemaNode = Union[
    MSAUITemplate,
    "MSAUINode",
    List["MSAUINode"],
    Dict[str, Any],
    List[Dict[str, Any]],
]
```

### MSAUITemplate `module-attribute`

```
MSAUITemplate = Union[str, 'MSAUITpl']
```

### MSA_UI_API `module-attribute`

```
MSA_UI_API = Union[str, MSAUIAPI, dict]
```

## Classes

# MSABaseUIApiOut

Bases: `MSABaseUIModel`

MSA UI Api interface output data model

**Attributes**

data `class-attribute`

```
data: dict = None
```

msg `class-attribute`

```
msg: str = ''
```

status `class-attribute`

```
status: int = 0
```

# MSABaseUIModel

Bases: `BaseModel`

Core Pydantic Base UI Model.

**Classes**

**Config**

**Attributes**

extra `class-attribute`

```
extra = Extra.allow
```

json_dumps `class-attribute`

```
json_dumps = json.dumps
```

json_loads `class-attribute`

```
json_loads = json.loads
```

**Functions**

**msa_ui_dict**

```
msa_ui_dict()
```

**msa_ui_json**

```
msa_ui_json()
```

**update_from_dict**

```
update_from_dict(kwargs: Dict[str, Any])
```

**update_from_kwargs**

```
update_from_kwargs(**kwargs)
```

# MSAUIAPI

Bases: `MSABaseUIModel`

Core MSA UI API

**Attributes**

adaptor `class-attribute`

```
adaptor: str = None
```

> Receive adapter, if the interface does not meet the requirements, you can configure an adapter to handle it as msa_ui needs. Also supports Function or string function styles

autoRefresh `class-attribute`

```
autoRefresh: bool = None
```

> Configure whether the interface needs to be automatically refreshed.

cache `class-attribute`

```
cache: int = None
```

Set cache to set the cache time in milliseconds, within the set cache time, the same request will not be repeatedly initiated, but will get the cached request response data.

## data `class-attribute`

```
data: Union[str, dict] = None
```

Data body of the request, supports data mapping

## dataType `class-attribute`

```
dataType: str = None
```

Default is json can be configured as form or form-data.

When the data contains a file, the form-data (multipart/form-data) format is automatically used. The application/x-www-form-urlencoded format when configured as form.

## headers `class-attribute`

```
headers: Dict[str, Any] = None
```

The requested headers

## method `class-attribute`

```
method: str = None
```

`GET` holds request method (get, post, put, delete)

## qsOptions `class-attribute`

```
qsOptions: Union[str, dict] = None
```

Useful when dataType is form or form-data.

Specific parameters, set by default to: { arrayFormat: 'indices', encodeValuesOnly: true }

## replaceData `class-attribute`

```
replaceData: bool = None
```

Whether the returned data replaces the current data, the default is false, i.e., append, set to true is a complete replacement.

## requestAdaptor `class-attribute`

```
requestAdaptor: str = None
```

> send adapter , msa_ui's API configuration, if you can not configure the request structure you
> want, then you can configure the requestAdaptor send adapter

## responseData `class-attribute`

```
responseData: Dict[str, Any] = None
```

> If the data structure returned by the interface does not meet expectations, you can modify it by
> configuring responseData, which also supports data mapping. The data available for mapping is
> the actual data of the interface (the data part returned by the interface), with additional api
> variables. where api.query is the query parameter sent by the interface, and api.body is the raw
> data of the content body sent by the interface.

## responseType `class-attribute`

```
responseType: str = None
```

> Return type, if it is a download it needs to be set to `blob`

## sendOn `class-attribute`

```
sendOn: MSAUIExpression = None
```

> Configure the request conditions

## trackExpression `class-attribute`

```
trackExpression: str = None
```

> Configure track variable expression, when autoRefresh is enabled, the default is the url of the api
> to automatically track variable changes. If you want to monitor variables outside of the url,
> configure traceExpression.

## url `class-attribute`

```
url: MSAUITemplate
```

> Current interface Api address

# MSAUIEvent

Bases: `MSABaseUIModel`

MSA UI Event Pydantic Model

**Attributes**

### actionType `class-attribute`

```
actionType: str = None
```

Action name

### args `class-attribute`

```
args: dict = None
```

Action parameter {key:value}, supports data mapping

### expression `class-attribute`

```
expression: Union[bool, MSAUIExpression] = None
```

Execution condition, not set means default execution

### outputVar `class-attribute`

```
outputVar: str = None
```

Output data variable name

### preventDefault `class-attribute`

```
preventDefault: Union[bool, MSAUIExpression] = None
```

False = prevent event default behavior

### stopPropagation `class-attribute`

```
stopPropagation: Union[bool, MSAUIExpression] = None
```

False = Stop the execution of subsequent actions

# MSAUINode

Bases: `MSABaseUIModel`

UI Node Component Configuration

**Attributes**

hidden `class-attribute`

```
hidden: bool = None
```

> Hide this node

hiddenOn `class-attribute`

```
hiddenOn: MSAUIExpression = None
```

> Condition for Hiding

id `class-attribute`

```
id: str = None
```

> ID of the Node

name `class-attribute`

```
name: str = None
```

> Name of the Node

onEvent `class-attribute`

```
onEvent: dict = None
```

> On MSAUIEvent Handler

type `class-attribute`

```
type: str = None
```

> Component Type

## visible `class-attribute`

```
visible: bool = None
```

Enable Visible/Show

## visibleOn `class-attribute`

```
visibleOn: MSAUIExpression = None
```

Condition for Visibility

# MSAUITpl

Bases: `MSAUINode`

MSAUITpl component

**Attributes**

## className `class-attribute`

```
className: str = None
```

Class name of the outer Dom

## tpl `class-attribute`

```
tpl: str
```

Configuration template

## type `class-attribute`

```
type: str = 'tpl'
```

Specify as MSAUITpl component

Last update: September 13, 2022

Created: September 13, 2022