

# msaSDK Module

## `.msaapi`

---

## Classes

### MSAFastAPI

Bases: `FastAPI`

a MSAFastAPI application is a special FastAPI application It includes justpy UI routing

#### Attributes

STATIC\_DIRECTORY instance-attribute

```
STATIC_DIRECTORY = STATIC_DIRECTORY
```

STATIC\_NAME instance-attribute

```
STATIC_NAME = STATIC_NAME
```

STATIC\_ROUTE instance-attribute

```
STATIC_ROUTE = STATIC_ROUTE
```

WebPage instance-attribute

```
WebPage = WebPage
```

ui\_current\_dir instance-attribute

```
ui_current_dir = os.path.dirname(  
    self.ui_current_module.__file__  
)
```

## ui\_current\_module instance-attribute

```
ui_current_module = sys.modules['justpy.justpy']
```

## Functions

### `__init__`

```
__init__(
    *,
    debug: bool = False,
    routes: Optional[List[BaseRoute]] = None,
    title: str = "FastAPI",
    description: str = "",
    version: str = "0.1.0",
    openapi_url: Optional[str] = "/openapi.json",
    openapi_tags: Optional[List[Dict[str, Any]]] = None,
    servers: Optional[
        List[Dict[str, Union[str, Any]]]
    ] = None,
    dependencies: Optional[Sequence[Depends]] = None,
    default_response_class: Type[Response] = Default(
        JSONResponse
    ),
    docs_url: Optional[str] = "/docs",
    redoc_url: Optional[str] = "/redoc",
    swagger_ui_oauth2_redirect_url: Optional[
        str
    ] = "/docs/oauth2-redirect",
    swagger_ui_init_oauth: Optional[Dict[str, Any]] = None,
    middleware: Optional[Sequence[Middleware]] = None,
    exception_handlers: Optional[
        Dict[
            Union[int, Type[Exception]],
            Callable[
                [Request, Any],
                Coroutine[Any, Any, Response],
            ],
        ],
    ] = None,
    on_startup: Optional[
        Sequence[Callable[[], Any]]
    ] = None,
    on_shutdown: Optional[
        Sequence[Callable[[], Any]]
    ] = None,
    terms_of_service: Optional[str] = None,
    contact: Optional[Dict[str, Union[str, Any]]] = None,
    license_info: Optional[
        Dict[str, Union[str, Any]]
    ] = None,
    openapi_prefix: str = "",
    root_path: str = "",
    root_path_in_servers: bool = True,
    responses: Optional[
```

```

    Dict[Union[int, str], Dict[str, Any]]
] = None,
callbacks: Optional[List[BaseRoute]] = None,
deprecated: Optional[bool] = None,
include_in_schema: bool = True,
swagger_ui_parameters: Optional[Dict[str, Any]] = None,
generate_unique_id_function: Callable[
    [routing.APIRoute], str
] = Default(generate_unique_id),
**extra: Any
) -> None

```

## add\_jproute

```

add_jproute(
    path: str, wpfunc: typing.Callable, name: str = None
)

```

add a route for the given Webpage returning func

PARAMETER	DESCRIPTION
<code>path(str)</code>	the path to use as route
<code>wpfunc(typing.Callable)</code>	a Webpage returning func
<code>name(str)</code>	the name of the route

## get\_page\_for\_func async

```

get_page_for_func(
    request, func: typing.Callable
) -> WebPage

```

get the Webpage for the given func

PARAMETER	DESCRIPTION
<code>request</code>	the request to pass to the given function
<code>func</code>	the function <b>TYPE:</b> <code>typing.Callable</code>

RETURNS	DESCRIPTION
<code>WebPage</code>	the Webpage returned by the given function <b>TYPE:</b> <code>WebPage</code>

### get\_response\_for\_load\_page

```
get_response_for_load_page(request, load_page)
```

get the response for the given webpage

PARAMETER	DESCRIPTION
<code>request(Request)</code>	the request to handle
<code>load_page(WebPage)</code>	the webpage to wrap with justpy and

RETURNS	DESCRIPTION
<code>Reponse</code>	the response for the given load_page

### handle\_session\_cookie

```
handle_session_cookie(
    request,
) -> typing.Union[bool, Response]
```

handle the session cookie for this request

RETURNS	DESCRIPTION
<code>typing.Union[bool, Response]</code>	True if a new cookie and session has been created

### jproute

```
jproute(
    path: str, name: typing.Optional[str] = None
) -> typing.Callable
```

## justpy route decorator

function will we "wrapped" as a response and a route added

PARAMETER	DESCRIPTION
<code>func(typing.Callable)</code>	the function to convert to a reponse

## response

```
response(func: typing.Callable)
```

response decorator converts a function to a response

see also <https://github.com/justpy-org/justpy/issues/532> castAsEndPoint

PARAMETER	DESCRIPTION
<code>func(typing.Callable)</code>	the function (returning a WebPage) to convert to a response

## route\_as\_text

```
route_as_text(route)
```

get a string representation of the given route

## set\_cookie

```
set_cookie(
    request,
    response,
    load_page,
    new_cookie: typing.Union[bool, Response],
)
```

set the cookie\_value

PARAMETER	DESCRIPTION
<code>request</code>	the request
<code>response</code>	the response to be sent

PARAMETER	DESCRIPTION
<code>load_page(WebPage)</code>	the WebPage to handle
<code>new_cookie(bool   Response)</code>	True if there is a new cookie. Or Response if cookie was invalid

## Functions

### handle\_event `async`

```
handle_event(data_dict, com_type = 0, page_event = False)
```

---

Last update: September 16, 2022

Created: September 16, 2022