

# msaSDK Module

## .utils.profiler

### Attributes

`__version__` module-attribute

```
__version__ = '0.1.0'
```

### Classes

## MSAProfilerMiddleware

PyInstrument Profiler as Middleware

Used to create a HTML from the Profiler result if enabled in the MSAServiceDefinition instance.

PARAMETER	DESCRIPTION
<code>msa_app</code>	Optional[Router] = None, Instance of the MSAApp <b>TYPE:</b> <code>Optional[Router]</code> <b>DEFAULT:</b> <code>None</code>
<code>profiler_interval</code>	float = 0.0001, <b>TYPE:</b> <code>float</code> <b>DEFAULT:</b> <code>0.0001</code>
<code>profiler_output_type</code>	str = "html", <code>text</code> or <code>html</code> should be html if Admin Site Profiler Page should be used. <b>TYPE:</b> <code>str</code> <b>DEFAULT:</b> <code>'html'</code>
<code>track_each_request</code>	bool = True, Tracks each single request and profiles it immediatly, if off then profiler creates result while shutdown event. <b>TYPE:</b> <code>bool</code> <b>DEFAULT:</b> <code>True</code>

PARAMETER	DESCRIPTION
<code>**profiler_kwargs</code>	other pyinstrument args like 'html_file_name' <b>TYPE:</b> <code>Dict</code> <b>DEFAULT:</b> <code>{}</code>

## Attributes

`app` `instance-attribute`

```
app = app
```

Linked/Mounted MSAApp Instance

## Functions

`__call__` `async`

```
__call__(
    scope: Scope, receive: Receive, send: Send
) -> None
```

register an event handler for profiler stop

`__init__`

```
__init__(
    app: ASGIApp,
    *,
    msa_app: Optional[Router] = None,
    profiler_interval: float = 0.0001,
    profiler_output_type: str = "html",
    track_each_request: bool = True,
    **profiler_kwargs: Dict
)
```

`get_profiler_result` `async`

```
get_profiler_result()
```

Produces the profiler result in the defined output type format, `text` or `html`

