

# msaJustPyUI Module

## .jpcore.template

Created on 2022-09-02 modified version, original from JustPy @author: wf (modification swelcker)

## Classes

### Context

legacy context handler, encapsulates context

#### Attributes

component\_engine\_type instance-attribute

```
component_engine_type = context_dict["frontend_engine_type"]
```

context\_dict instance-attribute

```
context_dict = context_dict
```

display\_url\_js instance-attribute

```
display_url_js = self.get_js_option('display_url', '')
```

frontend\_engine\_libs instance-attribute

```
frontend_engine_libs = context_dict['frontend_engine_libs']
```

justpy\_dict\_js instance-attribute

```
justpy_dict_js = justpy_dict_js.replace(
    "</" + "script>", '</" + "script>'
)
```

page\_id\_js instance-attribute

```
page_id_js = context_dict['page_id']
```

page\_options instance-attribute

```
page_options = PageOptions(  
    context_dict.get("page_options", {})  
)
```

redirect\_js instance-attribute

```
redirect_js = self.get_js_option('redirect', '')
```

title\_js instance-attribute

```
title_js = self.get_js_option('title', 'JustPy')
```

use\_websockets\_js instance-attribute

```
use_websockets_js = self.context_dict.get(  
    "use_websockets", "true"  
)
```

### Functions

\_\_init\_\_

```
__init__(context_dict: dict)
```

constructor

PARAMETER	DESCRIPTION
<code>context_dict</code>	a context dict in legacy format <b>TYPE:</b> <code>dict</code>

as\_html\_lines

```
as_html_lines(indent: str = ' ')
```

generate the html lines for justpy to work

### as\_javascript\_constructor

```
as_javascript_constructor(indent: str = '    ')
```

generate my initial JavaScript

### as\_javascript\_setup

```
as_javascript_setup(indent)
```

generate the java script setup code

### as\_script\_src

```
as_script_src(  
    file_name: str, indent: str = "    ", subdir=""  
)
```

### as\_script\_srcs

```
as_script_srcs(indent: str = '    ')
```

generate a list of javascript files to be imported

### get\_js\_option

```
get_js_option(key, default_value)
```

get the page\_option with the given key as javascript using the default value in case the value is not set or none

### get\_url\_for

```
get_url_for(name: str, path: str = None)
```

get url for the given route name

PARAMETER	DESCRIPTION
name	name of the route <b>TYPE:</b> str

PARAMETER	DESCRIPTION
<code>path</code>	path params <b>TYPE:</b> <code>str</code> <b>DEFAULT:</b> <code>None</code>

## PageOptions

legacy page\_options handler, encapsulating page\_options

### Attributes

events instance-attribute

```
events: List[str] = page_options_dict.get('events', [])
```

page\_options\_dict instance-attribute

```
page_options_dict = page_options_dict
```

### Functions

#### `__init__`

```
__init__(page_options_dict: dict)
```

constructor

#### `get_debug`

```
get_debug()
```

Get debug status of the PageOptions

#### `get_page_ready`

```
get_page_ready()
```

#### `get_reload_interval_ms`

```
get_reload_interval_ms() -> float
```

## get\_result\_ready

```
get_result_ready()
```

---

Last update: September 28, 2022

Created: September 28, 2022