msaJustPyUI Module

`.jpcore.justpy_app`

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

Attributes

FRONTEND_ENGINE_LIBS module-attribute

```
FRONTEND_ENGINE_LIBS = [
    fn[:-3]
    for fn in os.listdir(lib_dir)
    if fnmatch.fnmatch(fn, "*.js")
]
```

TEMPLATES_DIRECTORY module-attribute

```
TEMPLATES_DIRECTORY = config(
    "TEMPLATES_DIRECTORY", cast=str, default=template_dir
)
```

component_file_list module-attribute

```
component_file_list = create_component_file_list()
```

cookie_signer module-attribute

```
cookie_signer = Signer(str(SECRET_KEY))
```

grand_parent module-attribute

```
grand_parent = pathlib.Path(
    __file__
).parent.parent.resolve()
```

lib_dir module-attribute

```
lib_dir = os.path.join(
    template_dir, "js", FRONTEND_ENGINE_TYPE
)
```

template_dir module-attribute

```
template_dir = f'{grand_parent}/justpy/templates'
```

template_options module-attribute

```
template_options = {
    "tailwind": TAILWIND,
    "quasar": QUASAR,
    "quasar_version": QUASAR_VERSION,
    "highcharts": HIGHCHARTS,
    "aggrid": AGGRID,
    "aggrid_enterprise": AGGRID_ENTERPRISE,
    "static_name": STATIC_NAME,
    "component_file_list": component_file_list,
    "no_internet": NO_INTERNET,
    "katex": KATEX,
    "plotly": PLOTLY,
    "bokeh": BOKEH,
    "deckgl": DECKGL,
    "vega": VEGA,
}
```

templates module-attribute

```
templates = Jinja2Templates(directory=TEMPLATES_DIRECTORY)
```

Classes

JustpyAjaxEndpoint

> Bases: `HTTPEndpoint`
>
> Justpy specific HTTPEndpoint/app (ASGI application)
>
> **Functions**
>
> **__init__**

```
__init__(scope, receive, send)
```

constructor

on_disconnect async

```
on_disconnect(page_id)
```

post async

```
post(request: Request)
```

Handles post method. Used in Ajax mode for events when websockets disabled

PARAMETER	DESCRIPTION
`request`	the request to handle **TYPE:** `Request`

JustpyApp

Bases: `Starlette`

a justpy application is a special FastAPI/Starlette Addition/MixIn

Attributes

app instance-attribute

```
app = self
```

Functions

__init__

```
__init__() -> None
```

add_jproute

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

add a route for the given Webpage returning func

PARAMETER	DESCRIPTION
`path`	the path to use as route **TYPE:** `str`
`wpfunc`	a Webpage returning func **TYPE:** `typing.Callable`
`name`	the name of the route **TYPE:** `str` **DEFAULT:** `None`

get_page_for_func async

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

get the Webpage for the given func

PARAMETER	DESCRIPTION
`request`	the request to pass to the given function
`func`	the function **TYPE:** `typing.Callable`

RETURNS	DESCRIPTION
`WebPage`	the Webpage returned by the given function **TYPE:** `WebPage`

get_response_for_load_page

```
get_response_for_load_page(
    request: Request, load_page: WebPage
) -> HTMLResponse
```

get the response for the given webpage

PARAMETER	DESCRIPTION
`request`	the request to handle **TYPE:** `Request`
`load_page(WebPage)`	the webpage to wrap with justpy and

RETURNS	DESCRIPTION
`response`	the response for the given load_page **TYPE:** `HTMLResponse`

handle_session_cookie

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

handle the session cookie for this request

RETURNS	DESCRIPTION
`typing.Union[bool, Response]`	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
`func(typing.Callable)`	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
`func`	the function (returning a WebPage) to convert to a response **TYPE:** `typing.Callable`

route_as_text

```
route_as_text(route)
```

get a string representation of the given route

set_cookie

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

set the cookie_value

PARAMETER	DESCRIPTION
`request`	the request **TYPE:** `Request`
`response`	the response to be sent **TYPE:** `Response`
`load_page`	the WebPage to handle **TYPE:** `WebPage`
`new_cookie`	True if there is a new cookie. Or Response if cookie was invalid **TYPE:** `typing.Union[bool, Response]`

RETURNS	DESCRIPTION
`response`	the response object **TYPE:** `Response`

Functions

create_component_file_list

```
create_component_file_list()
```

handle_event async

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

Last update: September 28, 2022

Created: September 28, 2022