

[Przewodnik wdrożeniowca](#) > Konfiguracja Custom Widgets / Dodatku

Wprowadzenie

W systemie eDokumenty istnieje możliwość definiowania własnych przycisków na pasku tzw. pluginów lub custom widgets - tzw. toolbarze.

Tworzenie własnego przycisku

Wszelkie informacje o dodatkach są przechowywane w tabeli `custom_widgets`, która zawiera 9 kolumn. Zanim jednak wprowadzimy dane do tabeli należy przygotować skrypt oraz ikonę dla przycisku, który ten skrypt będzie uruchamiał.

Tworzenie skryptu

Skrypt ma postać pliku z rozszerzeniem `.inc` (np. `Test.inc`). Jest to klasa zapisana przy pomocy języka PHP. Listing przykładowej klasy został umieszczony poniżej:

```
<?php

final class Test {

    public static function init() {

        // wyświetla monit eDokumentowy
        JScript::alert(Translator::translate('Witaj to ja Twój plugin'));

        // wyświetla monit JavaScriptowy alert
        JScript::add('alert(666)');

    }

}

print_r($args);

Test::init();

?>
```

Jak widać powyżej w skrypcie można wykorzystywać także język JavaScript. Jednakże należy mieć na uwadze, że błędny lub szkodliwy skrypt może wpłynąć negatywnie na stabilność systemu !!!

Utworzony skrypt umieszczamy w katalogu `public_html/apps/edokumenty/scripts`

Przygotowanie grafiki/ikony

Ikona musi mieć rozmiar 24x24px format np. png i być umieszczona w katalogu: `public_html/framework/img/toolbarIcons/24x24/`.

Menadżer Custom Widget

Przycisku definiowane są w Panel sterowania > Manager Custom Widget. Gdzie pola:

`parametry = {"script":"Test.inc","image":"24x24Vikona.png"}` - parametry dodatku w formacie JSON, w tym:

- `script` to nazwa pliku z katalogu `public_html/apps/edokumenty/scripts`
- `image` - oznacza ikonę z katalogu `public_html/framework/img/toolbarIcons/24x24/ikona.png`. Nazwa ikony musi być poprzedzona `24x24Vnazwa.rozszerzenie`, w tym przykładzie `24x24Vikona.png`.
- `Typ` = button|JSMenulItem - typ: button = przycisk, JSMenulItem = opcja na menu dotyczy menu dokumentu i sprawy
- `Tekst` - Nazwa na pasku narzędziowym (może być pusta).
- `Pomoc` - Opis, który pokaże się w tooltipie (dymku odpowiedzi)
- `Użycie` - `c_path` - przyjmuje wartości:
 - `contacts/toolbar` dla modułu Klienci

- `adocuments/toolbar` dla modułu Dokumenty.
- `servicemod/toolbar` dla modułu Serwis.
- `custom/id_widgetu` dla Custom Widgetu
- `task_trmtyp(TODO|PHONECALL|MEETING))/toolbar` dla kartoteki zdarzenia o podanym typie (`trmtyp`).
- `vatnote_costs_list_form/toolbar` - lista pozycji na zakładce koszty w fakturze
- `document_products_form_{dctpid}/toolbar` - lista pozycji na danym dokumencie
- `document_{dctpid}/menu` - menu na dokumencie pod przyciskiem Dokument
- `process/menu` - menu na sprawie pod przyciskiem Dodatkowe zadania
- `product/toolbar` - pasek zadań na kartotece produktu
- `custom/5` (gdzie 1 jest równe `cswgid` z tabeli `custom_widgets`, czyli trzeba zobaczyć jaka jest największa wartość i wiemy że jak dodamy to nasz `custom_widget` będzie miał id np. 5) - taka konstrukcja pozwala używać później deklaracji w XML w CustomModule.

Istotą działania modułu dodatkowego jest przekazywanie danych z zaznaczonych elementów pod kluczami

- `contid` dla modułu Klienci
- `doc_id` dla modułu Dokumenty

Wstawienie do pliku xml - CustomModules.xml

```
<buttons>
  <button>
    <custom_widget>
      3
    </custom_widget>
  </button>
</buttons>
```

Uprawnienia

Domyślnie każdy użytkownik ma dostęp do utworzonego w ten sposób przycisku. Ograniczenie dostępu jest realizowane poprzez tabelę `access`, której opis jest zamieszczony w artykule: <http://support.edokumenty.eu/trac/wiki/DeployerGuide/Others/SettingRightsForFields>

Dane do ustawiania uprawnień dla przycisku:

- `clsnam` = `CUSTOM_WIDGET`
- `keyval` = `custom_widgets.cswgid`