

## [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 `24x24/nazwa.rozszerzenie`, w tym przykładzie `24x24Vikona.png`.
- `params` - może określać dodatkowe parametry przekazywane do skryptu, np. `"params": "doc_id:{DOC_ID}"` przekaże id dokumentu
- `Typ` = `button`/`JSMenulem` - `typ: button` = przycisk, `JSMenulem` = opcja na menu dotyczy menu dokumentu i sprawy
- `Tekst` - Nazwa na pasku narzędziowym (może być pusta).
- `Pomoc` - Opis, który pokaże się w tooltipie (dymku podpowiedzi)
- `Użycie` - `c_path` - przyjmuje wartości:

- contacts/toolbar dla modułu Klienci
- employees/toolbar dla modułu Pracownicy
- adocuments/toolbar dla modułu Dokumenty.
- servicemod/toolbar dla modułu Serwis.
- aregisters/toolbar dla modułu dzienniki.
- 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 (produkty) na danym dokumencie
- process\_products\_form/toolbar - lista pozycji (produkty) na sprawie
- document\_{dctpid}/menu - menu na dokumencie pod przyciskiem Dokument
- process/menu - menu na sprawie pod przyciskiem Dodatkowe zadania
- processes/toolbar - menu w module Sprawy
- 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.
- document\_travelcosts\_{dctpid}/toolbar - koszty delegacji
- cregisterview/toolbar:{{cregids}} - pasek zadań dla rejestru konstrukcja ścieżki obejmuje 2 definicje: cregisterview/toolbar - widget pokaże się na każdym rejestrze lub cregisterview/toolbar{{cregids}} - widget pokaże się na wybranych rejestrach. Zmienna {cregids} może być w formie 1 lub 1,2,3 gdzie identyfikator numeryczne to id z cregisters.register

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