

Wikiprint Book

Title: Dodatkowe moduły i zakładki

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Customization/AdditionalTabs

Version: 129

Date: 04/19/26 05:54:21

Table of Contents

<i>Dodatkowe moduły i zakładki</i>	3
<i>Menu</i>	3
<i>Dodatkowe zakładki</i>	3
<i>Ograniczanie widoczności zakładek</i>	4
<i>Dodatkowe moduły</i>	4
<i>Definiowanie dodatkowych modułów</i>	4
<i>Ograniczanie widoczności modułów</i>	8
<i>Filtry</i>	8
<i>Ikona modułu</i>	8

Dodatkowe moduły i zakładki

Menu

1. [Dodatkowe zakładki](#)
 - 1.1 [Ograniczanie widoczności zakładek](#)
2. [Dodatkowe moduły](#)
 - 2.1 [Definiowanie dodatkowych modułów](#)
 - 2.2 [Ograniczanie widoczności modułów](#)
 - 2.3 [Filtry](#)
 - 2.4 [Ikona modułu](#)

Dodatkowe zakładki

System eDokumenty umożliwia dodawanie customowych zakładek do kartoteki

- sprawy
- kontrahenta
- urządzenia
- dokumentu

Zakładki te oparte są na raportach oraz pliku konfiguracyjnym xml znajdującym się w

```
$APP_PATH\var\tpl\tabs
$APP_PATH oznacza /home/edokumenty/public_html/apps/edokumenty
lub
C:\Program files\BetaSoft\eDokumenty\public_html\apps\edokumenty
```

Jeśli katalog var\tpl\tabs jest pusty należy skopiować szablony plików xml z \$APP_PATH\var\tpl_default\tabs.

Nazwy obsługiwanych plików xml

- sprawy - process_tpl.xml
- kontrahenta - contact_tpl.xml
- urządzenia - device_tpl.xml
- dokumentu - document_tpl.xml
- ewidencji - evidence_tpl.xml

Aby dodać dodatkową zakładkę do jednej z wyżej wymienionych kartotek należy utworzyć odpowiedni raport oraz wyedytować wybrany plik xml dla danej kartoteki.

Definicja pliku xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tabs>
  <tab label="" rep_id="">
    <buttons>
      <button>
        <id>
          new
        </id>
        <label>
          Nowa
        </label>
        <dscrpt>
          Nowa
        </dscrpt>
        <onclick>
          App.openDialogByCls({CLSNAM}, {KEYVAL}, ({afterSubmit:'{AFTER_SUBMIT}', mode:'new'}).toJSONString())
        </onclick>
      </button>
    </buttons>
  </tab>
</tabs>
```

```

        <icon>
            new.gif
        </icon>
    </button>
</buttons>
</tab>
</tabs>

```

Zakładka dla dokumentu dodatkowo przyjmuje parametr **dctpid** (ID typu dokumentu) np.

```
<tab label="zakładka 1" rep_id="1" dctpid="1">
```

Zakładka pojawi się tylko w dokumentach o `dctpid = 1`. Brak tego parametru spowoduje dodanie zakładki dla wszystkich typów. Można też podać więcej identyfikatorów typu po przecinku (np. `dctpid="1,3,5,6,9"`).

Przejdź do [Menu](#)

Ograniczanie widoczności zakładek

Widoczność Zakładki może również być ograniczona poprzez parametr `grp_id` (np. `grp_id="2,5,10"`) który ograniczy widoczność zakładki wyłącznie do członków wymienionych po przecinku grup.

Przejdź do [Menu](#)

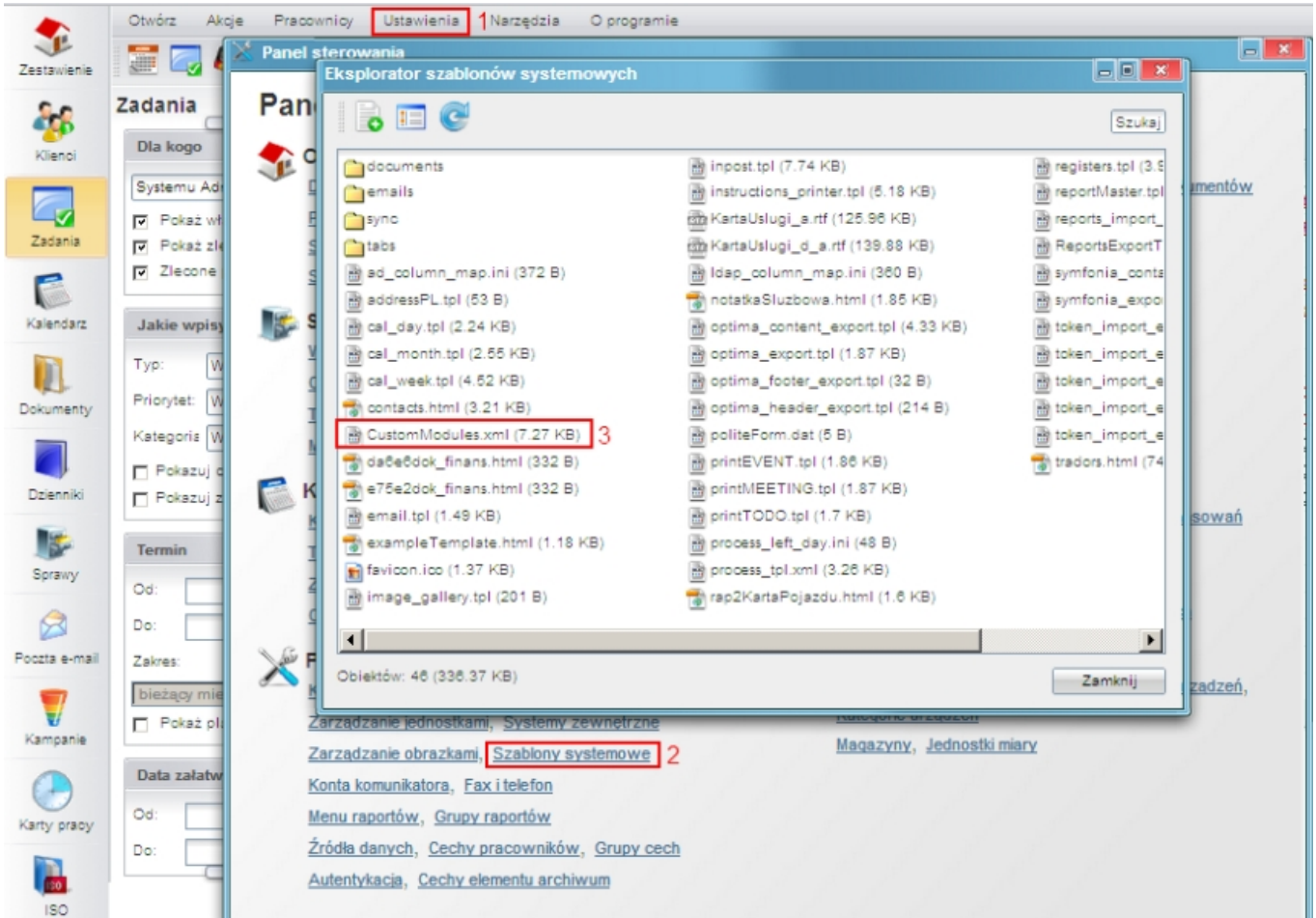
Dodatkowe moduły

System eDokumenty umożliwia tworzenie własnych modułów w oparciu o podobny mechanizm.

Definiowanie dodatkowych modułów

W systemie można również skonfigurować w oparciu o ten sam mechanizm własny moduł. Na wersji demonstracyjnej moduły dostępne przez użytkownika *jmamon* "Delegacje" oraz "Urlopy" dla użytkownika *serwis* są utworzone poprzez utworzenie następującego pliku w katalogu `$APP_PATH/var/tp1/CustomModules.xml`

Edycja tego pliku nie jest konieczna z poziomu konsoli. Dostęp do tego pliku jest możliwy z poziomu edokumentów. W Panelu sterowania szukamy odnośnika Szablony Systemowe. Uruchomi się okienko z plikami *Szablonów Systemowych*



(Szablony systemowe)

W okienku tym wyszukujemy plik "CustomModules.xml", klikamy na niego, a następnie zapisujemy na dysku. Edytujemy zmiany, po czym usuwamy istniejącą na serwerze plik i wgrujemy nową wersję.

```
<?xml version="1.0" encoding="UTF-8"?>
<modules>
  <module id="cModule_1" name="Delegacje" label="Delegacje"
  icon="processes.gif" right="bswfms.custom_module.delegations">
    <toolbar>
    </toolbar>
    <filters>
    </filters>
    <tabs>
    <tab label="Moje delegacje" rep_id="87">
      <buttons>
        <button>
          <id>
            new
          </id>
          <label>
            Nowy
          </label>
          <dscrpt>
            Nowy dokument
          </dscrpt>
          <onclick>
            App.openDialogByCls('DOCUMENT', null,
            ({afterSubmit: '{AFTER_SUBMIT}', dctpid:23,dctptp:'CustomDocument', mode:'new'}).toJSONString())
          </onclick>
        </button>
      </buttons>
    </tab>
  </tabs>
</module>
</modules>
```

```

        </onclick>
        <icon>
            new.gif
        </icon>
    </button>
    <button>
        <id>
            edit
        </id>
        <label>
            Edycja
        </label>
        <dscrpt>
            Edytuj dokument
        </dscrpt>
        <onclick>
            App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit: '{AFTER_SUBMIT}', dctpId:23, dctp: 'CustomDocument', mode: 'edit'}).toJSONString())
        </onclick>
        <icon>
            edit.gif
        </icon>
    </button>
    <button>
        <id>
            delete
        </id>
        <label>
            Usuń
        </label>
        <dscrpt>
            Usuń
        </dscrpt>
        <onclick>
            App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit: '{AFTER_SUBMIT}', dctpId:23, dctp: 'CustomDocument', mode: 'del'}).toJSONString())
        </onclick>
        <icon>
            delete.gif
        </icon>
    </button>
</buttons>
</tab>
</tabs>
</module>

<module id="Urlopy" name="Urlopy" label="Urlopy"
icon="travel.png" right="bswfms.custom_module.vacations">
    <toolbar>
    </toolbar>
    <filters>
    </filters>
    <tabs>
    <tab label="Moje wnioski" rep_id="95">
        <buttons>
            <button>
                <id>
                    new
                </id>
                <label>
                    Nowy

```

```

        </label>
        <dscript>
            Nowy wniosek urlopowy
        </dscript>
        <onclick>
            App.openDialogByCls('DOCUMENT', null,
({afterSubmit: '{AFTER_SUBMIT}', dctpId:21, dctp: 'CustomDocument', mode: 'new'}).toJSONString())
        </onclick>
        <icon>
            new.gif
        </icon>
    </button>
    <button>
        <id>
            edit
        </id>
        <label>
            Edycja
        </label>
        <dscript>
            Edytuj wniosek
        </dscript>
        <onclick>
            App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit: '{AFTER_SUBMIT}', dctpId:21, dctp: 'CustomDocument', mode: 'edit'}).toJSONString())
        </onclick>
        <icon>
            edit.gif
        </icon>
    </button>
    <button>
        <id>
            delete
        </id>
        <label>
            Usuł,,
        </label>
        <dscript>
            Usuł,,
        </dscript>
        <onclick>
            App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit: '{AFTER_SUBMIT}', dctpId:21, dctp: 'CustomDocument', mode: 'del'}).toJSONString())
        </onclick>
        <icon>
            delete.gif
        </icon>
    </button>
</buttons>
</tab>
</tabs>
</module>

</modules>

```

Przyciski mogą też do listy parametrów obsługiwać klucze z bean-ów, dla przykładu:

```

App.createDialog('createProcessForm', 'SimpleProcessCreatingForm', './modules
/APProcesses/forms/SimpleProcessCreatingForm.inc', 'Zakadanie', '513',

```

```
{clsnam: 'DOSS', strpid: 351, devcid: '{devcid}', contid: '{contid'}}.toJSONString(), null, 'fast')
```

Sprawa automatycznie otrzyma atrybuty id urządzenia oraz id kontrahenta urządzenia. Na razie zaimplementowano w urządzeniu.

Przejdź do [Menu](#)

Ograniczanie widoczności modułów

Prawa dostępu do modułów (inaczej widoczność modułów) są rozwiązane w bardziej skomplikowany sposób. Do tworzonego modułu dodatkowego należy dodać do tabeli **right_def** definicje prawa (UWAGA !!! jeżeli tego prawa nie ma w tej tabeli) *bswfms.custom_module*, a potem szczegółowe prawo do zakładki np. do zakładki *Delegacje* może to być: *bswfms.custom_module.delegations*. W ww tabeli należy wypełnić następujące pola

- *prn_id* - (dla *bswfms.custom_module* będzie to prawdopodobnie 1, a do *bswfms.custom_module.delegations* wartość pola *def_id* rekordu *bswfms.custom_module* - Musi być zapewniona struktura drzewiasta!)
- *define* - nazwa prawa np. *bswfms.custom_module.delegations*
- *group_* - ustawić należy SYSTEM
- *commen* - nazwa prawa wyświetlana w systemie. np.: *Delegations: delegacje*.

Następnym krokiem w procesie jest ustawienie praw dla modułów w pliku *CustomModules.xml* w definicji właściwości *rights* modułu

```
...
<module id="cModule_1" name="Delegacje" label="Delegacje"
icon="processes.gif" right="bswfms.custom_module.delegations">
...

```

gdzie wstawiamy wartość pola *define* danego prawa.

Po zdefiniowaniu praw w tabeli oraz w pliku *CustomModules.xml* należy przejść do menu *Pracownicy*, a następnie wybrać *Grupy*. Jeżeli danej grupy nie ma, to tworzymy ją. Po zapisaniu grupy pojawiają się nowe zakładki. Przechodzimy do zakładki *Pracownicy* gdzie wprowadzamy wybranych przez nas użytkowników, a następnie klikamy w zakładkę *Prawa do systemu*, gdzie ze struktury drzewiastej praw wybieramy odpowiednie prawo i przypisujemy dostęp do prawa.

Przejdź do [Menu](#)

Filtry

Na chwilę obecną jest dostępny tylko jeden filtr

```
<filters>
  <filter name="Daty" type="MonthSelectorTreeFilter" height="auto" opened="0">
  </filter>
</filters>
```

Przejdź do [Menu](#)

Ikona modułu

Ikony w formacie .gif bądź .png o maksymalnym rozmiarze 36px x 36px wrzucamy do katalogu *public_html/framework/img/PageToolBar*.

Przejdź do [Menu](#)