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>
```

```
                <icon>
                    new.gif
                </icon>
            </button>
            <button>
                <custom_widget>
                    3
                </custom_widget>
            </button>
        </buttons>
    </tab>
</tabs>
```

Od wersji 4.1-alfa51 istnieje możliwość zadeklarowania przyciski jako custom widget (http://support.edokumenty.eu/trac/wiki/DeployerGuide/Others/CustomWidgets). Tag button musi zawierać informację w postaci taga custom_widget o wartości id danego widgeta czyli kolumna cswgid. Tabela custom_widget powinna zawierać rekord zgodnie z definicją w dokumentacji a wartość w kolumnie c_path to

```
c_path = custom/cswgid czyli custom/3
```

gdzie 3 to identyfikator rekordu. Id zaznaczonych rekordów są przekazywane pod parametrem keyval.

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

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

lub

```
 <tab label="zakladka 1" iframe="http://www.onet.pl"></tab>
```

Gdzie:

- rep_id - identyfikator raportu z systemu eDokumenty
- iframe - url do strony, którą chcemy wyświetlić w eDokumentach w zakładce

Dodatkowo do taga onclick oraz atrybutu iframe można dodawać parametry z postaci danych z aktualnie otwartej kartoteki. Dla przykładu niech posłuży kartoteka produktu oraz dodatkowa zakładka w postaci iframe wyświetlającego katalog zdjęć:

```
<tab label="Zdjęcia" iframe="http://www.moj.serwer/produkty/?symbol={symbol}"></tab>
```

Token {symbol} zostanie zamieniony na symbol produktu z aktualnie otwartej kartoteki.

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").

Zakładka dla sprawy dodatkowo przyjmuje parametr **dos_id** (ID teczki z wykazu akt) np.

```
 <tab label="zakladka 1" rep_id="1" dos_id="1">
```

Zakładka pojawi się tylko w sprawach o dos_id = 1. Brak tego parametru spowoduje dodanie zakładki dla wszystkich spraw. Można też podać więcej identyfikatorów typu po przecinku (np. dos_id="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

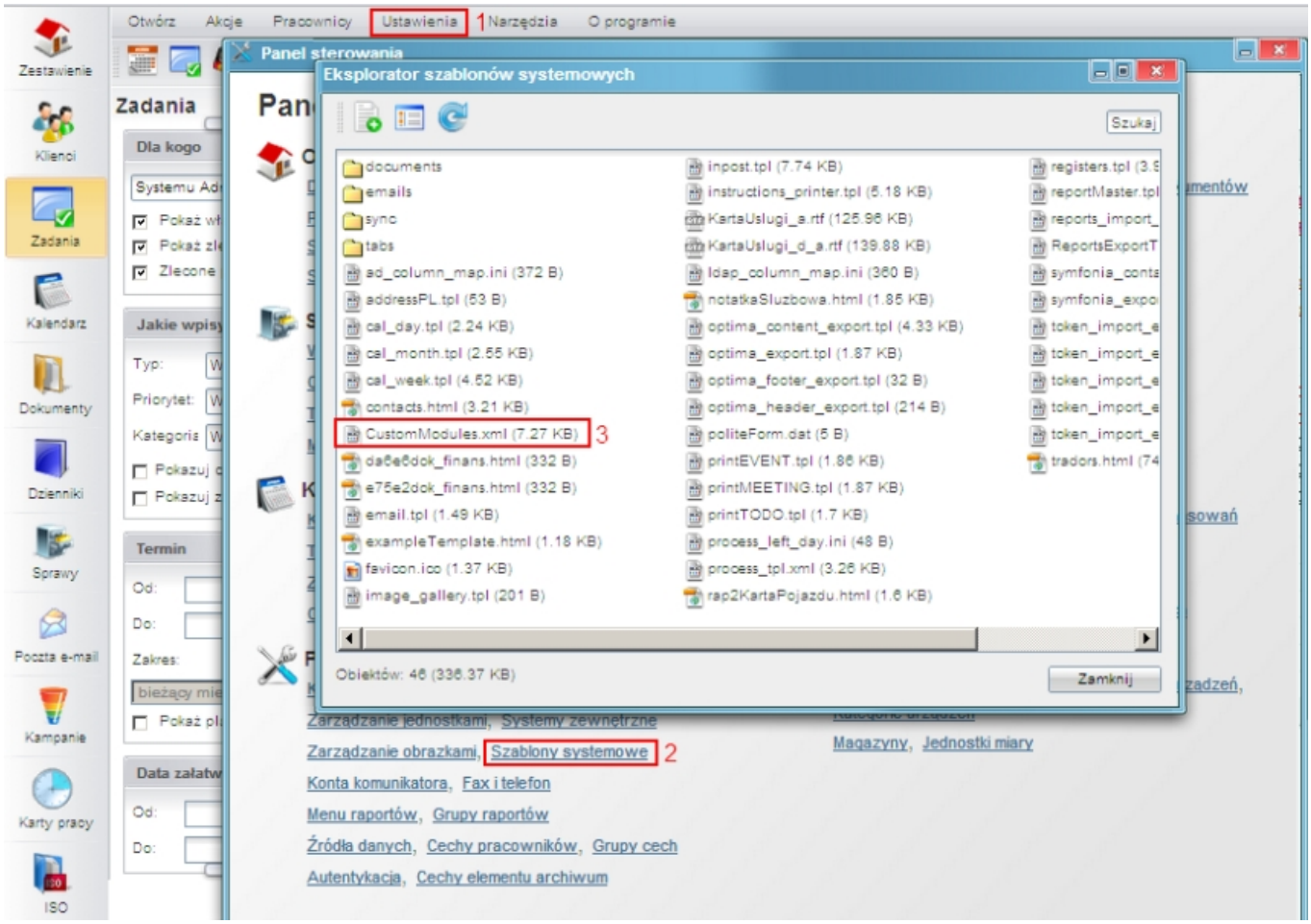
Dodatkowe moduły

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/tpl/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 istniejący na serwerze plik i wgrywamy 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>
               <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,dctptp:'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,dctptp:'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>
               <dscrpt>
                   Nowy wniosek urlopowy
               </dscrpt>
               <onclick>
                   App.openDialogByCls('DOCUMENT', null,
({afterSubmit:'{AFTER_SUBMIT}',dctpid:21,dctptp:'CustomDocument', mode:'new'}).toJSONString())
               </onclick>
               <icon>
                   new.gif
               </icon>
           </button>
           <button>
               <id>
                   edit
               </id>
               <label>
                   Edycja
               </label>
               <dscrpt>
                   Edytuj wniosek
               </dscrpt>
               <onclick>
                   App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit:'{AFTER_SUBMIT}',dctpid:21,dctptp:'CustomDocument', mode:'edit'}).toJSONString())
               </onclick>
               <icon>
                   edit.gif
               </icon>
           </button>
           <button>
               <id>
                   delete
               </id>
               <label>
                   UsuĹ„
               </label>
               <dscrpt>
                   UsuĹ„
               </dscrpt>
               <onclick>
                   App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit:'{AFTER_SUBMIT}',dctpid:21,dctptp:'CustomDocument',mode:'del'}).toJSONString())
               </onclick>
               <icon>
                   delete.gif
```

```
            </icon>
          </button>
        </buttons>
    </tab>
        </tabs>
    </module>


</modules>
```

Przycisk tworzący sprawę

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

```
App.createDialog('createProcessForm','SimpleProcessCreatingForm','./modules
/AProcesses/forms/SimpleProcessCreatingForm.inc','Zakadanie','513',
({clsnam:'DOSS',strpid:351,devcid:'{devcid}',contid:'{contid}'}).toJSONString(), null, 'fast')
```

Dodatkowo od wersji 4.0 obsługiwany jest parametr {KEYVALS}, który odpowiada ze przekazanie do dialoga wszystkich zaznaczonych elemenetów z listy z nie tylko pierwszego jak to ma miejsce w przypadku parametru {KEYVAL}.

Parametr strpid jest dostępny w **Ustawienia -> Panel sterowania -> Sprawy -> Wyciąg z wykazu akt -> Kolumna Miejsce (strpid)** W przypadku jeśli kolumna ta nie jest włączona należy kliknąć na ikonę "Widoczne kolumny" (na dole listy) i zaznaczyć ptaszka.

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 w/w 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 dostepny tylko jeden filtr, którego można zdefiniować bezpośrednio z poziomu xml:

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

Dodatkowe filtry są realizowane za pomocą definiowanych filtrów dla raportów zgodnie z dokumentacją:

Filtry dla raportów

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