# Wikiprint Book

Title: Przewodnik użytkownika > Dodatkowe moduły i zakładki

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

Version: 129

Date: 07/28/25 03:37:44

# **Table of Contents**

Przewodnik użytkownika > Dodatkowe moduły i zakładki	3
Menu	4
Dodatkowe zakładki	4
Dodatkowe tokeny do użycia w tagu onclick:	5
Widoczność zakładki	5
Wymuszenie otwarcia jako domyślna zakładka	5
Ustawienie na odpowiedniej pozycji	6
Domyślnie schowany panel z filtrami	6
Ograniczanie widoczności zakładek	6
Dodatkowe moduły	6
Definiowanie dodatkowych modułów	6
Otwieranie modułu w nowym oknie	10
Przycisk tworzący sprawę	10
Przekazywanie parametrów	11
Przyciski operujące na rejestrze	11
Ograniczanie widoczności modułów	12
Filtry	12
Filtry statyczne	13
Ikona modułu / kolor	13
Wywołanie raportu	13
Wykonywanie etapów workflow z przycisków	14
Moduł operujący na istniejących obiektach	14
Wersja	15
1.5	15
1.4	16
1.3	17
1.2	17
1.1	17
1.0	17
0.8	18
0.7	19

# Przewodnik użytkownika > Dodatkowe moduły i zakładki

Informacje o wprowadzenie funkcjonalności:

Wersja systemu	Wersja modułu/funkcji	Data kompilacji	Zmiany	Opis
6.22.0	<u>1.5</u>	-	zmiana	Obsługa tokena {REFRESH_FILTERS} oraz parametru js_refresh_filters do customowych przycisków
6.19.0	<u>1.4</u>	-	zmiana	Obsługa unikalnych ID dla tab CustomModules
6.0.274	<u>1.3</u>	-	zmiana	Obsługa dodatkowego atrybutu tagu tab enabled, explain_how_to_enabled oraz visible
6.0.229	<u>1.2</u>	-	zmiana	Obsługa dodatkowego atrybutu tagu module "always_show_tabs"
6.0.229	<u>1.1</u>	-	zmiana	Obsługa menu dokumentów
6.0.113	<u>1.0</u>	-	zmiana	Obsługa custom_widget w menu
6.0.52(Ready_), 5.4.10	0.9	zmiana	18-05-2018	Każdy Custom moduł może być trzymany w osobnym pliku
5.2.61	0.6	-	zmiana	Dodano token {EXPORT_TO_PRINTER} za pomocą którego można wywołać systemowy dialog drukowania
6.0 (Ready_)	<u>0.8</u>	-	zmiana	Obsługa menu
5.2.33	0.7	02.11.2017	zmiana	Obsługa pionowego panelu zakładek
5.2.12	0.5	29.08.2017	zmiana	Dodano nowy token {EXPORT_TO_XLS} za pomocą którego można eksportować raport do XLS
5.0	0.4	18.11.2015	Zmiana	Dodano filtr (globalny oraz dla raportu) Projektu oraz obsługę parametru {PROJID}

4.8.100, 4.10.16, 4.11.2	0.3	18.11.2015	Zmiana	Dodano możliwość definicji separatora dla przycisków jako tag <separator></separator>
-----------------------------	-----	------------	--------	--

#### 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
- pracownik user\_tpl.xml
- produkt product\_tpl.xml
- profil użytkownika user\_profile\_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



# Dodatkowe tokeny do użycia w tagu onclick:

Token	Opis
{CLSNAM}	klasa zaznaczonego elementu na liście jeśli zostało podane pole w raporcie
{KEYVAL}	identyfikator zaznaczonego elementu na liście jeśli zostało podane pole w raporcie
{KEYVALS}	identyfikatory zaznaczonych elementów - w przypadku dokumentów: {doc_id:{KEYVALS}
{AFTER_SUBMIT}	akcja odświeżenia widoku po edycji wpisu lub np. jako dodatkowa akcji na pasku zadań
{MODE}	wartość pobierana z taga <id></id> używać jako new, edit, del, delete

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.

Od wersji 4.3.23 obsługiwany jest atrybut def\_tab dla taga tab. Oznacza on, że dana zakładka ma być domyślnie otwierana. Działa tylko dla customowych modułów. W przypadku customowych zakładek dla kartotek ten atrybut nie działa.

## Widoczność zakładki

Dodatkowo od wersji 4.6.4, 4.7.2 dla zakładki możemy zdefiniować czy ma ona się pokazać domyślnie na pasku zakładek czy po kliknięciu "więcej...". Dokonujemy tego poprzez dodanie atrybutu alwaysVisible dla taga tab podobnie jak def\_tab.

# Wymuszenie otwarcia jako domyślna zakładka

Zdarza się, że chcemy ustawić dodatkową zakładkę jako domyślnie otwartą w danej kartotece podczas jej otwierania. Standardowo ustawienie, która zakładka ma się otworzyć jest przypisane w kartotece na stałe. Zachowanie można to zmienić i wymusić otwarcie dodatkowej zakładki poprzez atrybut forceOpen="1". Atrybut ten można ustawić tylko dla 1 zakładki.

# Ustawienie na odpowiedniej pozycji

Domyślnie zakładki są dodawanie w kolejności wpisania ich w definicji xml. Zachowanie to można zmienić i ustawić zakładki w dowolnej kolejności (nawet pomiędzy standardowymi zakładkami) poprzez atrybut prior (od 0 do n gdzie n <= liczbie zakładek). Atrybut ten w połączeniu z forceOpen umożliwia nam dodanie nowej domyślnie otwartej zakładki na pierwszej pozycji w kartotece.

Od wersji 6.0.7

### Domyślnie schowany panel z filtrami

Aby polepszyć wrażenia estetyczne i dać większe obszar pracy dla użytkownika można domyślnie zwinąć panel z filtrami. Służy te tego atrybut taga <tab> showFilters="0".

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

Od wersji 6.0.52 jest możliwość definiowania pojedynczego modułu w jednym pliku w katalogu \$APP\_PATH/var/cfg/CustomModules/{module\_id}.xml

Gdzie wartość {module\_id} zamieniamy na wartość atrybutu id w elemencie o nazwie module. Listing takiego pliku wygląda dokładnie tak samo jak 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* 

Otwórz	Pracownicy	Ustawienia	<b>1</b> Narzędzia	O programie			
Zestav		_		Panel sterow	ania		_ <b>0</b> ×
	Panel s	terowa		Eksplora	tor szablonó	w systemowych	_ <b>□</b> ×
<u>ين</u>			6	5			Szukaj
	Da	ne podmiotu	bsconne 📁	ect	Ľ	circlegraph.tpl (10.6 KB)	files_previev
	Pro	ocedury, Sta	e docume	nts	Ľ	contact_summary.tpl (37.35 KB)	graph.tpl (4
	Fo	ldery wyszuk	📒 emails			contacts.html (3.21 KB)	🗋 image_gall¢
	Ek	ran logowani	📒 market		Ľ	custom_list_fields.xml_default (874 B)	🗋 inpost.tpl (i
			📒 sync			CustomMapService.json (2 B)	instructions
	- sh	orawy	📒 tabs			CustomModules.xml (63 B) 3	📴 KartaUslugi
	Ka	tegorie spraw	ad_colum	mn_map.ini (372 B)		document_history_adds.tpl (0 B)	📑 KartaUslugi
_	Dr	oiekty, Mask	addressP	PL.tpl (63 B)	Ľ	einvoices_tabs.ini (259 B)	ldap_colum
	Sz	ablony projel	archive_l	ist.tpl (2.14 KB)	Ľ	email.tpl (2.42 KB)	login_page.
	-			940.tpl (703 B)		example_ext_list.html (550 B)	🗋 login_page.
	T <sub>21</sub> Ka	alendarz i	🗋 cal_day.t	pl (2.24 KB)	Ľ	exampleTemplate.html (1.18 KB)	notatkaSluz
	Ka	tegorie zdarz		th.tpl (2.57 KB)	Ľ	extresp_html.tpl (8.24 KB)	offer_pdf_t
	Тур	py usług, Po	Cal_week	tpl (5.34 KB)	4	favicon.ico (1.37 KB)	🗋 optima_cor
	Za	soby, Cechy					
	Ce	chy zdarzeń,	<				>
	Po	ozostałe u	Obiektów: 69	(449.34 KB)			Zamknij
	🖉 👅 Ka	nały RSS, Por	tlety, Zarządz	anie ISO, Fax i telefon		Wagazyn	
	Za	rządzanie jedn	nostkami, Gru	py cech		Grupy produktów, Cechy produktó	ów, Cechy urządzeń,
	Za	rządzanie obra	azkami <mark>,</mark> Szabl	kami, Szablony systemowe 2 Kategorie urządzeń, Magazyny, Jer			
_	Ka źw	nta komunika ( dia daavab	tora, Menu ra	aportów, Grupy raportów		Warunki handlowe, Rabaty, Cenni	ki
	20	odła danych,	archiwum Co	nikow, Autentykacja		Narzedzia administracyjn	e
	Ce	rtvfikat pieczec	ci elektroniczn	ei. Listv dystrybucvine.		Odblokowywanie kont, Integralnos	ć repozvtorium.
	Ka	nta pocztowe	współdzielone	e, Definicje uprawnień,	Integralność systemu, Backup baz	y danych	
	Mechanizm CustomWidg						-
	Po	dgląd zdarzeń	i systemowych	1			
	W	yszukiwarka, O	Cechy struktuŋ	y organizacyjnej			
(Szablonv :	svstemowe)						

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"?>
<module id="cModule_U" name="Modul" label="Mój modul" icon="fixdefect-48.png" color="#c0392b" right="bswfms.processes" fil
       <toolbar>
       </toolbar>
      <filters>
       </filters>
<tabs>
  <tab label="AKTYWNE" rep_id="1">
       <buttons>
      </buttons>
  </tab>
  <tab label="WSZYSTKIE" rep_id="2">
  <buttons>
      <button>
         <id>refresh26</id>
         <label>Odśwież</label>
         <dscrpt>Odśwież</dscrpt>
         <onclick>
         {AFTER_SUBMIT}
         </onclick>
         <icon>sync-24.png</icon>
         </button>
     </buttons>
  </tab>
 </tabs>
</module>
```

Atrybut filtersMode dostępny od wersji 5.1.4, 5.0.184. Za pomocą tego atrybutu możemy zmieniać położenie panela filtrów. Standardowo panel filtrów widoczny jest po lewej stronie (wartość atrybutu filtersMode = left nie trzeba go podawać to jest domyślna wartość). Można to zmienić poprzez wpisanie do atrybutu filtersMode wartości right\_abs. Panel filtrów pojawi się z prawej strony i będzie on nachodził na listę umieszczoną w module.

```
<?xml version="1.0" encoding="UTF-8"?>
<modules>
 <module id="Urlopy" name="Urlopy" label="Urlopy"
icon="travel.png" right="bswfms.custom_modules.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.png</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.png</icon>
           </button>
```

```
<button>
               <id>delete</id>
               <label>Usuń</label>
               <dscrpt>Usuń</dscrpt>
               <onclick>
                   App.openDialogByCls('DOCUMENT', {KEYVAL},
({afterSubmit:'{AFTER_SUBMIT}',dctpid:21,dctptp:'CustomDocument',mode:'del'}).toJSONString())
               </onclick>
               <icon>delete.png</icon>
           </button>
<separator/>
         <button>
           <id>report</id>
           <label/>
           <dscrpt>Raport</dscrpt>
           <onclick>window.open('litescript.php/demo/?script=GenerateReport&amp;ent_id=2&amp;ent_name=Demo S.A.&amp;rep_id
           <icon> reports.png</icon>
         </button>
         <button>
           <id>printer</id>
           <label/>
           <dscrpt>OD WERSJI 5.2</dscrpt>
           <onclick>{EXPORT_T0_PRINTER}</onclick>
           <icon>print.png</icon>
        </button>
         <button>
           <id>xls</id>
           <label/>
           <dscrpt>EKSPORT D0 XLS D0 WERSJI 5.0 WŁĄCZNIE</dscrpt>
           <onclick> window.open('litescript.php/demo/?script=GenerateReport&amp;ent_id=2&amp;ent_name=Demo S.A.&amp;rep_i
           <icon>xls.png</icon>
         </button>
         <button>
           <id>xls</id>
           <label/>
           <dscrpt>EKSPORT D0 XLS OD WERSJI 5.2</dscrpt>
           <onclick>{EXPORT_T0_XLS}</onclick>
           <icon>xls.png</icon>
         </button>
         <button>
           <id>pdf</id>
           <label/>
           <dscrpt>EKSPORT D0 PDF D0 WERSJI 5.0 WŁĄCZNIE</dscrpt>
           <onclick> window.open('litescript.php/demo/?script=GenerateReport&amp;ent_id=2&amp;ent_name=Demo S.A.&amp;rep_i
           <icon> pdf.png</icon>
         </button>
         <button>
           <id>pdf</id>
           <label/>
           <dscrpt>EKSPORT D0 PDF 0D WERSJI 5.2</dscrpt>
           <onclick>{EXPORT_T0_PDF}</onclick>
           <icon> pdf.png</icon>
         </button>
       </buttons>
  </tab>
       </tabs>
   </module>
</modules>
```

Funkcję openDialogByCls() można wywoływać z innymi parametrami zamiast 'DOCUMENT', co spowoduje otwarcie innych typów okien. Poniżej lista najbardziej przydatnych:

- CONTACT Okienko edycji kontaktu (wpisujemy CONTACT\_EDIT)
- PROCESS Okienko edycji sprawy
- RCP Karta pracy
- MEETING Spotkanie
- EVENT Termin
- TODO Zadanie
- PHONECALL Rozmowa telefoniczna
- DEVICE Urządzenie
- PRODUCT Produkt

Przykład zakładania nowego kontaktu (osoba fizyczna):

App.openDialogByCls('CONTACT\_EDIT', null,({afterSubmit:'{AFTER\_SUBMIT}', phyper:'true' }).toJSONString())

Przykład zabezpieczenia w przypadku gdy raport zwróci zero wyników a na zakładce w module będzie dostępny przycisk Edytuj:

var k = {KEYVAL}; if (k) {App.openDialogByCls('DOCUMENT', {KEYVAL},({afterSubmit:'{AFTER\_SUBMIT}',dctpid:24,dctptp:'Custom

var ss={KEYVAL}; if (ss) {App.openDialogByCls('DOCUMENT',ss,({afterSubmit:'App.ADocumentslist.refresh()'}).toJSONString())

W przypadku gdy zaznaczymy dwa dokumenty:

```
var ss={KEYVALS}; if(ss.length==1) {
App.openDialogByCls('DOCUMENT',ss[0],({afterSubmit:'{AFTER_SUBMIT}'}).toJSONString());
}else{
if(ss.length>1) {
NewFrame.showInfo('Proszę zaznaczyć tylko jeden dokument do edycji.');
} else {
NewFrame.showInfo('Proszę zaznaczyć dokument do edycji.');
}
```

Przykład tworzenia nowej umowy z uzupełnionym klientem na podstawie zestawienia (keyval):

### Otwieranie modułu w nowym oknie

Przykład w javascript:

window.open('litescript.php/aps/?script=NewWindowModule&ent\_id=2&modnam=CustomModule&submodule=cExactoris', '\_blank', 'too

### Przycisk tworzący sprawę

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

Przycisk tworzący sprawę - pełna formatka formatka tworząca sprawę

App.openDialogByCls('PROCESS\_EDIT', null, ({afterSubmit:'{AFTER\_SUBMIT}', clsnam:'DOSS', keyval:39, mode:'new' }).toJSONS

keyval przechowuje ID teczki

Przycisk tworzący sprawę - prosta formatka tworząca sprawę

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

Opis parametrów przycisku tworzącego sprawę

- createProcessForm nazwa w kodzie HTML okna
- · SimpleProcessCreatingForm nazwa klasy php'owej
- ./modules/AProcesses/form... ścieżka do tej klasy
- · Sprawa opis na górnej belce formatki (najczęściej jest nadpisywany przez kod klasy)
- 513 rodzaj formatki 513 to pełna z przyciskami na górnej belce inne rodzaje (kody) raczej są przeznaczone do wew. wywołań
- ({clsnam:'DOSS',strpid:1253,keyval:776,mode:'new'}).toJSONString() parametry do formatki w formacie JSON

Ostatnie dwa parametry zawsze są takie same.

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.

# Przekazywanie parametrów

Od wersji 4.4.40, 4.6 oraz 4.5.24 w przypadku przycisku customowego w konfigurowanych zakładkach na kartotekach np. sprawy istnieje możliwość przekazania parametrów z beana sprawy podobnie jak do akcji onClick.

Wymagane jest aby w definicji przycisku w polu Parametry podać odpowiedni klucz - params. Przykład poniżej:

```
"script":"Test.inc",
"image":"24x24\/deadline.png",
"params":"prc_id:'{prc_id}',dsexid:'{dsexid}'"
}
```

W momencie generowania przycisku na zakładce klucze w nawiasach klamrowych zostaną zastąpione na właściwe atrybuty sprawy. Następnie cały ciąg z klucza params zostanie doklejony do wywołania widgeta. Po stronie widgeta w tablicy \$args będą dostępne klucze jak podano w params - w naszym przypadku będzie to prc\_id oraz dsexid.

Przejdź do <u>Menu</u>

### Przyciski operujące na rejestrze

W custom module można również operować na dowolnym rejestrze. Aby edytować rekord raport skojarzony z modułem musi posiadać klucze clsnam, keyval. Wówczas działać będzie przekazywanie wartości {KEVAL}

```
<buttons>
<button>
<id>new</id>
<label>Nowy</label>
<dscrpt>Nowy wpis</dscrpt>
<onclick>
App.openDialogByCls('CREGISTER_ENTRY', null,
```

```
({afterSubmit:'{AFTER_SUBMIT}', mode:'new',cregid:1}).toJSONString())
</onclick>
<icon>new.png</icon>
</button>
<button>
<id>edit</id>
<label>Edycja</label>
<dscrpt>Edytuj wpis</dscrpt>
<onclick>
       App.openDialogByCls('CREGISTER_ENTRY', {KEYVAL},
                                   ({afterSubmit:'{AFTER_SUBMIT}', mode:'edit',cregid:1}).toJSONString())
</onclick>
<icon>edit.png</icon>
</button>
<button>
<id>delete</id>
<label>Usuń</label>
<dscrpt>Usuń</dscrpt>
<onclick>
       App.openDialogByCls('CREGISTER_ENTRY', {KEYVAL},
              ({afterSubmit:'{AFTER_SUBMIT}', mode:'del',cregid:1}).toJSONString())
</onclick>
<icon>delete.png</icon>
</button>
</buttons>
```

#### Ograniczanie widoczności modułów

Prawa dostępu do modułów (inaczej widoczność modułów) należy zdefiniować w Panelu sterowania > Definicje uprawnień. Zaleca się utworzenie prawa bswfms.custom\_modules (Moduły dodatkowe). A następnie dodanie prawa podrzędnego do custom\_modules.

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



#### 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 <u>Menu</u>

### Filtry

Na chwilę obecną dostępne są następujące filtry, które można zdefiniować bezpośrednio z poziomu xml:

- MonthSelectorTreeFilter wybór dat z drzewka (parametry {DATE\_FROM} i {DATE\_TO}); zakres dat dla drzewka wyznaczany jest jako przedział od min(documents.sysdat) do max(documents.adddat) dla modułów customowych oraz zakres min(documents.adddat) do max(documents.sysdat) dla modułu Ewidencja. W celu wygenerowania drzewka na większy okres należy dodać do systemu np. notatkę służbową z datą dodania np. 2016-12-12.
- ClearingUnitsFilter wybór jednostki rozliczeniowej (parametr {ACORID})
- ProjectsFilter wybór projektu (parametr {PROJID})

```
<filters>
<filter name="Daty" type="MonthSelectorTreeFilter" height="auto" opened="0">
</filter>
<filter name="Jednostka rozliczeniowa" type="ClearingUnitsFilter" height="30px" opened="0">
```

```
</filter>
</filters>
```

### <filters>

```
<filter name="Data delegacji" type="MonthSelectorTreeFilter" height="auto" opened="0" startDate="SELECT min(start_) FROM</pre>
</filter>
</filters>
```

Od wersji 4.5.8 następuje zmiana i dochodzą dwa nowe parametry

- {DATE\_FROM\_RANGE}
- {DATE\_TO\_RANGE} ٠

Odpowiedni przechowują dane do swych odpowiedników bez końcówki RANGE. Modyfikacja ta ma na celu umożliwienie korzystanie z panelu filtrów oraz drzewka gdzie oba parametry były wypełniania przez 2 różne komponenty. Teraz jeśli zdefiniujemy parametr {DATE\_FROM} lub {DATE\_TO} to zostanie od pokazany na panelu filtrów i dodatkowo jeśli dodamy do tpl filter wtedy będą dodatkowe tokeny.

#### Filtry statyczne

Filtry statyczne działają zarówno dla modułu jak i pojedynczej zakładki. Definicja takiego filtru jest taka sama jak powyżej z tą różnicą, że nazwa komponentu wskazuje na StaticFilter

```
<filters>
    <filter name="Daty" type="StaticFilter" token="{ANNA}">
        1=2
    </filter>
</filters>
```

Ważne w takim filtrze jest określenie tokena (osadzamy go w raporcie) w tym przypadku token="{ANNA}". Token może mieć dowolną nazwę. Wartość takiego filtra to stała wartość filtrująca. W tym przypadku jest to 1=2 czyli nigdy nic nie pokaże. Jednak można tu wstawić np.

# d.name = 'Pompa'

gdzie d to alias na tabelę z raportu a name to kolumna, która występuje w tabeli. która jest ukryta pod d.

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

#### Filtry dla raportów

Przejdź do Menu

### Ikona modułu / kolor

lkony w formacie .png,.svg o maksymalnym rozmiarze 48px x 48px wrzucamy do katalogu public\_html/framework/img/PageToolBar. Kolor kodowy modułu, który jest też kolorem tła dla ikony, określamy w atrybucie color tagu module. np.:

```
<module id="cModule_U" name="Moduł" label="Mój moduł" icon="icon.png" color="#c0392b" right="bswfms.processes">
. . .
</module>
```

# Wywołanie raportu

Aby otworzyć raport w module customowym i przekazać zaznaczone elementy z listy jako parametry do raportu należy dodać wywołanie

App.openDialogByCls('REPORT\_VIEW', {rep\_id}, ({keyval:{KEYVALS}}).toJSONString())

gdzie {rep\_id} to id raportu.

W raporcie będzie dostępny parametr {KEYVAL}, który będzie przechowywał zaznaczone elementy dlatego dla wielu zaleca się użycie IN ({KEYVAL}).

Przykład wywołania raportu o rep\_id=23 z parametrem FILTER\_STRING do którego zostaną przekazane po przecinku zaznaczone elementy na liście. W raporcie tym należy umieścić w WHERE parametr {FILTER\_STRING}.

```
<tab label="Przydzielone" rep_id="121">
     <buttons>
        <button>
            <custom_widget>1</custom_widget>
        </button>
     <button>
      <id>Do maila</id>
      <label>Tabela</label>
      <dscrpt>Otwiera tabelę HTML</dscrpt>
      <onclick>
            var keyval = {KEYVALS};
            App.openDialogByCls('REPORT_VIEW', 23, ({filter_string:'fkelid IN (' + keyval + ')'}).toJSONString());
      </onclick>
      <icon>table24.png</icon>
    </button>
        <button>
            <custom_widget>7</custom_widget>
        </button>
     </buttons>
</tab>
```

Wykonywanie etapów workflow z przycisków

```
<tab label="WSZYSTKIE" rep_id="1">
<buttons>
<button>
<id>selectFR</id>
  <label>Potwierdź</label>
   <dscrpt>Potwierdza etap workflow</dscrpt>
   <onclick>
   App.openDialogByCls('P_WORKFLOW', null,({afterSubmit:'{AFTER_SUBMIT}',clsnam:'PROCESS',keyval:{KEYVAL},next_ptstid:336,
   </onclick>
   <icon>pass-24.png</icon>
   </button>
 </buttons>
</tab>
<tab label="WSZYSTKIE" rep_id="1">
<buttons>
<button>
<id>selectFR</id>
  <label>Potwierdź</label>
   <dscrpt>Potwierdza etap workflow</dscrpt>
   <onclick>
   App.openDialogByCls('P_WORKFLOW', 345,({afterSubmit:'{AFTER_SUBMIT}',clsnam:'DOCUMENT',keyval:{KEYVAL},ptstid:336}).toJ
   </onclick>
   <icon>pass-24.png</icon>
   </button>
 </buttons>
</tab>
```

# Moduł operujący na istniejących obiektach

W poniższym przykładzie ewidencjonować można sprawy. Należy zmienic: dsexid - to id z wyciągu z wykazu akt, rep\_id - id raportu

```
<module id="cModule_U2" name="NEW" label="New" icon="fixdefect-48.png" right="bswfms.processes">
      <toolbar>
      </toolbar>
      <filters>
      </filters>
<tabs>
  <tab label="AKTYWNE" rep_id="1">
      <buttons>
<button>
    <id>newbug</id>
    <label>Zgłoszenie</label>
    <dscrpt>Nowe zgłoszenie. Sprawa zostanie założona w teczce</dscrpt>
    <onclick>
    App.createDialog('createProcessForm','SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingFor
    </onclick>
    <icon>fix-defect-24.png</icon>
    </button>
    <button>
        <id>refresh20</id>
         <label>Odśwież</label>
         <dscrpt>Odśwież</dscrpt>
         <onclick>
         {AFTER_SUBMIT}
         </onclick>
         <icon>sync-24.png</icon>
         </button>
     </buttons>
  </tab>
  <tab label="WSZYSTKIE" rep_id="2">
  <buttons>
     <button>
         <id>refresh26</id>
         <label>Odśwież</label>
         <dscrpt>Odśwież</dscrpt>
         <onclick>
         {AFTER_SUBMIT}
         </onclick>
        <icon>sync-24.png</icon>
         </button>
    </buttons>
  </tab>
 </tabs>
</module>
```

# Wersja

# 1.5

Wersja wprowadza możliwość odświeżania panelu filtrów (jeśli filtry zostały zdefiniowane w raporcie - nie dotyczy filtrów zdefiniowanych w pliku XML).

Ogólnie sprawa wygląda dość prosto. Dodany został dodatkowy token (aby zachować wsteczną zgodność) o nazwie {REFRESH\_FILTERS}. Dla przykładu niech posłuży definicja przycisku odświeżającego listę

```
<button>
<id>refresh26</id>
<label>Odśwież</label>
<dscrpt>Odśwież</dscrpt>
<onclick>
{AFTER_SUBMIT} {REFRESH_FILTERS}
</onclick>
```

```
<icon>refresh.png</icon>
</button>
```

W tej postaci oprócz odświeżenia listy nastąpi odświeżenie panelu filtrów (tokeny można łączyć).

Drugi przykład wykorzystania to przekazanie funkcji do customowego skryptu - czyli definicja przycisku wygląda np. tak

<button> <custom\_widget> 71 </custom\_widget> </button>

Wtedy w kodzie php obsługując przekazaną tablicę \$args mamy dodatkowy klucz js\_refresh\_filters. Oczywiście wartość tego klucza może być pusta jeśli raport nie ma filtrów.

Przykładowe dane z tablicy \$args

Ar	ray
(	
	<pre>[cswgid] =&gt; 71 (string =&gt; string)</pre>
	[name] => cModule_UcModule_U_PPA (string => string)
	[tab] => cModule_UcModule_U_PPATH80b0babd932952a (string => string)
	<pre>[rep_id] =&gt; 89 (string =&gt; integer)</pre>
	[REPORT_NAME] => REP_cModule_UcModule_U_PPATH80b0babd932952aLi_Bean_89 (string => string)
	[js_after] => App.cModule_UcModule_U_PPATH80b0babd932952aLirefresh(); (string => string)
	[js_filters_refresh] => App.cModule_UcModule_U_PPATH80b0babd932952arefreshFilters(); (string => string)
	[keyval] => array() (string => array)
	[listName] => cModule_UcModule_U_PPATH80b0babd932952aLi (string => string)
)	

1.4

. . .

Dodatkowy atrybut dla taga tab to atrybut id

```
<tab id="moduleName_tab_chart" label="Wykres" rep_id="190">
</tab>
```

Zalecaną konwencją nazewnictwa jest dodanie prefixu jako nazwy modulułu. Obecne zastosowanie tego atrybutu to "ręczne" otwieranie modułu customowego np. z przycisku oraz automatycznie zaznaczenie konkretnej zakładki (obecnie tylko głównej - nie z pionowego paska)

Przykład wykorzystania

```
App.openDialogByCls('APPLICATION', 0, ({name:'moduleName', action:'custommodule', selectTab:'moduleName_tab_chart'}).toJSC
```

Gdzie moduleName to wartość wprowadzona do atrybutu id tagu module

<module id="moduleName" name="Kanban" label="Kanban" icon="CallTransfer.svg" color="#c0392b">

Opcja umożliwia przekazanie zaznaczenia zakładki również dla poziomego paska zakładek (czyli struktura XML tabs -> tab -> tabs). Aby to zrobić należy użyć wywołania

App.openDialogByCls('APPLICATION', 0, ({name:'moduleName', action:'custommodule', selectTab:'moduleName\_tab\_chart/moduleName', action:'custommodule', selectTab:''moduleName', action:'custommodule', selectTab:''moduleName', action:''custommodule', selectTab:''moduleName', action:''custommodule', selectTab:''moduleName', action:''custommodule', selectTab:''', selectTab:''', selectTab:''', selectTab'', selectTab''', selectTab'', selectTab'', se

Czyli dodajemy separator "slash" (/).

Przykładowa struktura XML

```
<tab id="moduleName_tab_chart" label="Wykresy" rep_id="89">
           <tabs>
               <tab label="WSZYSTKIE" rep_id="89">
                   <buttons>
                       <button>
                           <id>refresh26</id>
                           <label>{translate(#Mój modułd#,#cModule_U#)}</label>
                           <dscrpt>Odśwież</dscrpt>
                           <onclick>
                                {AFTER_SUBMIT}
                           </onclick>
                            <icon>refresh.png</icon>
                       </button>
                   </buttons>
               </tab>
               <tab id="moduleName_tab_sub_chart" label="Kołowy" rep_id="189">
. . .
```

#### 1.3

Dodatkowe atrybuty dla taga tab to:

- enabled umożliwia dodanie zakładki bez jej wizualizacji. W zamian możliwy jest do wyświetlenia tekst zastępczy atrybut explain\_how\_to\_enabled
- explain\_how\_to\_enabled tekst informujący jak włączyć funkcjonalność danej zakładki
- visible flaga czy dana zakładka ma być widoczna nie ma znaczenia atrybut grp\_id lub prawo

Jako wartości dla tych atrybutów (oprócz explain\_how\_to\_enabled) należy zastosować następującą konstrukcję:

<tab label="{translate(#Zakładka#)}" rep\_id="90" enabled="{getGlobalSystemConfig(#RR#,#DDD.AAA#)}" explain\_how\_to\_enabled= </tab>

Wartość dla tych atrybutów przechowywane są w bazie w tabeli global\_sys\_conf. Aby je pobrać należy użyć funkcji getGlobalSystemConfig, która jak pierwszy argument przyjmuje wartość w kolumnie objnam (tabela global\_sys\_conf) a drugi argument to wartość w kolumnie varnam. Jako wartość value\_dla tych dwóch flag należy ustawić 0 lub 1.

# 1.2

Obsługa dodatkowego atrybutu tagu module "always\_show\_tabs". Ustawienie atrybutu always\_show\_tabs="TRUE" powoduje wymuszenie wyświetlenia zakładek górnego panelu

<module id="cModule\_U2" name="NEW" label="New" icon="fixdefect-48.png" always\_show\_tabs="TRUE">

</module>

### 1.1

Obsługa przycisku menu dokumentów

```
<br/>
<button type="DOCUMENTS_MENU" group="REQUEST_DOCUMENT_GROUP"><br/>
<id>new</id><br/>
<label>Nowy wniosek</label><br/>
<dscrpt>Nowy wniosek</dscrpt><br/>
<icon>new.png</icon><br/>
</button>
```

Dla atrybutu \*group\* możemy podać ID grupy dokumentów (types\_of\_documents\_groups.dctgid) lub GUID (types\_of\_documents\_groups.dgguid) co spowoduje wyświetlenie tylko elementów z tej gałęzi drzewa.

Obsługa customowego przycisku dla opcji menu zdefiniowanej jak dla zmiany 0.8

<button></button>	
	<id>pdf2</id>
	<label>asdsad</label>
	<dscrpt>PDF</dscrpt>
	<icon> pdf.png</icon>
	<pre><buttons></buttons></pre>
	<button></button>
	<custom_widget></custom_widget>
	3
	<button></button>
	<id>pdf</id>
	<label></label>
	<dscrpt>PDF</dscrpt>
	<pre><onclick> window.open('litescript.php/iyp/?script=GenerateReport&amp;ent_id=2&amp;ent_name=</onclick></pre>
	<icon> pdf.png</icon>
	<button></button>
	<custom_widget></custom_widget>
	38
	<button></button>
	<id></id>
	new
	<label></label>
	Nowa
	<dscrpt></dscrpt>
	Nowa pmAddTask
	<onclick></onclick>
	App.createDialog('pmAddTask', 'pmAddTask', './scripts/pmAddTask.inc', 'Utwórz nowe zada
	<icon></icon>
	new.gif

# 0.8

Pożądana funkcja przez wielu. Możliwość grupowania przycisków w jeden jako menu. Aby tego dokonać należy wewnątrz sekcji <br/>button> dodać podrzędną sekcję <br/>buttons> jak poniżej

<button></button>		
	<id>pdf2</id>	
	<label>asdsad</label>	
	<dscrpt>PDF</dscrpt>	
	<icon> pdf.png</icon>	
	<pre><buttons></buttons></pre>	
	<button></button>	
	<id>pdf</id>	
	<label></label>	
	<dscrpt>PDF</dscrpt>	
	<onclick> window.open('litescript.php/iyp/?script=GenerateReport&amp;ent_id=2&amp;</onclick>	ent_nam



#### 0.7

Obsługa pionowego panelu zakładek umożliwia tworzenia zakładek wewnątrz danej głównej zakładki. Użycie tego jest bardzo proste. Wystarczy pod danym tagiem tab (taki tag nie musi posiadać atrybutu rep\_id i żadnych innych tagów pod sobą oprócz tabs) dodać kolejną sekcję z tabs czyli np.

```
<module id="cModule_U2" name="NEW" label="New" icon="fixdefect-48.png" right="bswfms.processes">
       <toolbar>
       </toolbar>
       <filters>
       </filters>
<tabs>
  <tab label="AKTYWNE">
       <tabs>
  <tab label="AKTYWNE">
       <buttons>
<button>
    <id>newbug</id>
    <label>Zgłoszenie</label>
    <dscrpt>Nowe zgloszenie. Sprawa zostanie zalożona w teczce</dscrpt>
    <onclick>
    App.createDialog('createProcessForm', 'SimpleProcessCreatingForm', './modules/AProcesses/forms/SimpleProcessCreatingForm', '.
    </onclick>
    <icon>fix-defect-24.png</icon>
    </button>
     <button>
         <id>refresh20</id>
         <label>Odśwież</label>
         <dscrpt>Odśwież</dscrpt>
         <onclick>
         {AFTER_SUBMIT}
         </onclick>
         <icon>sync-24.png</icon>
         </button>
      </buttons>
```

```
</tab>
              <tab label="AKTYWNE">
                               <buttons>
<button>
                       <id>newbug</id>
                       <label>Zgłoszenie</label>
                       <dscrpt>Nowe zgłoszenie. Sprawa zostanie założona w teczce</dscrpt>
                        <onclick>
                       App.createDialog('createProcessForm','SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcesses/forms/SimpleProcessCreatingForm', './modules/AProcesses/forms/SimpleProcessCreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcessEreatingForm','./modules/AProcesEreatingForm','./modules/A
                       </onclick>
                       <icon>fix-defect-24.png</icon>
                       </button>
                       <button>
                                          <id>refresh20</id>
                                          <label>Odśwież</label>
                                          <dscrpt>Odśwież</dscrpt>
                                          <onclick>
                                          {AFTER_SUBMIT}
                                          </onclick>
                                         <icon>sync-24.png</icon>
                                         </button>
                           </buttons>
              </tab>
                                </tabs>
              </tab>
    </tabs>
 </module>
```

<div onclick="{FILE\_EXPLORER}">Eksplorator plików</div>

Przejdź do Menu