

Title: Portal

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

Version: 13

Date: 05/20/24 09:12:52

## Table of Contents

<i>Portal</i>	3
<i>pages</i>	3
<i>Tabela content</i>	3
<i>Menu główne - tabela menu</i>	4
<i>Opcje - tabela options</i>	4
<i>CSS - custom.css</i>	4
<i>Ikony</i>	4

## Portal

Moduł portal jest rozszerzeniem do systemu eDokumenty, które daje możliwość utworzenia portalu dostępnego dla użytkowników nieposiadających konta w systemie. Do struktury eDokumentów został dodany katalog portal w **public\_html/apps/**, wewnątrz którego znajduje się cała struktura modułu - wraz z silnikiem.

<https://edokumenty.oskar.eu/apps/portal?pageid=1>

**1** Szablon main.tpl

```

{header:h}
{slider:h}

<main class="container mb-5" role="main">
  {breadcrumbs:h}

  {content:h}
</main>

{footer:h}
          
```

**2**

components/header.tpl

components/slider.tpl

components/breadcrumbs.tpl

**3**

**content pageid=1**

Content może być

a) static - wówczas brana jest zawartość pola static z tabeli content

b) Szablon html - wówczas ładowany jest HTML z szablonu np. el\_strony1.html

c) Szablon html z zagnieżdżonymi odwołaniami do innych elementów content (znacznik z nazwą zawartą w polu uudid tabeli content np. {wnioseki:h})

**4**

components/footer.tpl

**Tabela portal.content**

contid	pageid	htmltpl	static	title
[PK] serial	integer	character varying(128)	character varying(128)	character varying(128)
1	1	components/header.tpl		
2	2	components/slider.tpl		
3	3	10		0 projekcie
4	4	2		Usługi
5	5	3	uslugi.tpl	
6	6	4	errors/404.tpl	404 - nie znaleziono
7	7	components/breadcrumbs.tpl		
8	8	components/footer.tpl		
9	9	1		
10	10	7	details.tpl	
11	11	6		
12	12	9	components/login.tpl	
13	13	1		

1. Pierwsze ładuje się szablon strony głównej

Szablony .tpl portalu znajdują się w katalogu **public\_html/apps/edokumenty/var/tpl/portal**, a plik strony głównej to **main.tpl**.

1. W szablonie main.tpl zawarte są makra ładujące szablony nagłówka, menu i okruszków.
1. Następnie w tabeli content wyszukiwane są elementy przypisane do strony która jest aktualnie załadowana - na podstawie parametru pageid w URL np. ?pageid=1. Parametr pageid może być również zastąpiony skrótem tzw. shortcodem zapisanym w tabeli pages.
1. Na końcu ładowana jest dalsza część szablonu main.tpl - czyli w praktyce jakaś stopka - footer.html

## pages

```

pageid INTEGER - id strony (klucz główny)
prntid INTEGER - pageid strony nadrzędnej - domyślnie NULL
name__ VARCHAR - tytuł strony
url__ VARCHAR - url - na potrzeby tworzenia linków na stronie i w menu
short_ VARCHAR - skrócony adres j/w
params JSON - dodatkowe parametry przekazywane do strony
          
```

Każda strona powinna zawierać treść, która jest definiowana w tabeli content

## Tabela content

Tabela content zawiera następujące pola:

```

contid INTEGER - id treści (klucz główny)
pageid INTEGER - id strony, na której treść ma zostać wyświetlona, w przypadku wartości NULL wyświetla wszędzie tam, gdzie
htmltpl VARCHAR - jeśli ustawione, to używa szablonu, który znajduje się w folderze public_html/apps/edokumenty/var/tpl/port
          
```

```

title_ VARCHAR - tytuł treści
static TEXT - statyczna treść strony - używana w przypadku braku definicji szablonu
tags_ VARCHAR - tagi, oddzielane przecinkami
author VARCHAR - autor treści
s_from TIMESTAMP - w przypadku ustawienia treść zostanie wyświetlona dopiero po tym terminie
s_to_ TIMESTAMP - w przypadku ustawienia treść przestanie być wyświetlana po tym terminie
order_ INTEGER - wartość po której sortowana jest treść na stronie (od najmniejszego do największego)
params JSON - parametry przekazywane do contentu
uuid_ TEXT - identyfikator treści - jeśli jest to zwykła treść, to musi być ustawiony na content, inaczej dajemy unika
type_ VARCHAR - domyślnie NULL, możliwe REPORT dla raportu oraz FORM dla formularza

```

Element content może być wyświetlany na 3 sposoby:

- jako **zawartość pola static** z tabeli content
- jako **szablon zapisany w polu htmlpl**
- jako element zagnieżdżony w innym szablonie za pomocą znacznika {uuid:h} - gdzie uuid to nazwa elementu zapisana w tabeli content w polu uuid.

W szablonie można stosować oprócz HTML-a znaczniki [Flexy](#).

## Menu główne - tabela menu

Zawartość menu wyświetlana jest na podstawie tabeli menus, która posiada strukturę:

```

menuid INTEGER - id elementu (klucz główny)
prntid INTEGER - id elementu nadrzędnego (menuid) - pozwala tworzy dwupoziomowe menu
pageid INTEGER - id strony, do której ma prowadzić link
label_ VARCHAR - tekst, który zostanie wyświetlony w menu oraz jako title (po najechnaniu myszką na link)
order_ INTEGER - służy do sortowania kolejności elementów w menu - od najmniejszego do największego
icon_ VARCHAR - klasa ikony w formacie nazwa - lista dostępna na https://material.io/icons/ - w przypadku null ikona nie
onlyic BOOLEAN - w przypadku ustawienia na true label nie jest wyświetlany, a jedynie ikona

```

## Opcje - tabela options

Ustawienia systemowe zapisane są w kolumnie options w formie klucz-wartość:

```

opt_id - INTEGER - klucz główny, id wpisu
key_ - VARCHAR - nazwa klucza
value_ - VARCHAR - wartość

```

Lista kluczy obsługiwanych przez system:

```

default_page - id strony głównej
name - nazwa serwisu - widoczna wewnątrz tagu <title>
description - opis dla meta tagów w sekcji <head>
theme - folder szablonu, domyślnie jest to portal
404 - id strony błędu 404

```

## CSS - custom.css

Portal pozwala na dodawanie własnego kodu CSS. Szablon, który to umożliwia umiejscowiony został tam, gdzie pozostałe pliki typu assets, czyli w **public\_html/apps/portal/var/**, a w tym wypadku dokładnie w podfolderze css. Jego pełna ścieżka to: **public\_html/apps/portal/var/css/custom.css**

## Ikony

Lista dostępnych ikon znajduje się na stronie: <https://material.io/icons/> - w celu ich użycia należy użyć składni:

```

<i class="material-icons">nazwa_ikony</i>
np.:

```

```
<i class="material-icons">home</i>
```