

Title: Integracja z zewnętrznymi systemami

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

Version: 20

Date: 11/21/24 21:33:57

Table of Contents

<i>Integracja z zewnętrznymi systemami</i>	3
<i>Sposób utworzenia połączenia z systemem Optima</i>	3

Integracja z zewnętrznymi systemami

Sposób utworzenia połączenia z systemem Optima

```
!#comment
```

Aby umożliwić synchronizację z systemem Optima należy wykonać poniższe czynności.

```
* zmiana stałej w pliku config.inc
* sprawdzenie danych w tabeli wsd1_config_table w systemie eDokumenty
* utworzenie tabeli kolejki w systemie Optima
* założenie triggerów na tabele
  * dok!__Dokument
  * adr!__Ewid
  * kh!__Kontrahent
* założenie widoków (eDokumenty > 2.0 RC15)
  * optima_export_all_contacts_documents_view.sql
  * optima_export_all_contacts_view.sql
  * optima_export_all_queued_contacts_documents_view.sql
  * optima_export_all_queued_contacts_view.sql
```

== Zmiana stałej ==

W pliku config.inc należy zmienić stałą "OPTIMA_DATA_SOURCE_DTSCNM" na

```
{{{
define('OPTIMA_DATA_SOURCE_DTSCNM', 'nazwa');
}}}
```

gdzie nazwę należy zastąpić nazwą bazy danych systemu Optima

== Sprawdzenie danych w tabeli ==

Jeśli w systemie !eDokumenty tabela wsd1_config_table jest pusta należy wykonać odpowiedni skrypt SQL dla Optima znajduję

```
{{{
apps\edokumenty\classes\BsConnect\install\data\optima_wsd1_ins.sql
}}}
```

następnie należy przejść do punktu "Konfiguracja tabeli wsd1_config_table" aby dokonać niezbędnych poprawek oraz ustawić

== Ustawienia w tabeli wsd1_config_table ==

Następny krok to sprawdzenie tabeli wsd1_config_table oraz czy zawiera dane. W przypadku jeśli jest pusta proszę udać się

```
'''Opis kolumn'''
```

* Adres - adres pliku (serwera), jeśli w nazwie występuje słowo {host} należy ja zamienić na lokalizację systemu eDokumenty

```
{{{
UPDATE wsd1_config_table SET wsd1__ = replace(wsd1__, '{host}', 'localhost:8080/edokumenty');
}}}
```

* System - wskazuje na nazwę systemu dla którego konfigurowana jest dana metoda (zostawiamy bez zmian)

* Warunek SQL - warunek po jakim będą synchronizowane dane

* dla akcji "Podwiąż kontakt" z lewej strony znaku równości wskazujemy kolumnę z systemu zew. natomiast z prawej w w

```
{{{
(np. dla OPT!MY: Knt_Nip='{nip___}' AND Knt_Nazwa1='{name_1}' co oznacza, że będzie
wyszukiwanie kontaktu w systemie OT!MA gdzie kolumna Knt_Nip będzie równa numerowi nip
kontaktu z systemu eDokumenty (token {nip___} jest zamieniany na dane) itd)
}}}
```

* dla pozostałych akcji po lewej stronie jest nazwa kolumny z systemu eDokumenty a z prawej token z maski (zobacz w katal

Różnica między tymi metodami polega na tym iż w przypadku szukania kontaktu w systemie zew. należy wykonać zapytanie na ze

Reszta metod służy do porównania danych przychodzących z systemu zew. do systemu eDokumenty. Zapytanie jest wykonywane na

Tablica (mapa kolumn) ułatwia sparsowanie danych i lepsze rozeznanie przykład

```

{{{

```

```

apps\edokumenty\etc\sync\OPTIMA_columns_map.ini

```

Przykładowa konfiguracja tablicy mapy dla kontaktu dla OPT!MY

```

[contacts]

```

```

Knt_KntID = contid

```

```

Knt_Nazwa1 = name_1

```

```

Knt_Nazwa2 = name_2

```

```

Knt_Nazwa3 = name_2

```

```

Knt_Kraj = countr

```

```

Knt_Wojewodztwo = woj___

```

```

Knt_Powiat = powiat

```

```

Knt_Ulica = street

```

```

Knt_NrDomu = bldnum

```

```

Knt_NrLokału = fltnum

```

```

Knt_Miasto = city___

```

```

Knt_KodPocztowy = code___

```

```

Knt_Nip = nip___

```

```

}}}

```

Oznacza to, że do dyspozycji będą dane pod danymi kluczami np.: klucz "nip!_" będzie zawierał numer nip kontaktu z systemu

```

{{{

```

```

nip___='{nip___}'

```

```

}}}

```

Czyli szukamy kontaktu w bazie eDokumenty gdzie nip!_ (lewa strona) kontaktu z bazy eDokumenty jest równy numerowi nip z

== Utworzenie tabeli kolejki ==

W systemie Optima (w bazie) wykonujemy skrypt z pliku

```

{{{

```

```

apps\edokumenty\classes\BsConnect\install\sql\optima\optima_export_queue.sql

```

```

}}}

```

Po wykonaniu tego skryptu w systemie Optima powinna pojawić się dodatkowa tabela o nazwie export_queue.

== Założenie triggerów ==

Triggery mają za zadanie dodawać do kolejki dokumenty i kontakty które zostały zmodyfikowane bądź dodane do systemu Optima

```

{{{

```

```

apps\edokumenty\classes\BsConnect\install\sql\optima

```

```

}}}

```

znajdują się pliki oryginalne. Kolejność ich wykonywania nie ma znaczenia. Po ich wykonaniu należy sprawdzić czy tabele do

```

[[BR]]Dla tabeli dokumentów

```

```

[[BR]]

```

```

[[Image(opDk.png, nolink)]] [[BR]]

```

Natomiast dla tabeli Kontrahenci [[BR]]

```

[[Image(opKh.png, nolink)]] [[BR]]

```

Możemy przetestować działanie mechanizmu poprzez dodanie do systemu Optima dokumentu lub kontrahenta i sprawdzeniu czy w t

== Założenie widoków ==

Począwszy od wersji 2.0 RC16 można decydować jakie dane mają być pobierane z systemu zew. poprzez modyfikacje specjalnych

```

{{{

```

```

edokumenty\classes\BsConnect\install\sql\optima\views

```

```

}}}

```

Można też je pobrać z załączonego niżej katalogu optima.zip. Widoki należy wykonać na bazie Optima (MSSQL).

```

''' Opis widoków '''

```

```

* optima_export_all_contacts_documents_view - wyświetla wszystkie dokumenty, które mają kontrahenta

```

```

* optima_export_all_contacts_view - wyświetla wszystkich kontrahentów

```

```

* optima_export_all_queued_contacts_documents_view - wyświetla wszystkie dokumenty, które mają kontrahenta oraz zostały z

```

```

* optima_export_all_queued_contacts_view - wyświetla wszystkich kontrahentów którzy zostali zmodyfikowani(dodani do syste

```

kolejność wykonywania nie ma znaczenia.

Widoki te można modyfikować według potrzeb(dodawanie kolumny itd) eliminując w ten sposób dodatkową pracę ze strony programisty.

Widoki modyfikujemy jeśli klient zażyczy sobie aby z systemu zew. były pobierane dodatkowe dane, których podstawowa definicja jest następująca:

```

{{{
OPTIMA_columns_map.ini
}}}

```

dodajemy kolejny wpis pod odpowiednim indeksem [contacts] - kontakty, [documents] - dokumenty z rzutowaniem danych z dodanej tabeli

```
''' Jaka metoda jaki widok wykorzystuje '''
```

```

* Importuj wszystkie kontakty - Optima_export_all_contacts_view
* Aktualizuj dane - optima_export_all_queued_contacts_view
* Aktualizuj dokumenty - optima_export_all_queued_contacts_documents_view
* Aktualizuj dane kontaktów - optima_export_all_queued_contacts_view
* Aktualizuj dokumenty kontaktów - optima_export_all_queued_contacts_documents_view
* Pobierz wszystkie dokumenty - optima_export_all_contacts_documents_view

```

```
== Reset ==
```

Na wypadek gdyby wprowadzone zmiany miały być z jakiegoś względu wycofane z bazy optymy, należy wykonać plik reset_optima.py

```
== Raporty z parametrem klienta ==
```

Aby raportować rozrachunki z klientem, sprzedaż czy cokolwiek z nim związane można do tego użyć raportów ze zdefiniowanym parametrem klienta