**Wikiprint Book**

**Title: Automatyzacja procesów workflow**

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

**Version: 133**

**Date: 09/05/25 11:50:12**

## Table of Contents

*Automatyzacja procesów workflow* 3
- *Podstawowe informacje* 3
- *Dla zaawansowanych* 5
- *Trochę teorii* 6

# Automatyzacja procesów workflow

## Podstawowe informacje

API Komend akcji Obecna implementacja komend nie uwzględnia ich pełnej funkcjonalności a jest jedynie zalążkiem. API komend będzie formatu podobnego do csv <nazwa_param>="wartość"[,] czyli np.

```
target="20",dscrpt="Wezwanie, uwaga!"
status="4",controlQuery="SELECT status = 3 FROM processes WHERE prc_id=$prc_id"
```

Nie zaleca się na obecnym etapie stosowania znaków specjalnych w wartościach parametrów, między nawiasami powinna się znaleść wartośc liczbowa ("5") lub tekstowa bez znaków specjalnych ("Wezwanie do wykonania etapy") API parametrów dla komend

Od wersji 2.2 została dodana możliwość wprowadzania komunikatów w klasach komend, czyli po wykonaniu komendy otrzymujemy informację jak poniżej

**Przekaż dokument**

Komenda służy do automatycznego przekazywania dokumentu na wybrane stanowiska za pomocą komendy oraz procedury.

- to = "1" - parametr wskazujący do kogo ma zostać przekazany oryginał , jeśli parametru nie będzie, lub będzie pusty oryginał zostaje
- dw = "2,3,4,5" - do wiadomości
- udw = "6,7,8" - ukryte do wiadomości

Każda z tych wartości to orunid (najlepiej sprawdzić w orgtree_view)

**Sprawdź czy pole jest wypełnione**

Komenda służy do sprawdzania czy dane pole formularza jest wypełnione. Przyjmuje 3 parametry z czego 1 jest konieczny.

- field="featid|8" lub "symbol" - pole które ma sprawdzić
- alert = "Wypełnij pole symbol" - wiadomość w przypadku pustej wartości w polu
- doAlertField = "t" - czy ma podświetlić pole jeśli było wymagane jako wypełnione (domyślnie f)

W przypadku parametru field może on obsłużyć zarówno cechy jak i inne pola. W przypadku cechy wpisujemy featid|id cechy z tabeli features dla której będzie wykonywana komenda. Jeśli podamy inny klucz wtedy zostanie od sprawdzony w danych formularza. Klucz wtedy powinien się nazywać tak samo jak nazwa kolumny w bazie w tabeli documents lub processes. Komenda nie pozwala na jednoczesne sprawdzanie dla kilku pól jednak jest możliwość ustawienia kilku tych samych komend z innymi parametrami dla akcji przed lub po zapisie etapu.

W przypadku jeśli cena będzie wpisana etap zostanie zaznaczony jako wykonany.

**Sprawdź prawdziwość warunku SQL**

Komenda służy do sprawdzania warunku SQL. Komenda może być użyta tylko dla definicji etapu. Przyjmuje 3 parametr

- query ="SELECT cena IS NOT NULL FROM table WHERE prc_id = {PRC_ID}/doc_id = {DOC_ID}" - sql
- alert = "Wypełnij pole cena" - wiadomość w przypadku niespełnienia sql
- success = "Pole cena wypełnione" - wiadomość w przypadku spełnienia sql

W przypadku jeśli cena będzie wpisana etap zostanie zaznaczony jako wykonany.

**Wróć do etapu**

Komenda służy do cofania procedury do konkretnego etapu podanego w parametrze. Komenda może być użyta tylko dla definicji etapu. Przyjmuje 1 parametr ptstid - id etapu z tabeli stages_def

- ptstid ="18" *np OKREŚLENIE WARTOŚCI ZLECENIA*

Co oznacza że po zaznaczenie danego etapu dla którego zdefiniowano taka komendę procedura zostanie cofnięta do etapu 18 (np OKREŚLENIE WARTOŚCI ZLECENIA) a co za tym idzie wszystkie etapy od 18 do aktualnie zaklikniętego włącznie zostaną ustawione na niewykonane.

**Utwórz zdarzenie**

Komenda służy do utworzenia zadania dla zalogowanego pracownika lub innego wpisanego w parametrze emp_id jaki usr_id. Zgodnie z #2390. komenda przyjmuje dwa parametry opis zadania oraz emp_id (zgodnie z wymogami komenda akceptuje parametr {LOGGED_USER} który wskazuje na zalogowanego pracownika)

- dscrpt="Aktualizacja dokumentacji wdrożenia"
- emp_id="{LOGGED_USER}" (lub usr_id z tabeli users)

UPDATE Zgodnie z wymogami ticketa #3155 dodano kolejne parametry

- trmtyp="TODO" - typ zdarzenia, może być do wyboru MEETING, PHONECALL, ALARM
- start_="featid|81"; - na kiedy ma dodać zdarzenie, wartość jest pobierana z cechy dlatego definicja np featid|81
- interval="+ 4 days" - interval za jaki utworzyć się zdarzenie, można tworzyć wstecz poprzez -

Nowe parametry dostępne w wersji >= 2.7.9

- ptstid="2" - id etapu z tabeli stages_def lub (ptstid="next") dla następnego etapu (ten parametr nie jest wymagany)

**Sprawdź czy istnieje dokument**

Komenda służy do sprawdzania czy w danej sprawie występuje konkretny dokument. Zgodnie z #2082. komenda przyjmuje dwa parametry typ dokumenty (id z tabeli types_of_documents) oraz status (wartość z kolumny state_ z tabeli documents)

- dctpid="9"
- state_="2"

**Utwórz zadanie dla etapu**

Komenda działa w dwojaki sposób. Bez parametru dodaje zadanie dla aktualnie zaklikiwanego etapu. Z parametrem doda zadanie dla etapu wskazanego w parametrze. Przyjmuje 1 parametr - ptstid. Parametr ten nie jest wymagany. Parametr ptstid jest to id etapu. z tabeli stages_def.

- ptstid="9"

Czyli do kolumny params w tabeli action_commands wpisujemy np. tpstid="9" lub w interfejsie użytkownika w słownikach

**Zmień status sprawy**

Obecnie przyjmuje tylko jeden parametr jest to klucz główny tpstid z tabeli types_of_processes_states który oznacza na jaki status zostanie zmieniony status sprawy. Komenda wpisuje kod statusu state_ do klucza pr_sta. Obecnie działa tylko w połączeniu ze sprawami ze względu na to iż dokumenty jest nie obsługują statusów

- tpstid="9"

Czyli do kolumny params w tabeli action_commands wpisujemy np. tpstid="9" lub w interfejsie użytkownika w słownikach

**Uprawnij grupę pracowników do sprawy**

Komenda działa niezależnie od tego czy jest ustawiona dla procedury czy etapu. Uprawnia ona wpisana grupe w parametrach na określoną maskę do sprawy. Jeśli dana grupa pracowników została już wcześniej uprawniona wtedy dostajemy komunikat o tym jednak zama sprawa, komenda, etap jest zapisywana.

Parametry

- grpnam="Pracownicy" - nazwa grupy z tabeli groups
- attrib="rwnd" - maska uprawnień według specyfikacji Integracja z systemem Subiekt?

Oba parametry są wymagane czyli ciąg z parametrami wygląda np. grpnam="Pracownicy",attrib="rwnd"

Znaczenie poszczególnych flag:

r - Odczyt
w - Zapis zadań i dokumentów
d - Oglądanie wszystkich dokumentów

n - Powiadamianie o nowych dokumentach, zadaniach i komentarzach
m - Zarządzanie (karta ogólne i Uprawnienia)

**Utwórz dokument**

Komenda ta tworzy dokument dla etapu procedury (nie obsługuje utworzenie dokumentu dla procedury). Przyjmuje również tylko 2 parametry i jest nim klucz główny dctpid z tabeli types_of_documents oraz opis jaki zostanie użyty w generowanym dokumencie.

- dctpid="5" *-id typu dokumentu*
- dctptp="Note" *-typ dokumentu*
- dscrpt="Wezwanie do wykonania etapy procedury" *-opis dokumentu*
- do="1" *-id jednostki do której przekazać dokument(orunid)*
- dw="2,3,4" *-(dw = do wiadomości) jeden lub więcej id (rozdzielone znakiem interpunkcyjnym, zwanym przecinkiem) jednostki do której przekazać kopie dokumentu(orunid)*

Przy standardowej instalacji taki parametr utworzy dokument typu notatka służbowa.
Do kolumny params w tabeli action_commands wpisujemy np. dctpid="5",dscrpt="Wezwanie do wykonania etapy procedury" lub w interfejsie użytkownika w słownikach

Późniejsza implementacja będzie również uwzględniach parametry z formularza np nazwę etapu, procedury, lub klucz główne jak np prc_id, contid itd. Pełna funkcjonalność parametrów już niebawem ;)

**Utwórz zadanie dla następnego etapu** Komenda tworzy zadanie dla następnego etapu procedury, o ile w kolejnym etapie jest określone stanowisko i jest na nim user. Po załatwieniu zadania automatycznie odznacza się etap procedury. Po załatwieniu etapu procedury, automatycznie zaznaczane jest zadanie jako załatwione.

Parametry:

- brak możliwości określenia - mile widziane zgłoszenia w tym temacie

## Dla zaawansowanych

W workflow biorą udział następujące tabele:

- procedures_def - tabela procedur - przechowuje informacje o procedurze np. Zatwierdzenie faktury kosztowej
- stages_def - tabela etapów - przechowuje definicje poszczególnych etapów np. Akceptacja Prezesa
- stages - instancje etapów - przechowuje informacje o zapisanych etapach konkretnych procesów: spraw, dokumentów
- proc_actions - akcje powiązane z procedurami lub z etapami, wykonują się przed lub po zapisie np. beforeStageChange
- action_commands - komendy wykonywane przez system na akcjach - wybierane spośród zawartych w katalogu commands - można dodać parametry, które dodają się do standardowych dwóch Obiektu Akcji oraz obiektu encji powiązanej z wykonywaną akcją np. Dokument albo Sprawa

```
AKCJE
oblig=# \d proc_actions
                         Table "public.proc_actions"
Column |  Type   |                          Modifiers
--------+---------+-------------------------------------------------------------
pracid | integer | not null default nextval('proc_actions_pracid_seq'::regclass)
actnam | text    | [onBeforeProcedureChange|onAfterProcedureChange|onBeforeStageChange|onAfterStageChange]
keyval | integer |
clsnam | text    |

KOMENDY
oblig=# \d action_commands
Table "public.action_commands"
Column |  Type   | Modifiers
--------+---------+-----------
pracid | integer |
cmdnam | text    |
params | text    |
```

## Trochę teorii

Tworzenie prostych procesów workflow nie wymaga dużego przygotowania, ale do tworzenia bardziej zaawansowanych modeli konieczna jest minimalna znajomość teoretycznych zasad rządzących przepływem procesów.

Podstawy teoretyczne