

Robotyzacja na punktach ACD typu „Import dokumentów zakupu z KSeF” – przykład

Zastosowanie

Skrypt **RPA** umożliwia automatyzację procesu pobierania faktur z platformy Krajowego Systemu e-Faktur **KSeF** bezpośrednio do systemu **Comarch BPM**. Rozwiązanie to eliminuje konieczność ręcznego importu, zapewniając ciągłość zasilania punktów **ACD** nowymi dokumentami oraz odpowiednią reakcję na limity wydajnościowe serwerów ministerialnych.

Zawartość przykładu

- **Skrypt C#:** Implementacja przygotowana w języku C# z wykorzystaniem obiektów globalnych systemu (Globals.ACD, Globals.Common).

[Kliknij, aby pobrać przykład](#)

Uwaga

Funkcjonalność obsługi limitów ministerialnych (reakcja na błąd 429) jest wspierana od wersji Comarch BPM 2026.0.0.2.

Zasada działania

Głównym zadaniem skryptu jest nawiązanie połączenia z serwerami ministerialnymi i przekazanie dokumentów do zdefiniowanego punktu ACD oraz określonego typu obiegu.

- **Konfiguracja Punktu ACD:** Możliwość wskazania konkretnej skrzynki odbiorczej (np. Dokumenty z KSeF – (A)FZ).
- **Automatyczne przypisanie do obiegu:** Zaimportowane faktury trafiają do konkretnego procesu, np. „Faktura kosztowa”.
- **Mechanizm Retry (Ponawianie):** Odporność na błędy komunikacji poprzez automatyczne podejmowanie do 3 prób importu w przypadku problemów.
- **Obsługa limitów (Błąd 429):** W przypadku przeciążenia serwerów KSeF, skrypt wstrzymuje pracę na 5 minut przed kolejną próbą.

Skrypt działa w oparciu o pętlę sterowaną odpowiedziami z API KSeF:

- **Inicjalizacja:** Resetuje poprzednie ustawienia filtrów metodą `ResetKSeFFilters()`, aby uniknąć konfliktów w sesji `Globals.ACD`.
- **Konfiguracja celu:** Ustawia Punkt ACD oraz Obieg jako parametry kontekstu importu.
- **Import i weryfikacja:** Wywołuje metodę `ImportKSeFDocuments()` i analizuje wynik.
 - Jeśli wynik wynosi 0, **import zakończył się sukcesem**.
 - Jeśli wynik wynosi -2, **skrypt rozpoznaje błąd 429 (przekroczenie limitu) i uruchamia pauzę przed ponowieniem**.

Parametry konfiguracyjne

Zmienna	Opis
<code>maxRetries</code>	Maksymalna liczba prób w przypadku wystąpienia błędów tymczasowych (domyślnie: 3).
<code>baseWaitTimeSeconds</code>	Podstawowy czas oczekiwania (90s) używany do wyliczania pauzy po otrzymaniu limitu żądań.
<code>punktACD</code>	Nazwa punktu wejścia dla dokumentów w systemie.
<code>nazwaObiegu</code>	Nazwa procesu, w którym pojawiają się zaimportowane faktury.

Techniczne aspekty kodu

- **Logowanie Startu:** Skrypt rejestruje postępy w `Globals.Common.Trace`, co jest kluczowe dla monitorowania pracy automatu w logach systemowych.
- **Exponential Backoff:** Skrypt stosuje wykładniczy wzrost czasu oczekiwania. Przy kolejnych próbach po błędzie 429 czas pauzy wynosi odpowiednio: 90s, 180s i 360s. Dzięki temu automat „cierpliwie” czeka na odblokowanie dostępu do API.
- **Bezpieczeństwo sesji:** W kodzie przewidziano opcjonalne resetowanie filtrów (`ResetKSeFFilters`) oraz możliwość wymuszenia konkretnego zakresu dat pobierania faktur (`SetKSeFFilters`), co zapobiega pobieraniu duplikatów.
- **Ochrona przed zapętleniem:** Jeśli skrypt napotka nieznany błąd (inny niż limit żądań), przerywa pętlę (`break`), umożliwiając administratorowi analizę kodu błędu w logach.

Konfiguracja przykładu

Aby skrypt mógł poprawnie realizować automatyczny import, niezbędna jest odpowiednia konfiguracja:

- **Utworzenie punktu ACD – Import dokumentów zakupu z KSeF** – gotowe przykłady (Punkt ACD, Typ obiegu), z których można skorzystać: [Przykłady procesów biznesowych](#)
- **Wskazanie w pliku BMP.exe.config w kluczu RPAFolderPath, w sekcji <appSettings> ścieżki do folderu z Comarch BPM Desktop, w którym system przechowuje skompilowane skrypty RPA.**

Przykład

Przykład wpisu: `<add key="RPAFolderPath" value="C:\Comarch BPM\RPA" />`.

Uwaga

Brak poprawnie zdefiniowanej ścieżki uniemożliwi przekompilowanie i zapisanie skryptu C#.

- W oknie **Konfiguracji automatycznego trybu pracy**:
 - **Dodanie skryptu**: Skrypt należy wkleić w zakładce „Edytor skryptów”.



- **Kompilacja**: Niezbędne jest użycie przycisku [Kompiluj i zapisz], aby system zweryfikował poprawność kodu i przygotował go do wykonania przez RPA.
- **Zapis**: Po poprawnej kompilacji wymagane jest zapisanie ustawień punktu za pomocą przycisku



[Zapisz].

Konfiguracja automatycznego trybu pracy

Edytor skryptów Uprawnienia

Nazwa programu: Robotyzacja_ACD_KSeF_Faktury

```
1 //Globals.ACD.ResetKSeFFilters();
2
3 //Globals.ACD.SetKSeFFilters(null, new DateTime(2024,6,4), new DateTime(2025,6,4));
4
5 // --- KONFIGURACJA ---
6
7
8
9 Globals.ACD.Show();
10
11 Globals.ACD.SetPoint("Dokumenty z KseF - (A)FZ",1);
12 Globals.ACD.SetDocumentFlow ("Faktura kosztowa (Elementy)_KSEF");
13
14 // Import faktur z KSeF z obsługą limitów
15
16 int maxRetries = 3;
17
18 int retryCount = 0;
19
20 int baseWaitTimeSeconds = 90;
21
22 bool success = false;
23
24 while (!success && retryCount < maxRetries)
25 {
26     int result = Globals.ACD.ImportKSeFDocuments();
27
28     if (result == 0)
29
30
31 }
```

Przekompilowany Skrypt C#

Gotowy Skrypt C#

```
//Globals.ACD.ResetKSeFFilters();
```

```
//Globals.ACD.SetKSeFFilters(null, new DateTime(2024,6,4), new  
DateTime(2025,6,4));
```

```
// --- KONFIGURACJA ---
```

```
Globals.ACD.Show();
```

```
Globals.ACD.SetPoint("Dokumenty z KseF - (A)FZ",1);
```

```
Globals.ACD.SetDocumentFlow      ("Faktura      kosztowa  
(Elementy)_KSEF");
```

```
// Import faktur z KSeF z obsługą limitów
```

```

int maxRetries = 3;

int retryCount = 0;

int baseWaitTimeSeconds = 90;

bool success = false;

while (!success && retryCount < maxRetries)

{

int result = Globals.ACD.ImportKSeFDocuments();

if (result == 0)

{
Globals.Common.Trace("Import faktur z KSeF zakończony sukcesem");

success = true;
}

else if (result == -2)

{
retryCount++;

// Wykładniczy wzrost czasu oczekiwania (exponential backoff)

// 1 próba: 90s, 2 próba: 180s, 3 próba: 360s

int waitTimeSeconds =
baseWaitTimeSeconds * (int)Math.Pow(2, retryCount - 1);

Globals.Common.Trace($"Przekroczono limit żądań KSeF (429 Too Many Requests). " +

$"Próba {retryCount}/{maxRetries}. " +

$"Oczekiwanie {waitTimeSeconds} sekund przed ponowną

```

```
próba...");  
  
if (retryCount < maxRetries)  
{  
    System.Threading.Thread.Sleep(waitTimeSeconds * 1000);  
}  
  
else  
{  
    Globals.Common.Trace("Osiągnięto maksymalną liczbę prób. " +  
    "Import faktur z KSeF nie powiódł się. " +  
    "Spróbuj ponownie później (zalecany interwał: minimum 15  
    minut).");  
}  
  
}  
  
else  
{  
    Globals.Common.Trace($"Błąd importu faktur z KSeF. Kod  
    błędu: {result}");  
  
    break;  
}  
  
}
```

Rozpoczynasz pracę z Comarch BPM (dawniej DMS) i chcesz

dowiedzieć się, jak korzystać z programu? A może masz już podstawową wiedzę o Comarch BPM (dawniej DMS) i chcesz dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch BPM!](#)

[Powrót do początku artykułu](#)