



## Lista kontrolna REF bez pozycji w

### Wymagania wstępne

Określ, jaki typ danych automatyka musi przetwarzać dla każdej transakcji; zdecyduj o typie danych dla zmiennej *TransactionItem*.

### Konfiguracja w UiPath Studio

- ☐ Utwórz **nowy projekt** używając szablonu **Robotic Enterprise Framework**:
  - ☐ Ustaw odpowiednią nazwę dla projektu - na podstawie nazwy procesu
  - ☐ automatycznego; Ustaw odpowiedni opis dla projektu.
- ☐ Edytuj **plik konfiguracyjny** projektu, znajdujący się w folderze projektu, *Data/Config.xlsx*:
  - ☐ Arkusz *ustawień*:
    - ☐ Ustaw odpowiednią nazwę dla parametru *logF\_BusinessProcessName*; zazwyczaj jest to ta sama nazwa co nazwa projektu; przydatne jest powiązanie procesu z komunikatami dziennika generowanymi podczas wykonywania;
    - ☐ Zdefiniuj wszelkie dodatkowe wartości wymagane dla procesu, takie jak ścieżki plików wejściowych, ścieżki aplikacji, parametry kontrolujące przepływ automatyki, wiadomości e-mail itp.
    - ☐ Aktywa poświadczeniowe również powinny być definiowane w tym arkuszu, zamiast w arkuszu Aktywa;
    - ☐ Parametry *OrchestratorQueueName* i *OrchestratorQueueFolder* mogą zostać usunięte.
  - ☐ Arkusz *Stałe*:
    - ☐ Ustaw wartość *MaxRetryNumber* na wartość większą niż 0, jeśli chcemy korzystać z mechanizmu retry wbudowanego w framework.
    - ☐ Opcjonalnie można zmienić wartości pozostałych pozycji zdefiniowanych arkuszu w zależności od automatyzowanego procesu.
  - ☐ Karta *aktywów*:
    - ☐ Dodaj wpisy dla wszystkich aktywów (oprócz aktywów po zdefiniowanych w Orchestrator dla tego procesu.
- ☐ Zapisz i zamknij plik *Config.xlsx*.

## Zmiana typu danych TransactionItem

- ☐ Edytuj przepływ pracy **Main.xaml**, aby zmienić typ danych dla zmiennej *TransactionItem* z domyślnej *QueueItem* na wymagany dla automatyzowanego procesu:

- ☐ Zlokalizuj zmienną *TransactionItem* w panelu *Variables*; zmień typ *Variable* na wymagany;

Zlokalizuj zmienną *TransactionData* w panelu *Variables*; w razie potrzeby zmień typ

- ☐ *Variable* na wymagany.

### W stanie **Get Transaction Data**:

- ☐ Zlokalizuj aktywność *End Process (Zatrzymaj żądany proces)* i ustaw wartość na domyślną wartość dla typu danych *TransactionItem*;

- ☐ Zlokalizuj aktywność przepływu *Invoke GetTransactionItem* i zmień typ danych dla argumentu *out\_TransactionItem* na wymagany typ oraz *io\_TransactionData* z *DataTable* na wymagany typ; zmiany i zamknąć okno dialogowe;

- ☐ Rozwiń sekcję *Exception* w sekcji *Try GetTransactionItem* i zmień typ danych dla argumentu *End Process (Could not get new transaction)*; zmianę do tej wykonanej na początku tej listy.

**Uwaga**  
2021-09-14 12:45:51

Jest to sygnał, że proces powinien zatrzymać. Może to być *Nothing* dla wielu typów danych jak *QueueItem*, *DataRow*, *DataTable*, *String*, *Object*, itp., 0 dla *Integer*, *Double*, *Decimal* lub innych typów

### W stanie **Process Transaction**:

- ☐ Zlokalizuj aktywność przepływu *Invoke Process* i otwórz listę *Argumenty*; zmień typ danych dla argumentu *in\_TransactionItem* z *QueueItem* na wymagany typ; kliknij OK, aby zapisać zmiany i zamknąć okno dialogowe;

- ☐ Zlokalizuj aktywność *Invoke SetTransactionStatus (Success)* i otwórz listę *Argumenty*; zlokalizuj wpis dla argumentu *in\_TransactionItem* i zmień wartość (nie typ danych!) z *TransactionItem* na *Nothing*;

- ☐ Zlokalizuj aktywność *Invoke SetTransactionStatus (Failure)* i zmień wartość (nie typ danych!) z *TransactionItem* na *Nothing*; zmianę jak poprzednio;

- ☐ Zlokalizuj aktywność *Invoke SetTransactionStatus (Error)* i zmień wartość (nie typ danych!) z *TransactionItem* na *Nothing*; zmianę jak poprzednio.

**Uwaga**  
2022-06-27 14:46:42

Spowoduje to wyłączenie raportowania statusu w kolejce. Nie należy używać kolejki. Jeśli wymagane jest inne raportowanie statusu, należy je zaimplementować w przepływie

## Stosowane aplikacje: otwórz/zamknij/zabij

- ☐ Edytuj przepływ pracy **Framework/InitAllApplications.xaml**; powinien on zawierać aktywności uruchamiające wszystkie aplikacje, przeprowadzające uwierzytelnianie i wstępna konfiguracja wymagana dla wszystkich transakcji w procesie;

- ☐ Edytuj przepływ pracy **Framework/CloseAllApplications.xaml**; wywołaj aktywność wylogować i zamknąć aplikacje otwarte w przepływie *InitAllApplications*;

- ☐ Edytuj przepływ pracy **Framework/KillAllProcesses.xaml**; umieść aktywność zamknięcie aplikacji używanych w automatyzacji.

**Uwaga**  
2022-06-27 14:48:57

**Uwaga**  
2022-06-27 14:48:55

To na wypadek, gdyby nieoczekiwany błąd uniemożliwił jego prawidłowe zamknięcie w przepływie pracy *CloseAllApplications.xaml*.

## Proces biznesowy: Dane transakcyjne i proces

### ☐ Edytuj przepływ pracy **Framework/GetTransactionData.xaml:**

- ☐ Zmień typ danych dla argumentu *out\_TransactionItem* z *QueueItem* na wymagany typ;
- ☐ Zmień typ danych dla argumentu *io\_TransactionDate* z *DataTable* na wymagany typ, jeśli to konieczne;
- ☐ Usuń aktywność *Get transaction item* (typu *UiPath.Core.Activities.GetQueueItem*);
- ☐ Przechowuj dane transakcji do argumentu *io\_TransactionDate*, jeśli to konieczne;
- ☐ Dodaj działania, aby sprawdzić, czy nowa transakcja jest dostępna;
- ☐ Na gałęzi *Transakcja dostępna* dodaj czynności niezbędne do pobrania informacji, które mają być przetworzone w jednej transakcji;
- ☐ W gałęzi *no more transactions available* przypisz *out\_TransactionItem* wartość stop;
- ☐ Ustaw odpowiednią wartość dla argumentu *out\_TransactionID* aktywność *out\_TransactionID*;
- ☐ Ustaw odpowiednie wartości dla argumentów *out\_TransactionField1* i *out\_TransactionField2* w kolejnych dwóch czynnościach pr

### ☐ Edytuj przepływ pracy **Process.xaml:**

- ☐ Zmień typ danych dla argumentu *in\_TransactionItem* z *QueueItem* na wymagany typ;
- ☐ Umieść wszystkie czynności i logikę wymaganą do przetworzenia

Uwaga  
2021-09-14 12:46:12

Uwaga  
2021-09-14 12:46:12

Uwaga  
2021-09-14 12:46:12

Pola te są dodawane do komunikatów dziennika, aby dodać dodatkowych informacji o transakcji.

Uwaga  
2022-06-27 14:47:48

Czynności powinny być pogrupowane w komponenty na podstawie wykonywanej akcji, w tym przepływie będziemy wywoływać te komponenty i dodawać niezbędne kroki łączące komponenty.