

Zatwierdzanie faktury zakupu w Comarch ERP XL z wykorzystaniem skryptów C#

Spis treści artykułu

- [Informacje podstawowe](#)
- [Konfiguracja typu obiegu](#)
- [Zatwierdzanie dokumentu FZ w Comarch ERP XL – jak działa](#)

Informacje podstawowe

Uwaga

Aby przykład działał poprawnie, konieczne jest spełnienie następujących warunków:

1. [Instalacja systemu Comarch ERP XL na serwerze IIS](#)
2. [Poprawna konfiguracja współpracy pomiędzy systemami Comarch ERP XL oraz Comarch DMS](#)
3. [Nadanie użytkownikowi IISUSER uprawnień do zapisu](#) w katalogu `bin\Scriptcs_bin` oraz katalogu `bin\.scriptcs_cache`, które znajdują się w [plikach aplikacji serwerowej Comarch DMS](#)
4. [Skopiowanie biblioteki `cdn\_api\[wersja\].net.dll` do katalogu `bin\Scriptcs\_bin`](#), biblioteka znajduje się w katalogu, w którym zainstalowano Comarch ERP XL, w przykładach wykorzystano `cdn_api20232.net.dll`
5. Należy sprawdzić, [czy w katalogu `bin` aplikacji web Comarch DMS znajduje się plik `cdn\_sys.dll`](#), jeżeli jest należy [usunąć go](#) z tej lokalizacji.

Skrypty C# zostały wykorzystane w kontrolkach typu Własna akcja i Komunikat. Skrypt dla kontrolki typu własna akcja utworzono na podstawie wzorca API XL

Wskazówka

Rekomendujemy wykorzystanie wzorca API XL jako elementu bazowego na potrzeby własnych rozwiązań.

Wzorzec API XL dostępny jest na indywidualnych stronach dla Partnerów w obszarze Comarch DMS\Przykłady\ERP XL: [Wzorzec API XL](#)

Konfiguracja typu obiegu

Przykład pokazuje, w jaki sposób administrator Comarch DMS może skonfigurować definicję obiegu, aby zatwierdzić dokument „Faktura zakupu” w systemie Comarch RP XL z poziomu Comarch DMS. W tym celu należy:

1. **dodać nowy typ obiegu** w ramach zakładki  **[Definicje obiegu dokumentów]**
2. **nadać nazwę typowi obiegu** – np. *Zatwierdzanie faktury zakupu* i prefix – np. ZAT
3. **nadać danemu administratorowi uprawnienie do inicjowania dokumentu** na zakładce „Ustawienia obiegu” w sekcji „Uprawnienia do inicjowania dokumentu”.

Zatwierdzanie faktury zakupu

Prefix: ZAT

KARTA OBIEGU SCHEMAT OBIEGU **USTAWIENIA OBIEGU**

Ogólne

Lista dokumentów jako książka nadawcza

Ustawienia dla aplikacji mobilnej

Filtry

Uprawnienia do inicjowania dokumentu

Nazwa spółki	Typ	Wartość	
Firma X	Pracownik	Administrator systemu	
Dodaj			
Obserwator dokumentu			

Nadawanie uprawnień do inicjowania dokumentu dla administratora – przykład

4. W ramach zakładki „**Schemat obiegu**” utworzyć etap początkowy „**Zatwierdzanie dokumentu**” i etap końcowy „**Koniec**” oraz **dodać uprawnienie dla danego administratora do etapów**.

Zatwierdzanie faktury zakupu

Prefix: ZAT

KARTA OBIEGU

SCHEMAT OBIEGU

USTAWIENIA OBIEGU

+

-

Etap

Szukaj

Zatwierdzanie dokumentu

Koniec

Etap

OGÓLNE

UPRAWNIENIA

PRZEKAZANIE

MAIL

KONTROLKI

Cofnij dokument

Właściciel dokumentu

Uprawnienia do etapu

Uprawnienia dodatkowe

Zawężanie uprawnień na wcześniejszym etapie

Dostęp do dok. dla operatora niepracującego z dok.

Uprawnienia

Nazwa spółki	Typ	Wartość	
Firma X	Pracownik	Administrator systemu	

Dodaj

Dodawanie etapów i uprawnień

5. W ramach zakładki „Karta obiegu” **dodać następujące kontrolki:**

A Kontrolka typu Dokument elektroniczny – „Faktura zakupu” (identyfikator: FakturaZakupu) – w polu „Typ” należy wybrać „Faktura zakupu”

Dokument elektroniczny

Nazwa wyświetlana	Faktura zakupu
Nazwa (identyfikator)	FakturaZakupu
Nazwa spółki	Firma X
Typ	Faktura zakupu
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Pokaż na liście	☐
Inicjowanie wartości	
Inicjowanie	Brak
Zmiana wartości (Obserwator)	
Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie
	Ustaw

Kontrolka „Faktura zakupu”

B kontrolka typu Własna akcja – „Zatwierdź dokument” (identyfikator: ZatwierdźDokument) – w ramach definicji kontrolki należy kliknąć w link „Ustaw” w polu „C# Script”, następnie w otwartym **oknie „Skrypt C#”** w polu „Miejsce wykonania skryptu” wybrać „Proces (exe)” (jeśli istnieje opcja wyboru) i **wprowadzić następujący kod**:

```
using System.Threading;
using System.Runtime.InteropServices;
using cdn_api;
```

```
static ThreadLocal<bool> threadToClarionAttached
= new ThreadLocal<bool>(() => false);
```

```

[DllImport("ClaRUN.dll")]
private static extern void AttachThreadToClarion(int flag);
public void AttachThreadToClarion()
{
    try
    {
        if (!threadToClarionAttached.Value)
        {
            AttachThreadToClarion(1); // C8
            threadToClarionAttached.Value = true;
        }
    }
    catch (DllNotFoundException ex)
    {
        // -----
        // Tu przekaz informacje o statusie działania skryptu
        // -----
        //Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text =
ex.Message;
        Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text=ex.Messa
ge;
    }
}

```

```

AttachThreadToClarion();

```

```

int _lSesjaID = 0;
string blad = "";

```

```

try
{
    int documentId = 0;
    int wersjaApi = odpowiedniawersjaapi;

    var loginInfo = new XLLoginInfo_odpowiedniawersjaapi()
    {
        Wersja = odpowiedniawersjaapi,
        ProgramID = "Comarch DMS",
    }
}

```

```

        Winieta = -1,
        TrybWsadowy = 1,
        Baza = "nazwafirmy",
        OpeIdent = "Login",
        OpeHaslo = "hasło",
        SerwerKlucza = @"serwerklucza"
    };

        var XLLoginResult =
cdn_api.cdn_api.XLLogin(loginInfo, ref _lSesjaID);
        if (XLLoginResult != 0)
        {
            blad = "Funkcja XLLogin zwróciła błąd nr:
" + XLLoginResult;
            throw new Exception();
        }

        // -----
-----
        // Tu dodaj kod obsługi API

        var otworzDocumentInfo
= new XL0twarcieNagInfo_odpowiedniawersjaapi
        {
            Wersja = odpowiedniawersjaapi,
            GIDTyp = 1521,
            GIDNumer
= Globals.MainFrame.FakturaZakupu.Id ?? 0,
            Tryb = 2

        };

        var XL0twarcieInfoResult =
cdn_api.cdn_api.XL0tworzDokument(_lSesjaID, ref documentId, ot
worzDocumentInfo);
        if (XL0twarcieInfoResult != 0)
        {
            blad = "Funkcja XL0tworzDokument zwróciła
błąd nr: " + XL0twarcieInfoResult;
            throw new Exception();
        }

```

```

                                var zamkniecieDokumentuInfo
= new XLZamkniecieDokumentuInfo_odpowiedniawersjaapi
                                {
                                    Wersja = odpowiedniawersjaapi,
                                    Tryb = 0
                                };

                                var XLZamknijDokumentResult =
cdn_api.cdn_api.XLZamknijDokument(documentId, zamkniecieDokume
ntuInfo);
                                if (XLZamknijDokumentResult != 0)
                                {
                                    blad = "Funkcja XLZamknijDokument zwróciła
bład nr: " + XLZamknijDokumentResult;
                                    throw new Exception();
                                }
                                // -----
                                -----
                                }
                                catch (Exception ex)
                                {
                                    if (blad == "")
                                        blad = "Nieokreślony bład: " + ex.Message;
                                }
                                finally
                                {
                                    var XLLogoutResult = cdn_api.cdn_api.XLLogout(_lSesjaID);
                                    if (XLLogoutResult != 0)
                                    {
                                        blad = "Funkcja XLLogout zwróciła bład nr:
" + XLLogoutResult;
                                    }
                                    // -----
                                }
                                -----
                                // Tu przekaż informacje o statusie działania skryptu
                                // -----
                                -----

                                //Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text =
blad;

```

```

        if (blad == "")
            Globals.MainFrame.ZatwierdźDokument_CSSSCRIPT.Text
= "Operacja zakończona sukcesem";
        else
            Globals.MainFrame.ZatwierdźDokument_CSSSCRIPT.Text =
blad;

}

```

gdzie **należy zmienić następujące wartości:**

odpowiedniawersjaapi – **wersja API ERP XL** zgodna z wersją pliku **cdn_api[wersja].net.dll**

nazwafirmy – **nazwa Firmy Comarch ERP XL** (nazwę firmy można sprawdzić w managerze baz Comarch ERP XL)

Login – **login operatora Comarch ERP XL**

hasło – **hasło operatora Comarch ERP XL**

serwerklucza – **serwer klucza licencji**



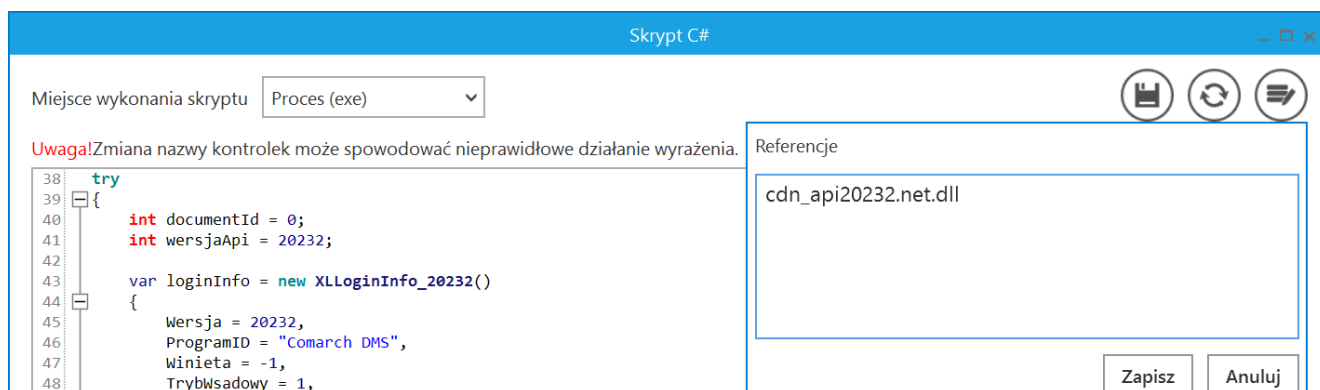
Następnie należy kliknąć w przycisk **[Referencje]**, znajdujący się w prawym górnym rogu okna „Skrypt C#”, w wpisać tam nazwę odpowiedniego pliku dll:

cdn_api[odpowiedni numer wersji].net.dll

Zapisz

Następnie należy nacisnąć przycisk

[Zapisz].



Fragment okna „Skrypt C#” z otwartym oknem „Referencje”

Po wprowadzeniu kodu do edytora i nazwy dll w oknie

„Referencje” należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.

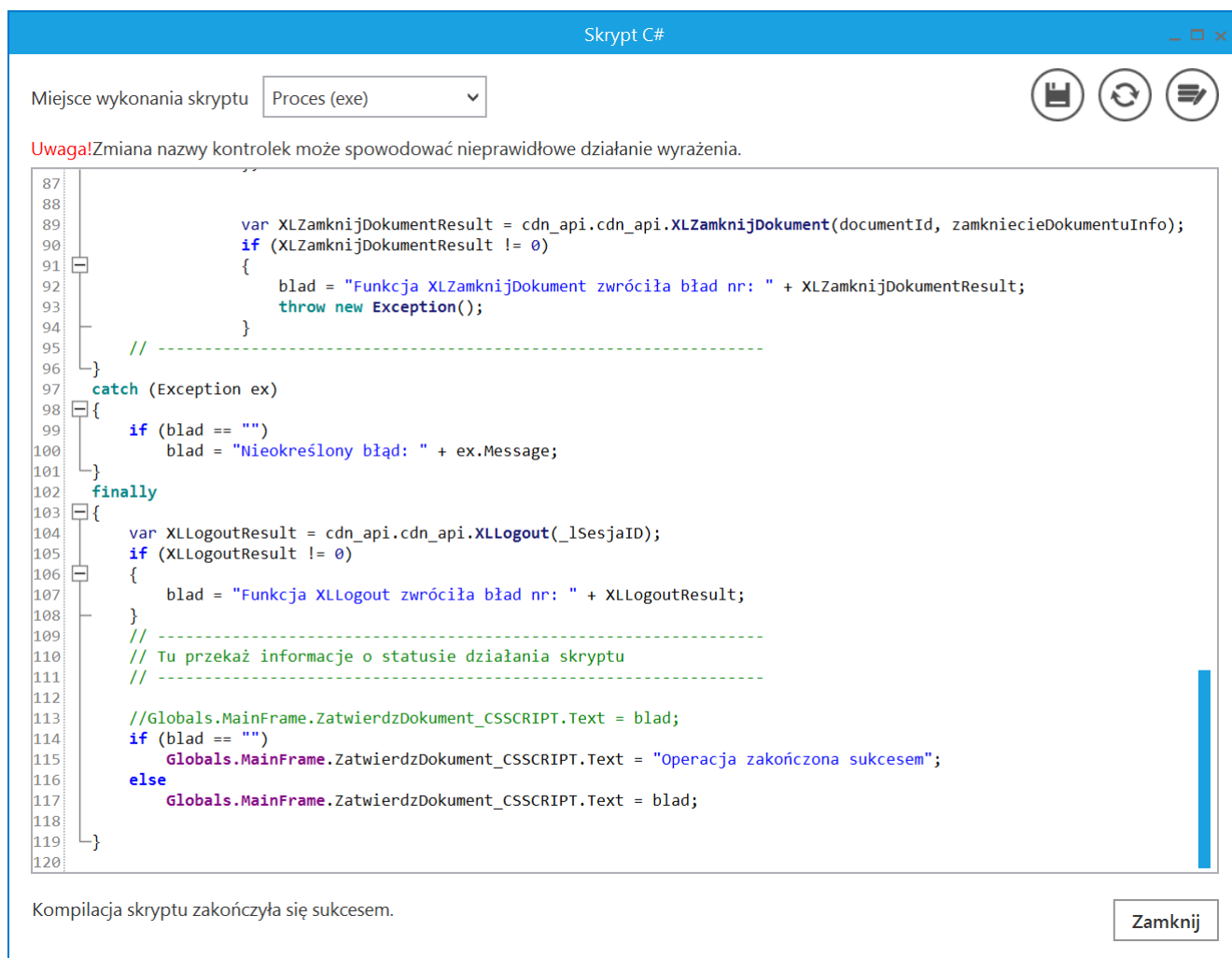
Uwaga

Jeśli w ramach okna „Referencje” po kliknięciu przycisku



[Referencje] wprowadzono niewłaściwą nazwę pliku dll,

wówczas po naciśnięciu przycisku  **[Kompiluj i zapisz]** w dolnej części okna wyświetlana jest informacja „Assembly not found”.



Widok okna „Skrypt C#” po skompilowaniu skryptu

C Kontrolka typu Komunikat – należy w ramach pola:

- „Nazwa (identyfikator)” wprowadzić nazwę „**Message1**”
- „Tytuł” wprowadzić tekst „**Status działania API ERP XL**”
- „Kontrolki powiązane” kliknąć w link „Dodaj” i wybrać kontrolkę „**Zatwierdź dokument**”
- „Wzór na wartość” wybrać wartość „**C# Script**”, kliknąć „Ustaw”, a następnie wprowadzić następujący kod:

```
Globals.MainFrame.Message1.Message  
= Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text
```

Po wprowadzeniu kodu należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.

Komunikat

Nazwa (identyfikator)

Message1

Zmiana wartości (Obserwator)

Tytuł

Status działania API ERP XL

Kontrolki powiązane

Zatwierdź dokument (ZatwierdzDokument) 

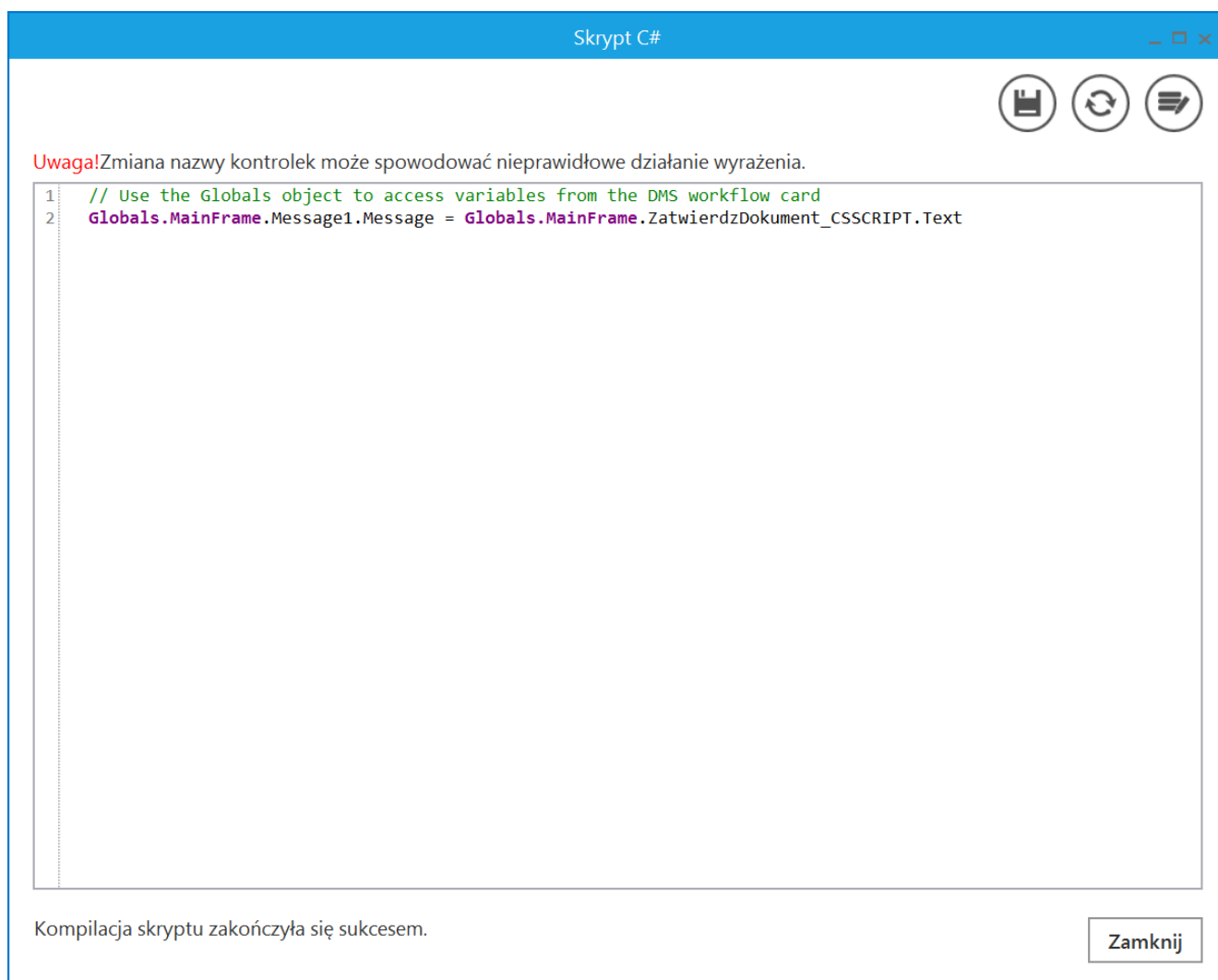
[Dodaj](#)

Wzór na wartość

C# Script 

[Ustaw](#)

Kontrolka „Message1”



Widok okna „Skrypt C#” po skompilowaniu skryptu

6. **zapisać typ obiegu** za pomocą przycisku



[Zapisz].

Zapisywanie typu obiegu

Zatwierdzanie dokumentu FZ w Comarch ERP XL – jak działa

Po zdefiniowaniu typu obiegu operator przechodzi na zakładkę



[Dokumenty], klika na typ obiegu „Zatwierdzanie faktury

zakupu” i naciska przycisk  [Dodaj]. Następnie kolejno:

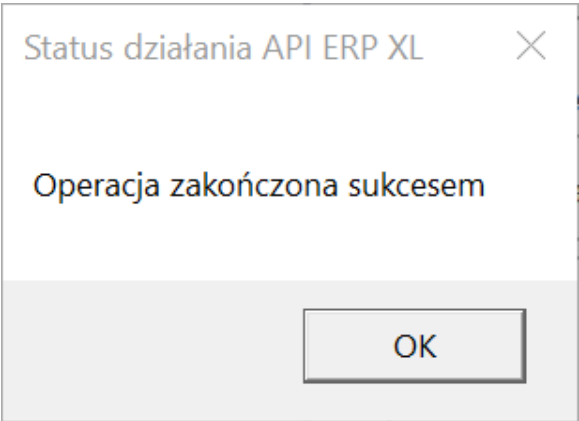
1. zapisuje nowy dokument, naciskając  [Zapisz]

2. dodaje dokument z Comarch ERP XL w ramach kontrolki „Faktura zakupu”

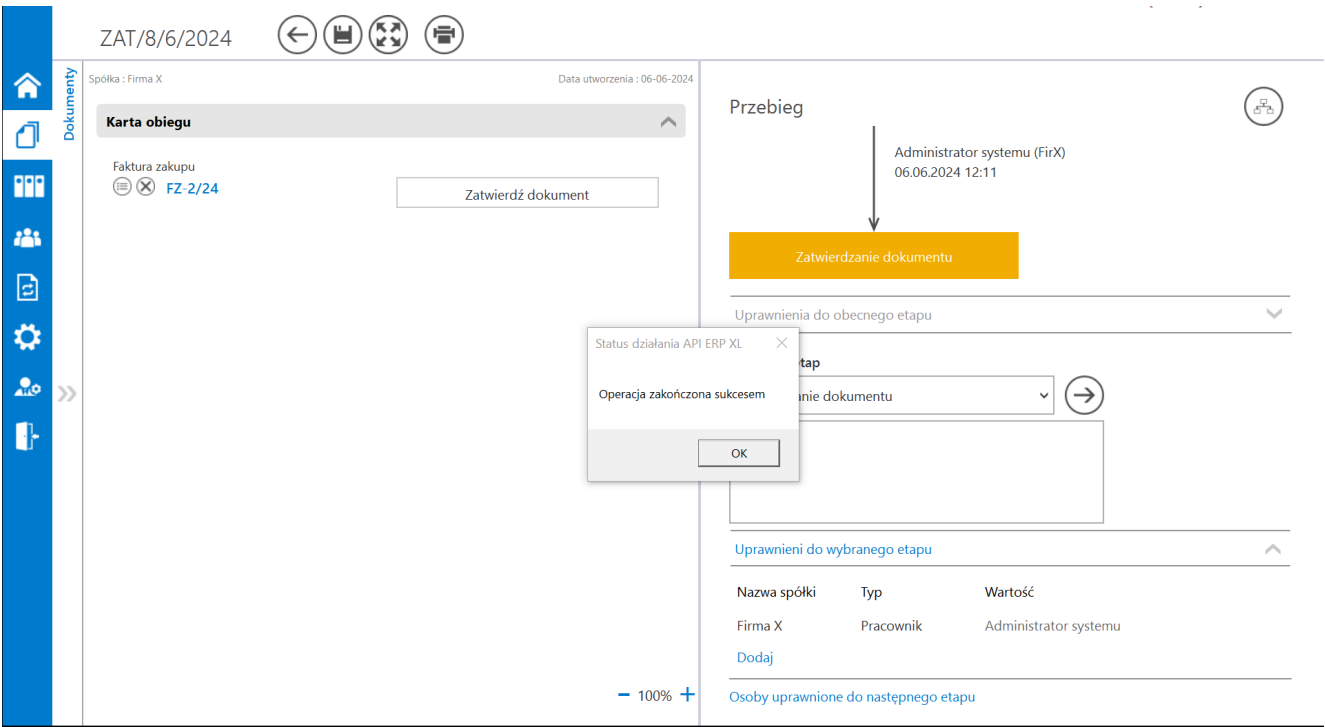
3. naciska przycisk „Zatwierdź dokument” – wówczas następuje zatwierdzanie dokumentu – poniżej przycisku widoczne są poruszające się punkty

4. Jeśli zatwierdzanie dokumentu zakończyło się powodzeniem,

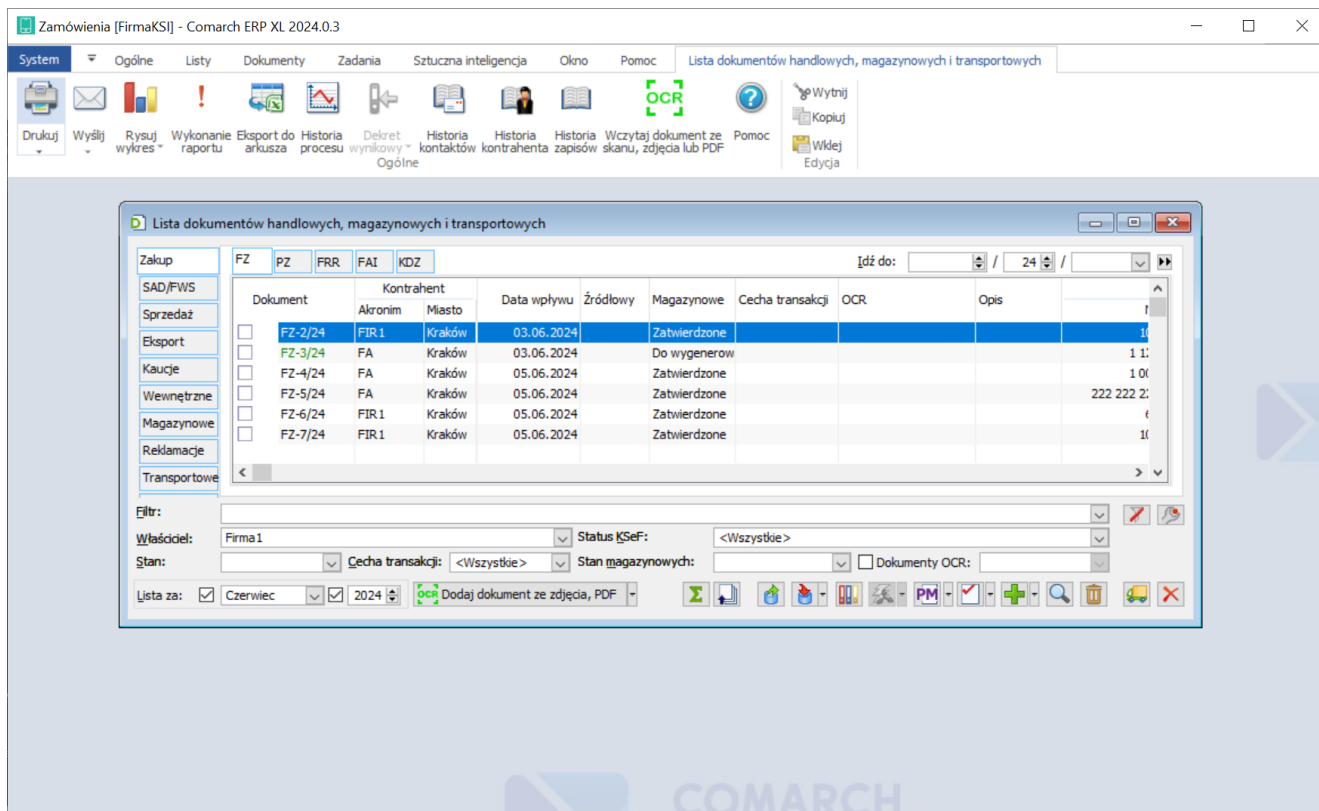
wówczas na środku ekranu wyświetlany jest następujący komunikat:



Komunikat wyświetlany, jeśli
zatwierdzanie dokumentu
zakończyło się powodzeniem



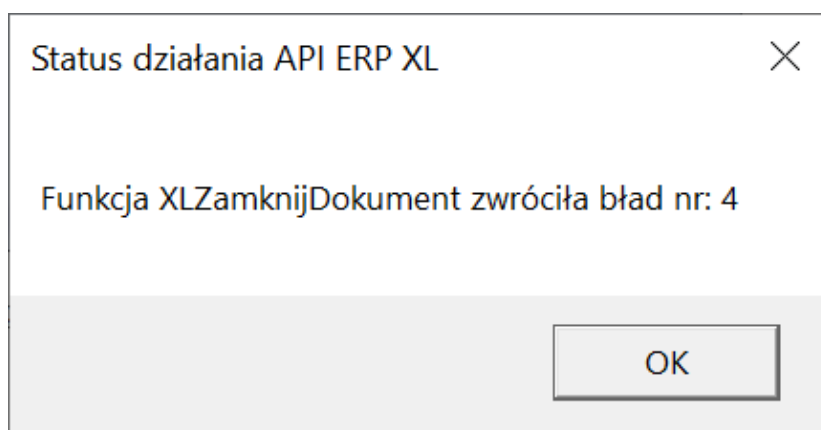
Widok szczegółów dokumentu z komunikatem o udanym
zatwierdzeniu dokumentu



Widok listy dokumentów w Comarch ERP XL z dokumentem FZ-2/24 zatwierdzonym z poziomu Comarch DMS

Uwaga

Jeśli operator w kontrolce „Faktura zakupu” wybierze dokument, który już został zatwierdzony w Comarch ERP XL, i kliknie przycisk „**Zatwierdź dokument**”, wówczas zostaje wyświetlony następujący komunikat:



Komunikat wyświetlany przy próbie
zatwierdzenia zatwierdzonego dokumentu

Rozpoczynasz pracę z Comarch DMS i chcesz dowiedzieć się, jak korzystać z programu? A może masz już podstawową wiedzę o Comarch DMS i chcesz dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch DMS!](#)

[Powrót do początku artykułu](#)