



SUNCODE
business process management



Instrukcja Użytkownika

Plus Workflow for OCI

SPIS TREŚCI

Rozdział 1. Wprowadzenie	3
Rozdział 2. Instalacja modułu Plus Workflow for OCI.....	3
Rozdział 3. Konfiguracja akcji formularza modułu w PWE	4
Rozdział 4. Ustawienia OCI	7
Spis rysunków.....	15



Rozdział 1. Wprowadzenie

Moduł Plus Workflow for OCI pozwala na komunikację systemu Plus Workflow z zewnętrznymi portalami B2B w standardzie OCI.

Omawiany moduł realizuje następujące funkcjonalności:

- połączenie się z zewnętrznymi portalami,
- ustawienie dostępu dla użytkowników do poszczególnych katalogów,
- pobranie produktów z koszyka portalu B2B do Plus Workflow.

Moduł Plus Workflow for OCI dostępny jest w postaci wtyczki do systemu Plus Workflow (instalacja wtyczki w załączniku).

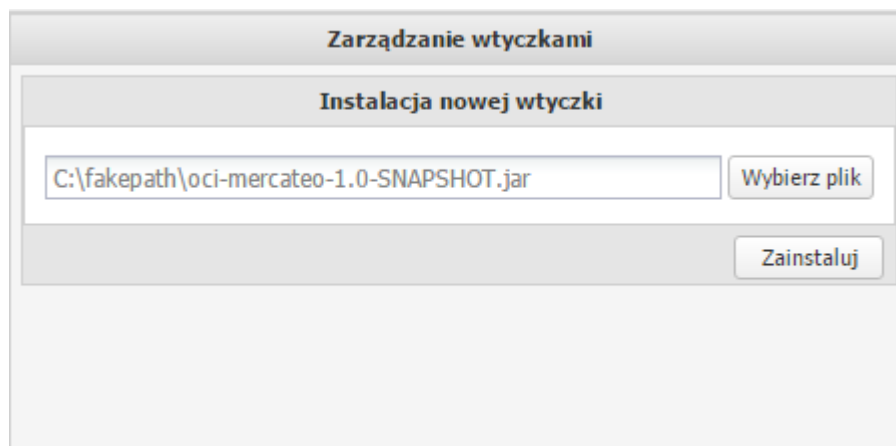
Dla poprawnego działania modułu system Plus Workflow powinien być w wersji co najmniej 3.1.56.

Rozdział 2. Instalacja modułu Plus Workflow for OCI

Aby móc zainstalować moduł Plus Workflow for OCI należy posiadać uprawnienie do Administracji.

W celu zainstalowania modułu Plus Workflow for OCI należy:

1. Pobrać wtyczkę z repozytorium Nexus - [oci-mercateo-1.0-20161014.105116-2.jar](#), dostarczoną przez Producenta.
2. W systemie Plus Workflow przejść do widoku Administracja -> Konfiguracja systemu -> Wtyczki.
3. W sekcji Zarządzanie wtyczkami należy wybrać pobrany plik i kliknąć przycisk Zainstaluj (patrz rysunek poniżej).






Rysunek 1. Okno zarządzania wtyczkami.

Rozdział 3. Konfiguracja akcji formularza modułu w PWE

Po instalacji wtyczki dostępna jest nowa akcja w Plus Workflow Editor (PWE). Akcja dostępna jest w kategorii Integracja z OCI pod nazwą „Idź do Mercateo”. Akcja użyta może być tylko na przycisku.

Definicja akcji

Poniżej przedstawiono parametry jakie mogą zostać zdefiniowane w akcji. Każdy parametr wiąże zmienną z formularza z daną jaka została zwrócona ze sklepu w formacie OCI.

Załącznik 	<input type="text" value="Załącznik"/>	+ 
Tytuł załącznika 	<input type="text" value="Tytuł załącznika"/>	+ 
CBA 	<input type="text" value="CBA"/>	+ 
Waluta 	<input type="text"/>	+ 
Czas dostawy 	<input type="text"/>	+ 
Opis 	<input type="text" value="Opis"/>	+ 
Identyfikator koszyka 	<input type="text" value="Identyfikator koszyka"/>	+ 
Liczba przedmiotów w koszyku 	<input type="text"/>	+ 
Numer przedmiotu w koszyku 	<input type="text"/>	+ 
Długi opis 	<input type="text"/>	+ 
Producent 	<input type="text"/>	+ 
Cena 	<input type="text" value="Cena"/>	+ 
Jednostka ceny 	<input type="text"/>	+ 
Ilość 	<input type="text" value="Ilość"/>	+ 
Usługa 	<input type="text"/>	+ 
Podatek 	<input type="text"/>	+ 
Jednostka 	<input type="text"/>	+ 
Sprzedawca 	<input type="text"/>	+ 
Numer towaru sprzedawcy 	<input type="text"/>	+ 

Rysunek 2. Parametry definiowane w akcji.

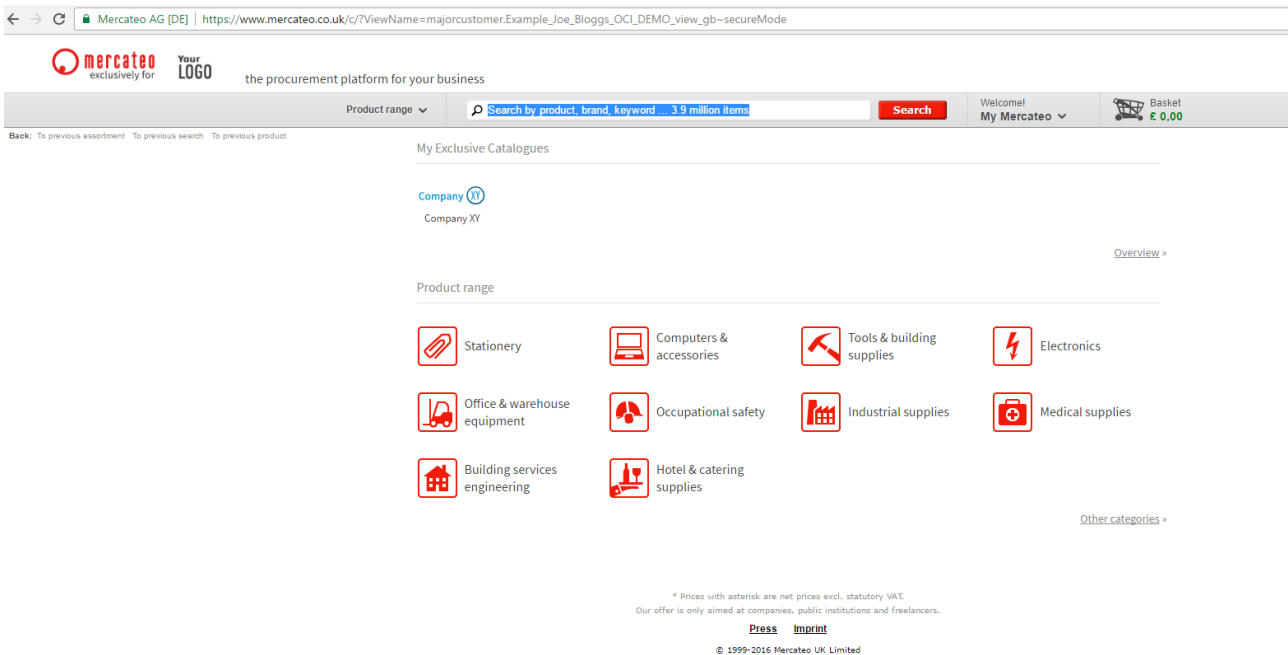
Nie jest wymagane aby wszystkie zmienne były mapowane.

Załącznik	Adres URL załącznika związanego z produktem
Tytuł załącznika	Tytuł załącznika związanego z produktem. W przypadku braku mapowania tego pola wykorzystana zostanie nazwa pliku załącznika.
CBA	Numer CBA
Waluta	Waluta ceny produktu
Czas dostawy	Czas dostawy w dniach
Opis	Krótki opis produktu
Identyfikator koszyka	Identyfikator koszyka ze sklepu Mercateo
Liczba przedmiotów w koszyku	Liczba przedmiotów jakie znajdują się w koszyku
Numer przedmiotu w koszyku	Numer przedmiotu znajdującego się w koszyku
Długi opis	Długi opis przedmiotu
Producent	Producent przedmiotu
Cena	Łączna cena netto produktu
Jednostka ceny	?
Ilość	Ilość produktów
Usługa	Określa czy dana pozycja jest usługą czy produktem (1 - produkt, 0 - usługa)
Podatek	Stawka podatku
Jednostka	?
Sprzedawca	Identyfikator sprzedawcy
Numer towaru sprzedawcy	Identyfikator towaru w systemie sprzedawcy

Opis przebiegu akcji dla sklepu Mercateo


Po kliknięciu przycisku na którym została zdefiniowana akcja następuje przekierowanie na stronę sklepu Mercateo.

Logowanie następuje automatycznie zgodnie z ustawieniami hasła w sekcji Ustawienia OCI.



Back: [To previous assortment](#) [To previous search](#) [To previous product](#)

My Exclusive Catalogues

Company 
Company XY [Overview »](#)

Product range

- Stationery
- Computers & accessories
- Tools & building supplies
- Electronics
- Office & warehouse equipment
- Occupational safety
- Industrial supplies
- Medical supplies
- Building services engineering
- Hotel & catering supplies

[Other categories »](#)

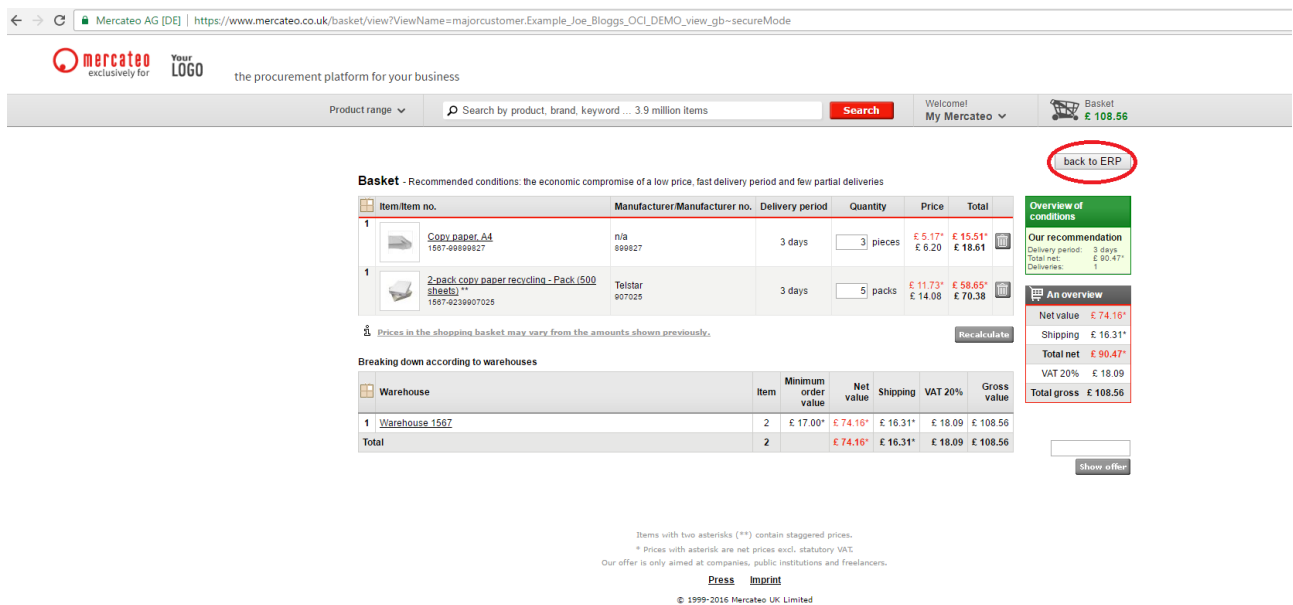
* Prices with asterisk are net prices excl. statutory VAT.
Our offer is only aimed at companies, public institutions and freelancers.

[Press](#) [Imprint](#)

© 1999-2016 Mercateo UK Limited

Rysunek 3. Strona internetowa sklepu Mercateo.

Po wybraniu przedmiotów i przejściu do koszyka wyświetlony zostaje przycisk „back to ERP”. Opis zmiany nazwy przycisku znajduje się na stronie Ustawienia OCI.



Basket - Recommended conditions: the economic compromise of a low price, fast delivery period and few partial deliveries

Item no.	Manufacturer/Manufacturer no.	Delivery period	Quantity	Price	Total
1	Copy paper A4 1587-99898827	n/a 899827	3 days 3 pieces	£ 5.17* £ 6.20	£ 15.51* £ 18.61
1	2-pack copy paper recycling - Pack (500 sheets) ** 1587-9239907025	Telstar 907025	3 days 5 packs	£ 11.73* £ 14.08	£ 58.65* £ 70.38

Prices in the shopping basket may vary from the amounts shown previously.

Breaking down according to warehouses

Warehouse	Item	Minimum order value	Net value	Shipping	VAT 20%	Gross value
1 Warehouse 1567	2	£ 17.00*	£ 74.16*	£ 16.31*	£ 18.09	£ 108.56
Total	2		£ 74.16*	£ 16.31*	£ 18.09	£ 108.56

[Recalculate](#)

[back to ERP](#)

Our recommendation
Delivery period: 3 days
Total net: £ 90.47*
Deliveries: 1

An overview
Net value: £ 74.16*
Shipping: £ 16.31*
Total net: £ 90.47*
VAT 20%: £ 18.09
Total gross: £ 108.56

[Show offer](#)

Items with two asterisks (**) contain staggered prices.
* Prices with asterisk are net prices excl. statutory VAT.
Our offer is only aimed at companies, public institutions and freelancers.

[Press](#) [Imprint](#)

© 1999-2016 Mercateo UK Limited

Rysunek 4. Przycisk "back to ERP".

Kliknięcie przycisku powoduje powrót do systemu PlusWorkflow oraz wczytanie listy przedmiotów zamówienia do procesu.

Rozdział 4. Ustawienia OCI

Do poprawnego działania akcji integracji z OCI należy najpierw skonfigurować podstawowe dane do poprawnego działania.

Zakładka Ustawienia OCI znajduje się w Administracja -> Ustawienia systemu.

Ustawienia podstawowe

Dane podstawowe	
Adres do sklepu:	<input type="text" value="https://www.mercateo.com/basket/oci"/>
Login do sklepu:	<input type="text" value="3452453"/>
Domyślne hasło do sklepu:	<input type="text" value="qwaszx"/>
Nazwy przycisku powrotu:	<input type="text" value="Do Plusworkflow"/>
Tryb bezpieczny:	<input checked="" type="checkbox"/>
<input type="button" value="Zapisz"/>	

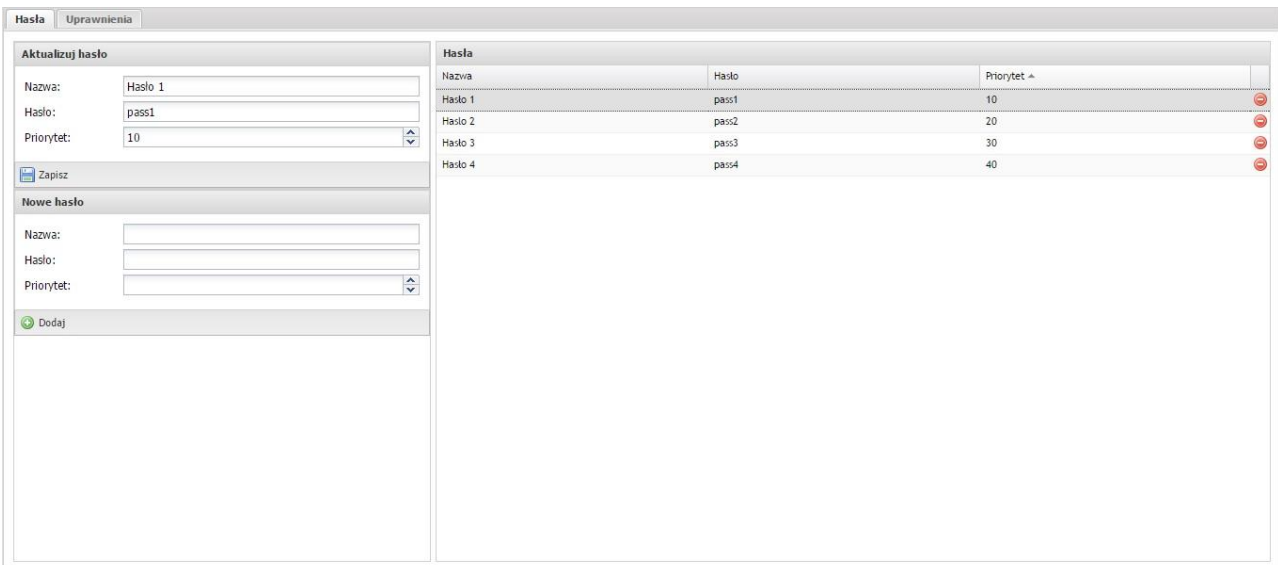
Rysunek 5. Okno danych podstawowych.

Minimalna potrzebna konfiguracja do prawidłowego działania integracji wymaga uzupełnienia danych w sekcji „Dane podstawowe”:

- Adres do sklepu - adres URL do sklepu z którym ma być przeprowadzona integracja OCI.
- Login do sklepu - login użytkownika w sklepie.
- Domyślne hasło do sklepu - hasło do domyślnego katalogu sklepu dla powyższego loginu.
- Nazwa przycisku powrotu - nazwa przycisku powrotu do systemu PlusWorkflow na stronie sklepu.
- Tryb bezpieczny - określa czy połączenie ze sklepem ma się odbyć za pomocą protokołu https (zalecane).

Zarządzanie hasłami

Sklep Mercateo umożliwia dostęp do różnych katalogów za pomocą różnych haseł. Zarządzanie hasłami odbywa się w zakładce „Hasła”.



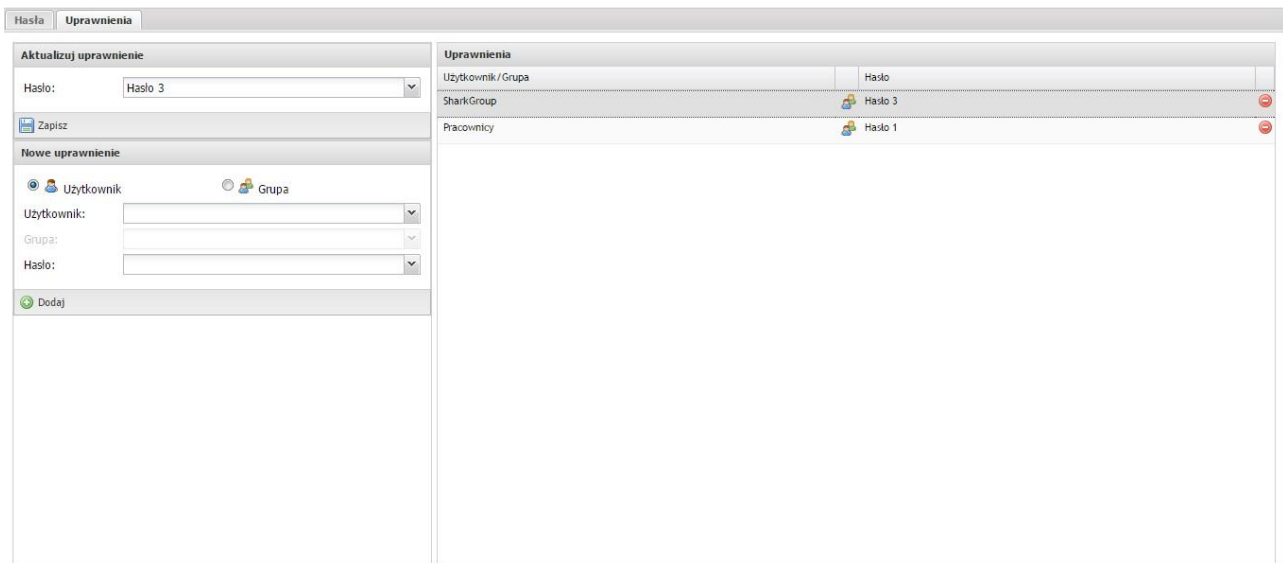
Nazwa	Hasło	Priorytet ▲
Hasło 1	pass1	10
Hasło 2	pass2	20
Hasło 3	pass3	30
Hasło 4	pass4	40

Rysunek 6. Zakładka "Hasła".

Zarządzanie uprawnieniami

Hasło jakim ma się logować użytkownik może zostać przypisane do samego użytkownika lub do grupy do której należy użytkownik.

Hasła można przypisać w zakładce „Uprawnienia”.



Rysunek 7. Zakładka "Uprawnienia".

Kolejność pobierania haseł

1. Hasło przypisane do użytkownika.
2. Hasło przypisane do grupy użytkownika o najwyższym priorytecie (im niższa wartość liczbowa tym wyższy priorytet)
3. Hasło domyślne

Przykłady

Załóżmy, że w systemie mamy zdefiniowane następujące hasła:

Hasło1	10
Hasło2	20

Następujące grupy:

- SharkGroup,
- Pracownicy.

Oraz następujących pracowników:

- admin,
- jnowak.

Przykład 1

Użytkownik jnowak nie należy do żadnej grupy i nie ma przypisanego żadnego hasła. Wówczas zwrócone zostanie hasło domyślne.

Przykład 2

Użytkownik jnowak należy do grupy Pracownicy, która ma przypisane hasło Hasło1. Wówczas zwrócone zostanie wartość hasła Hasło1.

Przykład 3

Użytkownik admin należy do grup SharkGroup i Pracownicy. Grupy mają przypisane następujące hasła:

- SharkGroup Hasło1
- Pracownicy Hasło2

Wówczas zwrócone zostanie Hasło1, ponieważ ma wyższy priorytet (niższa wartość liczbowa to wyższy priorytet).

Przykład 4

Użytkownik admin należy do grupy SharkGroup, która ma przypisane hasło Hasło1. Użytkownik ma także przypisane hasło Hasło2.

Wówczas zwrócone zostanie Hasło2, ponieważ jest przypisane bezpośrednio do użytkownika.

Wtyczka

Wtyczka jest bundlem OSGi instalowanym w środowisku OSGi. Może być ona dynamicznie:

- instalowana,
- aktualizowana,
- uruchamiana,
- zatrzymywana,
- odinstalowana.

Wtyczka posiada własny kontekst, który pełni rolę kontenera komponentów wtyczki, serwisów, kontrolerów. Możliwe jest wstrzykiwanie zależności i korzystanie ze wszystkich dobrodziejstw SpringFramework.

Deskryptor wtyczki (suncode-plugin.xml)

Deskryptor wtyczki to plik XML który dostarcza podstawowe informacje o wtyczce oraz jest miejscem na deklaracje wykorzystywanych modułów. Plik ten jest wymagany, ponieważ dostarcza informacje o wtyczce (jej unikalny identyfikator i wyświetlaną nazwę).

Deskryptor ma następującą strukturę:

suncode-plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin key="com.suncode.plugin-tutorial" name="Tutorial Plugin">
  <plugin-details>
    <description>
      <localized language="en">Description</localized>
      <localized language="pl">Opis</localized>
    </description>
    <author>Suncode</author>
  </plugin-details>

  <!-- Wszystkie kolejne elementy stanowią deklaracje modułów -->
</plugin>
```

Wtyczka może także zdefiniować PluginHook, jeżeli potrzebne jest wywołanie akcji przy starcie i zatrzymaniu wtyczki:

suncode-plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin key="com.suncode.plugin-tutorial" name="Tutorial Plugin"
hook="com.suncode.plugin.tutorial.Hook">
  <!-- ... -->
</plugin>
```

Plik suncode-plugin.xml musi znajdować się w głównym katalogu pliku jar.

Operacje na wtyczkach

Ten rozdział pokazuje, jak korzystać z API mechanizmu wtyczek. Rozdział Zarządzanie wtyczkami pokazuje, jak zarządzać nimi z poziomu systemu.

- Javadoc
- JavaDoc API znajduje się tutaj: <http://192.168.1.52/javadoc/plugin-framework/1.0.0/api/>

Głównym komponentem mechanizmu wtyczek jest PluginFramework. W systemie PlusWorkflow musimy pobrać ten obiekt z kontekstu aplikacji:

- wykorzystać wstrzykiwanie zależności

```
@Component
public class SomeComponent {
    @Autowired
    private PluginFramework framework;

    /**
     * ...
     */
}
```

- pobrać obiekt statycznie z kontekstu aplikacji

```
import com.suncode.plugin.framework.PluginFramework;
import com.suncode.pwfl.util.SpringContext;

public class SomeClass {
    public static void doSomething() {
        PluginFramework framework = SpringContext.getBean(
PluginFramework.class );
        /**
        ...
        */
    }
}
```

Poniżej przedstawiono wykonywanie kilku podstawowych operacji:

```
// pobrany w dowolny sposób
// PluginFramework framework = ...

// 1. Instalacja wtyczki z podanego pliku
File pluginFile = new File("/fakepath");
Plugin plugin = framework.installPlugin( pluginFile );

// 2. Aktualizacja wtyczki
File updatedPluginFile = new File("/fakepath");
plugin.update( updatedPluginFile );

// 3. Uruchomienie wtyczki
plugin.start();
plugin.getState(); // PluginState.ACTIVE

// 4. Pobranie tłumaczenia
// zwraca "message1" jeżeli moduł I18N nie jest obecny, w przeciwnym wypadku
zwraca znalezione tłumaczenie
plugin.getMessage("message1");

// 5. Zatrzymanie wtyczki
plugin.stop();
plugin.getState(); // PluginState.STOPPED
```

Rozdział 5. Konfiguracja kontekstu wtyczki

Wstęp

Każda wtyczka ma własny kontekst aplikacji ([ApplicationContext](#)), w którym rejestrowane są jej komponenty, kontrolery, importowane serwisy etc. Dzięki temu wtyczka może być pisana tak jak każda inna aplikacja wykorzystująca [SpringFramework](#).

Kontekst może być konfigurowany na 2 sposoby:

1. przez plik plugin-context.xml w katalogu /META-INF/spring/
2. przez klasę z adnotacją @Configuration

Konfiguracja XML

Jeżeli wtyczka zawiera plik /META-INF/spring/plugin-context.xml, to na jego podstawie tworzony jest XmlOsgiPluginContext.

Jeżeli projekt buduje Maven, plik powinien znajdować się w src/main/resources/META-INF/spring/plugin-context.xml

Plik konfiguracji jest standardowym plikiem konfiguracji kontekstu aplikacji springframework i może wyglądać następująco:

plugin-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd">

  <!-- Włączenie skanowania wtyczki w poszukiwaniu @Component -->
  <context:component-scan base-package="com.suncode.plugin.tutorial" />

</beans>
```

Konfiguracja poprzez plik XML ma swoje plusy i minusy. W ostateczności wszystko zależy od gustu autora wtyczki.

Konfiguracja Java

Od SpringFramework 3 możliwa jest konfiguracja kontekstu aplikacji w całości używając klas Java oznaczonych adnotacją @Configuration.

Jeżeli mechanizm wtyczek nie znajdzie pliku /META-INF/spring/plugin-context.xml to stworzy domyślny kontekst aplikacji, który skanuje całą wtyczkę w poszukiwaniu klas @Component (a więc również @Configuration).

Klasa konfiguracyjna w takim przypadku nie jest w ogóle potrzebna, jednak zazwyczaj chcemy dodatkowe funkcjonalności jak dostęp do bazy danych, transakcyjność etc.

Przykładowa konfiguracja (rejestruje tylko 1 bean klasy String):

Java Configuration

```
package com.suncode.plugin.tutorial;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class Config {
    @Bean
    public String someString() {
        return "asd";
    }
}
```

Spis rysunków

Rysunek 1. Okno zarządzania wtyczkami.	3
Rysunek 2. Parametry definiowane w akcji.....	4
Rysunek 3. Strona internetowa sklepu Mercateo.....	6
Rysunek 4. Przycisk "back to ERP".	6
Rysunek 5. Okno danych podstawowych.....	7
Rysunek 6. Zakładka "Hasła".	8
Rysunek 7. Zakładka "Uprawnienia".	9

Notatki

