

## 02. Rozszerzenia aplikacji redaktora i administratora

### Wprowadzenie

Rozszerzenia aplikacji redaktora i administratora bazują na [Java Plugin Framework](#) (JPF). Poniżej opisane są wszystkie możliwe rozszerzenia (punkty rozszerzeń) zdefiniowane dla aplikacji redaktora i administratora dLibra.

- 1 [Rozszerzenie metadata](#)
- 2 [Rozszerzenie fileParser](#)
- 3 [Rozszerzenie interfaceLanguage](#)
- 4 [Rozszerzenie graphicProvider](#)
- 5 [Rozszerzenie sourceLocator](#)
- 6 [Rozszerzenie dictionaryManager](#)
- 7 [Rozszerzenie filesHandler](#)
- 8 [Rozszerzenie tool](#)
- 9 [Rozszerzenie miniatureProvider](#)
- 10 [Rozszerzenie eventListener](#)
- 11 [Rozszerzenie objectPanel](#)
- 12 [Rozszerzenie dataSource](#)



#### Uwaga

Interfejsy programistyczne wyszczególnione w poniższych opisach znajdują się w bibliotece programistycznej [dcore-app-extension-api](#).

### Rozszerzenie metadata

Rozszerzenie **metadata** służy do poszerzania możliwości importujących i eksportujących metadane w aplikacji. Poprzez to rozszerzenie możemy dodać funkcję importowania lub/i eksportowania metadanych z/do zewnętrznego formatu.

Rozszerzenie przyjmuje dwa parametry:

- class - jest to nazwa klasy, która implementuje odpowiedni interfejs programistyczny tego rozszerzenia
- name - nazwa rozszerzenia

Interfejs programistyczny (język Java) dla tego rozszerzenia to [pl.psnc.dlibra.app.extension.metadata.MetadataFinder](#). Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Import metadanych z formatu MARC ([dlibra-app-extension-mf-marc](#)).
- Import metadanych z formatu XML ([dcore-app-extension-mf-xml](#)).
- Import metadanych z formatu BibTeX ([dlibra-app-extension-mf-bibtex](#)).
- Eksport w formacie RDF - to rozszerzenie jest zintegrowane z aplikacją redaktora i administratora - nie jest zdefiniowane w odrębnym projekcie.

Opis konfiguracji poszczególnych rozszerzeń można znaleźć [tutaj](#).

### Rozszerzenie fileParser

Rozszerzenie to odnajduje dla określonego pliku powiązane z nim inne pliki. W aplikacji redaktora i administratora po wybraniu pliku głównego (np. w kreatorze publikacji) następuje automatyczne sprawdzenie jakie inne pliki są a nim powiązane. W tym celu wykorzystywane są rozszerzenia **fileParser**. Domyślnie w aplikacji zainstalowane są rozszerzenia HTML oraz DjVu. Jeśli dla wybranego pliku nie ma zdefiniowanego rozszerzenia to redaktor musi sam zdecydować o tym, jakie pliki będą stanowić zawartość publikacji.

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to [pl.psnc.dlibra.app.extension.fileparser.FileParser](#). Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Format DjVu - rozszerzenie odnajduje pliki powiązane z danym plikiem w formacie DjVu ([dcore-app-extension-fp-djvu](#)).
- Format HTML - rozszerzenie odnajduje pliki powiązane z danym plikiem w formacie HTML ([dcore-app-extension-fp-html](#)).

### Rozszerzenie interfaceLanguage

Rozszerzenie to pozwala na dodawanie tłumaczeń interfejsu użytkownika aplikacji redaktora i administratora. Domyślnie w aplikacji zainstalowane są rozszerzenia dla języka polskiego, angielskiego i ukraińskiego. Poprzez to rozszerzenie można dodać tłumaczenie interfejsu w innych językach. Aby otrzymać listę fraz/słów do tłumaczenia prosimy o [kontakt](#).

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.language.InterfaceLanguage`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Polski - rozszerzenie udostępnia wszystkie słowa/frazy interfejsu użytkownika w języku polskim ([dlibra-app-extension-il-pl](#)).
- Angielski - rozszerzenie udostępnia wszystkie słowa/frazy interfejsu użytkownika w języku angielskim ([dlibra-app-extension-il-en](#)).

## Rozszerzenie `graphicProvider`

Rozszerzenie to pozwala na zmianę wyglądu elementów graficznych w aplikacji redaktora i administratora. Poprzez dodanie rozszerzenia tego typu można poprawić ikonki i obrazki widoczne w aplikacji tak, by lepiej pasowały do kontekstu (kulturowego lub technicznego), w którym będzie wykorzystywany system dLibra. Zalecane jest, aby w aplikacji zainstalowana była co najwyżej jedna wtyczka tego typu, ponieważ nie jest dostępny mechanizm wyboru kontekstu graficznego, analogiczny do wyboru języków.

W aplikacji domyślnie zainstalowana jest wtyczka zapewniająca domyślny wygląd ([dlibra-app-extension-gp](#)).

## Rozszerzenie `sourceLocator`

Rozszerzenie to pozwala na lokalizację plików z metadanymi, które później będą importowane. Korzystając z tego rozszerzenia możliwe jest wprowadzenie nowej funkcji lokalizacji pliku metadanych.

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.sourcelocator.SourceLocator`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Z39.50 - pozwala na lokalizowanie plików metadanych przy użyciu serwerów Z39.50 ([dlibra-app-extension-sl-z3950](#)).

Opis konfiguracji poszczególnych rozszerzeń można znaleźć [tutaj](#).

## Rozszerzenie `dictionaryManager`

Rozszerzenie to pozwala na importowanie i eksportowanie grup wartości atrybutów z i do zewnętrznych plików.

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.dictionarymanager.DictionaryManager`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Import grup wartości z plików MARC - pozwala na odczyt plików w formacie MARC-21 ([dlibra-app-extension-dl-marc](#)). Zapis do plików nie jest wspierany.
- Import/export grup wartości z/do plików XML - pozwala na zapis i odczyt plików w postaci XML. Można je wykorzystać do kopiowania wartości atrybutów pomiędzy bibliotekami ([dcore-app-extension-dm-dictcopy](#)).

Opis konfiguracji poszczególnych rozszerzeń można znaleźć [tutaj](#).

## Rozszerzenie `filesHandler`

To rozszerzenie pozwala zdefiniować procedury, które będą realizowane po wybraniu pliku głównego w kreatorze nowej publikacji. Procedury te mogą obejmować przetwarzanie plików na dysku, wyświetlanie okien dialogowych i inne. Skutkiem działania procedury może być zmiana wybranego pliku głównego oraz pozostałych plików dodanych do wydania. Jeśli w aplikacji zainstalowanych jest więcej niż jedna wtyczka do tego rozszerzenia, są one odpytywane po kolei, aż do pierwszej wtyczki, która przetworzy pliki poprawnie (nie wyrzuci wyjątku).

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.fileshandler.FilesHandler`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Prezentacje JPG - jeśli wybrano folder zawierający pliki .jpg, ta wtyczka pozwala na zdefiniowanie kolejności obrazków i przypisanie opisu do każdego obrazka. Informacje te są zapisywane w pliku PresentationData.xml, który jest oznaczany jako plik główny ([dcore-app-extension-fh-jpg](#)).

## Rozszerzenie tool

To rozszerzenie pozwala dodać nowe elementy do menu Narzędzia w aplikacji administratora lub redaktora, jak pokazano na poniższym rysunku.

Rozszerzenie to przyjmuje dwa parametry:

- class, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to [pl.psync.dlibra.app.extension.tool.Tool](#). Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).
- type, który określa przeznaczenie narzędzia, tj. czy jest to narzędzie aplikacji administratora czy aplikacji redaktora. W związku z tym parametr ten może przyjąć tylko dwie wartości: ADMIN (określający aplikację administratora) oraz EDITOR (określający aplikację redaktora). W efekcie ustawienie parametru sprawi, że narzędzie będzie wyświetlane w menu „Narzędzia” w aplikacji administratora, bądź aplikacji redaktora.

Lista domyślnie zainstalowanych rozszerzeń tego typu jest następująca:

- Lista publikacji w publikacji grupowej - zapisuje publikacje należące do wybranej publikacji grupowej w pliku html ([dcore-app-extension-tl-leaflist](#)).
- Narzędzie do czyszczenia słownika atrybutu - pozwala usunąć ze słownika wartości, które nie są przypisane do żadnej publikacji. To rozszerzenie jest zintegrowane bezpośrednio z aplikacją administratora - nie jest zdefiniowane w odrębnym projekcie.

Oprócz powyższych rozszerzeń, zespół rozwijający system dLibra przygotował dla programistów zainteresowanych tworzeniem własnych rozszerzeń [narzędzie przykładowe](#), ilustrujące możliwości rozszerzeń typu tool.

## Rozszerzenie miniatureProvider

To rozszerzenie pozwala na automatyczne generowanie miniatur dla dodawanych publikacji na podstawie plików wybranych do wydania. Możliwe jest wygenerowanie kilku miniatur dla publikacji, z których redaktor będzie mógł wybrać jedną.

Rozszerzenie to przyjmuje jeden parametr class, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to [pl.psync.dlibra.app.extension.miniatureprovider.MinatureProvider](#). Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Domyślnie zainstalowaną wtyczką, która implementuje to rozszerzenie, jest wtyczka tworząca prezentacje na podstawie plików jpg (zobacz [opis rozszerzenia filesHandler](#)). Poza tym, w aplikacji redaktora zintegrowano wtyczki pozwalające na generowanie miniatur z następujących plików:

- najpopularniejsze formaty graficzne (jpg, png, gif)
- pliki djVu
- pliki pdf

## Rozszerzenie eventListener

Rozszerzenie typu **eventListener** pozwala reagować na zdarzenia zachodzące w aplikacji redaktora i administratora.

Rozszerzenie to reprezentowane jest w oprogramowaniu dLibra jako interfejs programistyczny języka Java [pl.psync.dlibra.app.extension.eventListener.EventListener](#).

Interfejs ten posiada jedną metodę o nazwie eventPushed przyjmującą następujące parametry:

- event - reprezentuje zdarzenie, które miało miejsce w aplikacji redaktora i administratora. Parametr ten jest implementacją interfejsu [pl.psync.dlibra.app.extension.eventlistener.AppEvent](#), który umożliwia określenie identyfikatora obiektu jakiego zdarzenie dotyczy oraz typu zdarzenia, który może być jedną z wartości określonych w typie wyliczeniowym [pl.psync.dlibra.app.extension.eventlistener.AppEventType](#).
- serverInterface - reprezentuje interfejs dostępu do serwera dLibry, za pośrednictwem którego można odwołać się do usług umożliwiających pobranie danych dotyczących obiektów w dLibrze.

W domyślnej konfiguracji aplikacji redaktora i administratora nie ma rozszerzeń tego typu. Rozszerzenie to zaprojektowane zostało z myślą o narzędziach wykonujących specyficzne działania w zewnętrznych systemach, które z oczywistych względów nie są wspierane przez aplikację redaktora i administratora. Dla programistów chcących stworzyć narzędzie tego typu, powstało [przykładowe rozszerzenie](#).

## Rozszerzenie **objectPanel**

Rozszerzenia tego typu pozwalają na dodanie dodatkowej zakładki w aplikacji redaktora, widocznej w dolnej części ekranu po zaznaczeniu obiektu określonego typu. Wtyczka taka ma pełną swobodę w przygotowaniu zawartości zakładki i może w tym celu korzystać z dostępu do funkcji serwera. Rozszerzenie to może posłużyć do wyświetlania dodatkowych informacji związanych z wybranym obiektem, np. pobranych z zewnętrznych systemów.

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.objectpanel.ObjectPanel`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Domyślnie w aplikacji redaktora nie jest zainstalowane żadne rozszerzenie tego typu, ale można pobrać [przykładową implementację](#), która demonstruje możliwości rozszerzenia.

## Rozszerzenie **dataSource**

Ten typ rozszerzeń umożliwia dodanie nowych sposobów definiowania treści publikacji. W kreatorze nowej publikacji (a także dodawania plików do publikacji, oraz podmiany plików w publikacji), na stronie wyboru plików, użytkownik może wybrać jedno z zainstalowanych rozszerzeń **dataSource**, dzięki czemu zamiast standardowego panelu wyboru plików pojawi się panel przygotowany przez to rozszerzenie. Kiedy użytkownik wybierze pliki do publikacji w tym panelu, rozszerzenie przekazuje tę informację do aplikacji redaktora i użytkownik może przejść do następnych kroków kreatora. Przy pomocy tego rozszerzenia można wskazać nie tylko pliki na dysku lokalnym, ale też zasoby dostępne w Internecie - w takim przypadku serwer dLibry pobierze je i zapisze w swoim repozytorium (pobieranie może się również odbyć za pośrednictwem komputera redaktora).

Rozszerzenie to przyjmuje jeden parametr `class`, który specyfikuje nazwę klasy implementującej interfejs programistyczny rozszerzenia. Interfejs programistyczny (język Java) dla tego rozszerzenia to `pl.psnc.dlibra.app.extension.datasource.DataSource`. Szczegółowe informacje związane z tym interfejsem znajdują się w komentarzach metod i samego interfejsu (JavaDocs).

Domyślnie w aplikacji redaktora nie jest zainstalowane żadne rozszerzenie tego typu, ale można pobrać [przykładową implementację](#), która demonstruje możliwości rozszerzenia.