

## 03. Budowa aplikacji czytelnika systemu dLibra

### Budowa aplikacji czytelnika systemu dLibra

- [Podstawowe pojęcia związane z działaniem aplikacji czytelnika](#)
- [Opis plików konfiguracyjnych aplikacji czytelnika systemu dLibra 6.0](#)
  - [Pliki konfiguracyjne znajdujące się w katalogu dlibra-webapp/WEB-INF/conf](#)
- [Opis struktury pod dlibra-webapp\style](#)
- [Dodatkowe elementy](#)

### Podstawowe pojęcia związane z działaniem aplikacji czytelnika

**Strony** - aplikacja czytelnika zbudowana jest z wielu stron (widoków), np. strona główna, strona z informacjami o obiekcie, strona kontaktu. Są one podstawą nawigacji. Każda strona posiada swój unikalny identyfikator np.:

| Identyfikator | Strona                           | Przykładowy adres                                       |
|---------------|----------------------------------|---------------------------------------------------------|
| main          | strona główna                    | http://demo.dl.psnc.pl/dlibra                           |
| publication   | strona z informacjami o obiekcie | http://demo.dl.psnc.pl/dlibra/ <b>publication</b> /1000 |
| contact       | strona kontaktu                  | http://demo.dl.psnc.pl/dlibra/ <b>contact</b>           |

Każda z tych stron jest zdefiniowana w pliku **pages.xml** poprzez sekcję **<page>...</page>** np.:

| Identyfikator | Przykładowa definicja                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| main          | <div>pages.xml - main</div> <pre>&lt;page name="main" layout="home_layout" inherits="base-page"&gt; ... &lt;/page&gt;</pre>                     |
| publication   | <div>pages.xml - publication</div> <pre>&lt;page name="publication" layout="home_layout" inherits="base-page"&gt; ... &lt;/page&gt;</pre>       |
| contact       | <div>pages.xml - contact</div> <pre>&lt;page name="contact" layout="home_layout" secure="true" inherits="base-page"&gt; ... &lt;/page&gt;</pre> |

Znaczniki **page** posiadają 3 wyróżniające się pola:

**name** - identyfikator strony np.: main, publication, contact,

**layout** - nazwa layoutu (układu strony), na którym osadzone będą wszystkie komponenty dla danej strony,

**inherits** - identyfikator strony(może być abstrakcyjny), z której definiowana strona będzie pobierać komponenty oprócz komponentów własnych (np.: Strona kontaktu jest zdefiniowana w sekcji contact. Zawiera komponent ContactComponent, ale pole inherits ma wartość base-page dlatego strona kontaktu będzie zawierać również wszystkie komponenty jakie zostały zdefiniowane dla sekcji base-page ).

**Komponenty** - odpowiadają poszczególnym elementom strony np. sekcja z ostatnio dodanymi obiektami na stronie głównej to odrębny komponent. Komponenty mogą być wyświetlane w różnych miejscach w obrębie strony. To jakie komponenty są wyświetlane na stronie należy wyspecyfikować w deskrytorze danej strony w pliku pages.xml. Sposób definiowania komponentu pokazano poniżej:

#### pages.xml - component

```
<component name="...">
    <place>...</place>
    <position>...</position>
</component>
```

Znacznik component charakteryzujemy poprzez następujące atrybuty:

**name** - pełna nazwa klasy komponentu;

**place** - nazwa umiejscowienia w layoutcie;

**position** - pozycja, na której ma być umieszczony komponent w zadanym miejscu(**place**), musi mieć wartość unikalną, jeśli inny komponent umieszczono w tym samym miejscu.

Przykładowym komponentem może być komponent na stronie kontaktu. Strona ta jest zdefiniowana w elemencie page **contact**. Wykorzystuje layout o nazwie **home\_layout**. Posiada komponent zdefiniowany następująco:

#### pages.xml

```
<components>
    <component name="pl.psnc.dlibra.web.comp.pages.components.ContactComponent">
        <place>main</place>
        <position>1</position>
    </component>
</components>
```

To oznacza, że zawartość komponentu **ContactComponent** umiejscowionego w pliku **/WEB-INF/components/templates/ContactComponent.vm** jest renderowana w szablonie rozmieszczeń **/WEB-INF/layouts/templates/home\_layout.vm** w pętli:

#### home\_layout.vm

```
#foreach( $comp in $main)
    ${comp.RenderedTemplate}
#end
```

Każdy komponent składa się z trzech elementów:

- klasy Javowej (np. **pl.psnc.dlibra.web.comp.pages.components.IndexSearchComponent**),
- pliku szablonu(plik szablonu nazywa się tak samo jak klasa Javowa np. **IndexSearchComponent.vm**),
- plików z różnymi wersjami językowymi (np. **IndexSearchComponent\_xx.xml**).

W wersji 6 wszystkie komponenty znajdują się w archiwum **dcore-webapp-components-6.x.x.jar** (znajduję się on : **/WEB-INF/lib**). Klasa Javowa znajduje się w katalogu odpowiadającym jej nazwie pakietowej (w tym wypadku **/pl/psnc/dlibra/web/comp/pages/components/**) natomiast pozostałe pliki komponentu - w głównym katalogu tego archiwum.

Istnieje możliwość "przykrycia" domyślnych plików z archiwum JAR (**dcore-webapp-components-6.x.x.jar**) własnymi zmienionymi plikami. W tym celu najlepiej wypakować pliki szablonów, które mają być zmienione do odpowiedniego katalogu. W przypadku plików \*.vm jest to katalog **/WEB-INF/components/templates**, natomiast dla plików językowych \*.xml jest to katalog **/WEB-INF/components/resources**.

Tak wypakowane pliki można zmieniać i zmiany w nich "nadpisać" oryginalną zawartość plików pozostawionych w archiwum **jar**. Zmiany w plikach \*.vm będą widoczne natychmiastowo po ich zapisaniu. Zmiany w plikach \*.xml będą uwzględniane również natychmiastowo, pod warunkiem, że pliki te były już wypakowane przy starcie serwera Tomcat. W przeciwnym razie należy zrestartować Tomcat.

Dodatkowo w niektórych szablonach komponentów używane są makra, zdefiniowane w pliku **components\_library.vm** lub **layout\_library.vm**. Plik **components\_library.vm** znajduje się w archiwum **dcore-web-components-6.x.x.jar**. Własne makra oraz zmodyfikowane makra powinny zostać umieszczone adekwatnie w plikach **/WEB-INF/components/templates/custom\_library.vm** lub **/WEB-INF/layout/custom\_templates/custom\_layout\_library.vm**). Zmiany w tych plikach również powinny być uwzględniane zaraz po ich zapisaniu pod warunkiem, że edytowane przez nas makro było już dostępne przy starcie serwera Tomcat.

## Opis plików konfiguracyjnych aplikacji czytelnika systemu dLibra 6.0





Proszę pamiętać, aby edytować pliki konfiguracyjne za pomocą edytora, który wspiera kodowanie UTF-8 i nie umieszcza na początku pliku znacznika BOM np. darmowego edytora JEdit.

Poniżej znajduje się opis najważniejszych plików konfiguracyjnych aplikacji czytelnika systemu dLibra. Zmiany dokonywane w niektórych plikach wymagają restartu aplikacji czytelnika, pliki te zostały oznaczone (bezpośrednio po nazwie umieszczono **(R)**). Pliki nie ujęte w poniższym zestawieniu nie powinny być modyfikowane bez uprzedniej konsultacji z twórcami oprogramowania dLibra.

- **actions.xml** - W pliku tym znajdują się dodatkowe parametry konfiguracyjne dla akcji. Można tu określić między innymi:
  - adres email na jaki mają być wysyłane wiadomości wysyłane przez użytkowników z formularza kontaktowego oraz format tej wiadomości (plain/text czy html),
  - konfiguracja zabezpieczeń mechanizmu proponowania słów kluczowych przez użytkowników (tylko dla wdrożeń, które mają dedykowane rozwiązania w tym zakresie):
    - czy propozycje mają być domyślnie odrzucane czy akceptowane,
    - maksymalną i minimalną długość propozycji słów kluczowych,
    - maksymalną liczbę propozycji, które może wprowadzić użytkownik w ciągu zadanego czasu (domyślnie maksymalnie 5 propozycji w ciągu 30min).
  - maksymalną i minimalną długość loginu który wprowadzają użytkownicy w czasie rejestracji.
  - pola metadanych, po których użytkownik ma mieć możliwość sortowania wyników wyszukiwania.
- **components.xml** - Dodatkowe parametry konfiguracyjne dla poszczególnych komponentów. W tym pliku zmieniamy:
  - Identyfikator kolekcji oraz "Polecane".
  - Dla komponentu indeksów listę tokenów, które jako prefiksy wartości indeksów nie będą uwzględniane w sortowaniu.
  - Maksymalną długość zapytania wyszukiwawczego.
  - Możliwość wyświetlania formularza z zapytaniem o dostęp do publikacji chronionej.
  - Możliwość filtrowania po kolekcjach, publikacjach grupowych oraz dostępności w wynikach wyszukiwania.
  - Nazwy atrybutów, po których nie ma być możliwości przeszukiwania.
  - Liczbę elementów wyświetlanych na stronie głównej dla: polecanych, ostatnio dodanych i najpopularniejszych publikacji oraz wiadomości.
  - Identyfikatory kolekcji, których nie chcemy wyświetlać w strukturze kolekcji oraz to czy struktura kolekcji ma być pełnym drzewem.
  - Nazwy atrybutów, które chcemy wyświetlać w komponencie indeksów.
  - Adres email osoby odpowiedzialnej za administrację repozytorium OAI.
  - Podstawowe atrybuty w widoku informacji o obiekcie cyfrowym, możliwość wyświetlania identyfikatora OAI
- **ignored\_ips.txt** i **ignored\_agents.txt** (R) - Pliki te związane są zliczaniem statystyk odwiedzin. W pierwszym podajemy adresy IP, które nie mają być zliczane (np. komputery na których pracują redaktorzy). Każdy wpis jest traktowany jako wyrażenie regularne można więc podawać również całe zakresy adresów IP. W pliku znajduje się przykładowy wpis dla 127.0.0.1. Przed dodaniem nowego wyrażenia regularnego warto sprawdzić jego poprawność używając bezpłatnych narzędzi dostępnych online np. <http://regexpal.com/>.

Na podobnej zasadzie działa plik **ignored\_agents.txt**, należy umieścić tutaj wyrażenia regularne dla wartości pola **user-agents**. Pole to jest częścią nagłówka żądania HTTP, jest to łańcuch znaków identyfikujący program za pomocą którego przeglądana jest strona np. przeglądarka internetowa, robot wyszukiwarki. Jeżeli wartość łańcucha zostanie dopasowane do podanego wzorca to identyfikujące się za niego programy nie będą zliczane do statystyk odwiedzin i wyświetleń treści publikacji. W pliku znajduje się przykład dla googlebot'a. Listę identyfikatorów **user agents** można znaleźć w sieci.

- **pages.xml** - Plik zawiera definicję stron wyświetlanych przez aplikację czytelnika systemu dLibra. Modyfikacja tego pliku jest konieczna w przypadku:
  - zmiany kolejności komponentów wyświetlanych na danej stronie,
  - dodawania nowego komponentu do konkretnej strony,
  - tworzenia nowej strony w ramach aplikacji czytelnika.Zagadnienia związane z wykonywaniem zmian w tym pliku zostaną szczegółowo omówione w rozdziałach : "Dostosowywanie wyglądu aplikacji Czytelnika" oraz "Najczęściej zadawane pytania".
- **pages\_titles.xml** (R) - W pliku tym określamy tytuły i pod tytuły poszczególnych stron.
- **templates.properties** (R) - Zawiera parametry konfiguracyjne dostępne z poziomu wszystkich szablonów.
- **periodic.xml** (R) - Plik ten zawiera definicję zadań okresowych wykonywanych przez aplikację czytelnika. Konfigurujemy w nim:
  - czas wykonywania kopii zapasowej JCR, ścieżkę w której ma być przechowywana kopia zapasowa JCR oraz liczbę przechowywanych uprzednio stworzonych kopii,
  - mechanizmu importu wiadomości z zewnętrznego kanału RSS,
  - czas generowania i zapisu a także generowania widoków prezentacyjnych statystyk, identyfikatory atrybutów na podstawie których generowane są statystyki wartości atrybutów
  - czas wysyłania biuletynów z ostatnio dodanymi publikacjami,
  - położenie katalogu do którego zapisywane są informacje o zapytaniach wyszukiwawczych wydawanych przez użytkowników aplikacji czytelnika,
  - czas i mechanizm tworzenia mapy strony.
- pliki **dlibra-webapp\WEB-INF\components\resources\WEBAPP\_xx.xml** - Gdzie xx oznacza wersję językową (np. pl) w pliku tym znajdują się najczęściej modyfikowane etykiety tekstowe dla każdej z zainstalowanych wersji językowych.
- **dlibra-webapp\WEB-INF\components\templates\custom\_library.vm** - Zawiera dodatkowe makra języka VTL stworzone przez administratorów danej biblioteki dla komponentów. Informacje o tworzeniu makr języka VTL można znaleźć na stronach projektu Velocity.

- **dlibra-webapp\WEB-INF\layout\custom\_templates\custom\_layout\_library.vm** - Zawiera dodatkowe makra języka VTL stworzone przez administratorów danej biblioteki dla szablonów rozmieszczeń. Informacje o tworzeniu makr języka VTL można znaleźć na stronach projektu Velocity.
- Pliki **dlibra-webapp\WEB-INF\layout\resources\layout\_xx.xml** - Gdzie xx oznacza wersję językową (np. pl) w pliku tym znajdują się etykiety tekstowe używane w szablonach rozmieszczeń.
- Katalog **dlibra-webapp\WEB-INF\formats** - zawiera konfiguracje wszystkich rozszerzeń związanych z prezentacją treści publikacji. Więcej informacji znajduje się w sekcji [Rozszerzenia aplikacji czytelnika](#).

## Pliki konfiguracyjne znajdujące się w katalogu dlibra-webapp/WEB-INF/conf

Aplikacja czytelnika korzysta ze swojej lokalnej bazy danych, jest to baza zupełnie nie związana z bazą danych wykorzystywaną przez serwer dLibry. Na konfigurację tej bazy składają się poza omówionymi poniżej dwoma plikami również pliki jcr-init.properties oraz jcr-repository.xml. Nie powinny być one jednak pod żadnym pozorem modyfikowane.

- **jcr-init-desc.xml** (R) - specyfikacja podstawowej zawartości (wartości domyślne), która jest umieszczana w bazie danych podczas pierwszego startu aplikacji czytelnika.
- **jcr.properties** (R) - konfiguracja podstawowych parametrów dostępowych do bazy danych aplikacji czytelnika. W przypadku gdy katalog roboczy aplikacji czytelnika zostanie przeniesiony należy uaktualnić ścieżkę do bazy danych, która znajduje się w tym pliku.
- **services.properties** (R) - parametry połączeniowe do serwera dLibry. W przypadku gdy serwer dLibry zostanie przeniesiony na inną maszynę należy uaktualnić znajdujący się w tym pliku adres serwer.
- **smtp.properties** (R) - parametry dostępowe do serwera SMTP. Parametry te wykorzystywane są przez aplikację czytelnika do wysyłania wiadomości email.
- **cache.properties** (R) - zawiera ustawienia pamięci podręcznej (cache) aplikacji czytelnika. Użycie cache znacznie przyspiesza działanie aplikacji czytelnika wykorzystanie pamięci podręcznej jest niezbędne w przypadku bibliotek posiadających powyżej 1000 publikacji.
- **exlibris.xml** - treść tego pliku zawiera konfigurację mechanizmu exlibris. Więcej na temat konfiguracji exlibris można dowiedzieć się w sekcji konfiguracji Aplikacji Czytelnika.
- **pubcreator.properties** - plik podstawowych parametrów konfiguracji mechanizmu Self-Archiving. Więcej na temat konfiguracji Self-Archiving można znaleźć w sekcji konfiguracji Aplikacji Czytelnika. Dotyczy dedykowanych wdrożeń z włączoną funkcją Self-Archiving.
- **pubc-metadata.xml** - zawiera konfigurację formularza atrybutów dla funkcjonalności Self-Archiving. Więcej na ten temat można znaleźć w sekcji konfiguracji Aplikacji Czytelnika. Dotyczy dedykowanych wdrożeń z włączoną funkcją Self-Archiving.

## Opis struktury pod dlibra-webapp\style

W katalogu **dlibra-webapp\style** znajdują się większość grafik i arkuszy stylów css używanych w aplikacji czytelnika.

## Dodatkowe elementy

Poza standardowymi odwołaniami do stron zdefiniowanych w aplikacji czytelnika istnieją również zasoby dostępne z głównego poziomu:

- <http://demo.dl.psn.pl/Content> - odpowiedzialny za dostęp do treści publikacji.
  - Wywołanie `/Content/2952/` powoduje przejście do treści głównego pliku wydania o identyfikatorze 2952.
  - Wywołanie `/Content/2952/directory.djvu` powoduje pobranie pliku `directory.djvu` z wydania o identyfikatorze 2952.
- <http://demo.dl.psn.pl/zipContent> - odpowiedzialny za dostęp do treści publikacji w postaci archiwów ZIP. Wywołanie `/zipContent/2952/` powoduje pobranie treści całego wydania o identyfikatorze 2952 jako pliku zip.
- <http://demo.dl.psn.pl/image> - odpowiedzialny za dostęp do miniatur wydań. Wywołanie `/image?id=2940` powoduje pobranie miniaturki wydania o identyfikatorze 2940.
- <http://demo.dl.psn.pl/ka> - odpowiedzialny za szyfrowanie publikacji zabezpieczonych,
- <http://demo.dl.psn.pl/dlibra/dlibra-app.jnlp> - wpisując ten adres do przeglądarki uruchamiamy Aplikację Redaktora/Administratora.
- <http://demo.dl.psn.pl/dlibra/publication/81/content> - odnośnik do treści publikacji o identyfikatorze 81, jeżeli publikacja posiada więcej niż jedno wydanie wyświetlona zostanie lista wydań danej publikacji. W przypadku publikacji grupowej wyświetlona zostanie struktura publikacji grupowej.
- <http://demo.dl.psn.pl/dlibra/publication/81/> - odnośnik do opisu publikacji o identyfikatorze 81, jeżeli publikacja posiada więcej niż jedno wydanie wyświetlona zostanie lista wydań danej publikacji.