

# 04. Dostosowywanie wyglądu aplikacji czytelnika

## Dostosowywanie wyglądu aplikacji czytelnika

- Dodawanie oraz modyfikacje treści w komponencie
- Dodanie nowego komponentu do podstrony
- Usunięcie komponentu z podstrony
- Ustawienie komponentu w layoutcie
- Modyfikacja oraz dodawanie komunikatów językowych
- Podmiana istniejących obrazków (logo)
- Modyfikacje css

### Dodawanie oraz modyfikacje treści w komponencie

Większość danych wyświetlanych przez aplikację czytelnika jest pobierana z serwera systemu dLibra. Są to na przykład kolekcje, obiekty cyfrowe, wartości atrybutów itd. Istnieje jednak możliwość, aby dodać własne niestandardowe treści lub zmodyfikować sposób wyświetlania już istniejących w dowolnym miejscu wyświetlanego kodu html.

Przykład: Na stronie "Kontakt" chcemy umieścić obrazek ilustrujący zdjęcie pracowników instytucji.

1. Otwieramy stronę "Kontakt", sprawdzamy w pasku adresu nazwę dla strony kontakt np. dla adresu: <https://example.com/dlibra/contact?language=pl> nazwą strony w aplikacji czytelnika będzie **contact**.
2. Otwieramy plik **/WEB-INF/pages.xml** i odnajdujemy linię `<page name="contact" layout="home_layout" secure="true" inherits="base-page">`. W tej sekcji mamy wszystkie dostępne komponenty dla strony contact. Nas interesuje jedynie komponent **ContactComponent**. Sekcja **contact** oprócz swoich komponentów wykorzystuje również komponenty z sekcji **base-page** (ale tam nie chcemy robić żadnych zmian). Sekcja dla strony głównej np.: <https://example.com/dlibra/> nazywa się **main** (tutaj również nie chcemy robić zmian w tym wypadku). Do innych zmian może się przydać wiedza, że nasza strona contact jest osadzona w szablonie **home\_layout**. W szablonie tym możemy również dokonywać zmian. Znajdziemy go w katalogu: **/WEB-INF/layout/templates**. Przed modyfikacją należy zrobić jego kopię.
3. Rozpakowujemy archiwum **/WEB-INF/lib/dcore-webapp-components-6.0.0.jar**. Po rozpakowaniu kopiujemy plik **ContactComponent.vm** do katalogu **/WEB-INF/components/templates**
4. Od tego czasu możemy wykonać dowolne zmiany w pliku **ContactComponent.vm**. Zmiany będą miały wpływ na wyświetlaną treść na stronie "Kontakt". Po usunięciu **ContactComponent.vm** z katalogu **/WEB-INF/components/templates** nasze zmiany znikną i nastąpi powrót do domyślnego wyświetlania treści.
5. W **ContactComponent.vm** szukamy dowolnego miejsca, w którym chcielibyśmy dodać obrazek i umieszczamy interesujący fragment html np.: ``

### Dodanie nowego komponentu do podstrony

Jeśli uznamy, że chcemy wzbogacić treść naszej strony o własne dane - statyczne lub pobrane za pomocą zdalnego api, możemy utworzyć własny komponent.

Przykład: Chcemy na stronie głównej wyświetlać dane o pogodzie dla anglojęzycznych gości

1. W katalogu **/WEB-INF/components/templates** tworzymy plik **WeatherComponent.vm** z poniższą zawartością

#### WeatherComponent.vm

```
<div #if($userLanguage != "en") style="display:none" #end id="WeatherComponent"></div>
```

2. Tworzymy katalog **/style/dlibra/custom/js**, a w nim plik **WeatherComponent.js** z poniższą zawartością (skrypt pobiera dane o pogodzie z zewnętrznego serwisu yahoo i wyświetla je na naszej stronie w wybranym miejscu)

### WeatherComponent.js

```
var dlibra = dlibra || {};  
dlibra.WeatherComponent = {  
  init: function () {  
    $.get( "https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20weather.forecast%  
20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%3D%22poznan%2C%20pl%22)  
&format=json&env=store%3A%2F%2Fdatatables.org%2Falltables%2Fkeys", function( data ) {  
      var description = data.query.results.channel.item.description.replace("<![CDATA[", "").  
replace("]]>", "");  
      $('#WeatherComponent').html(description)  
    });  
  }  
}  
$(function () {  
  dlibra.WeatherComponent.init();  
})
```

3. Edytujemy plik `/WEB-INF/layout/templates/home_layout.vm` i tuż przed zamknięciem znacznika body dodajemy nasz skrypt

### home\_layout.vm

```
<script src="{homepageUrl}/style/dlibra/custom/js/WeatherComponent.js" type="text/javascript"><  
/script>  
</body>
```

4. Dodajemy komponent **WeatherComponent** na stronę główną. W tym celu w pliku `/WEB-INF/pages.xml` szukamy sekcji **main** odpowiadającej stronie głównej. Wewnątrz **components** dodajemy kolejny komponent

## pages.xml

```
<page name="main" layout="home_layout" inherits="base-page">
  <actions>
    <action name="ChangeLanguageAction"/>
    <action name="LogoutAction"/>
  </actions>
  <components>
    <component name="pl.psync.dlibra.web.comp.pages.components.
LatestEditionsComponent">
      <place>topSplit</place>
      <position>1</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.NewsComponent">
      <place>topSplit</place>
      <position>2</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.WeatherComponent">
      <place>main</place>
      <position>1</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.
CollectionsPresentationComponent">
      <place>main</place>
      <position>2</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.
PopularEditionsComponent">
      <place>main</place>
      <position>3</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.RecommendedComponent">
      <place>main</place>
      <position>4</position>
    </component>
  </components>
</page>
```

Musimy pamiętać aby pozostałym komponentom, których miejsce(place) także jest **main** nadać wyższą pozycję(**position**) - tak, żeby żaden komponent nie miał tej samej pozycji.

### 5. Na naszej stronie ukaże się pogoda w naszym mieście dla anglojęzycznych gości

**Przykładowy komponent pogody**



**Current Conditions:**  
Mostly Cloudy

**Forecast:**  
Fri - Mostly Cloudy. High: 56Low: 43  
Sat - Rain. High: 47Low: 42  
Sun - Scattered Showers. High: 41Low: 37  
Mon - Mostly Cloudy. High: 40Low: 36  
Tue - Mostly Cloudy. High: 40Low: 33

[Full Forecast at Yahoo! Weather](#)

## Usunięcie komponentu z podstrony

Może się zdarzyć, że administrator biblioteki cyfrowej zdecyduje się na usunięcie niektórych komponentów z różnych powodów. Komponent można usunąć w dwojaki sposób:

- poprzez zakomentowanie go w pliku `/WEB-INF/pages.xml` (bezpieczniej, bo zawsze możemy dany komponent przywrócić)
- poprzez usunięcie go z pliku `/WEB-INF/pages.xml`

Przykład: Decydujemy się na schowanie komponentu z obiektami najpopularniejszymi, które są wyświetlane na stronie głównej.

1. Otwieramy plik `/WEB-INF/pages.xml` i szukamy sekcji `main` zawierającej wyświetlane komponenty na stronie głównej. Okalamy komentarzem `<!--<br><component>...</component> -->` komponent `PopularEditionsComponent` jak w pliku przedstawionym poniżej.

## pages.xml

```
<page name="main" layout="home_layout" inherits="base-page">
  <actions>
    <action name="ChangeLanguageAction"/>
    <action name="LogoutAction"/>
  </actions>
  <components>
    <component name="pl.psync.dlibra.web.comp.pages.components.
LatestEditionsComponent">
      <place>topSplit</place>
      <position>1</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.NewsComponent">
      <place>topSplit</place>
      <position>2</position>
    </component>
    <component name="pl.psync.dlibra.web.comp.pages.components.
CollectionsPresentationComponent">
      <place>main</place>
      <position>1</position>
    </component>
    <!--<component name="pl.psync.dlibra.web.comp.pages.components.
PopularEditionsComponent">
      <place>main</place>
      <position>3</position>
    </component-->
    <component name="pl.psync.dlibra.web.comp.pages.components.RecommendedComponent">
      <place>main</place>
      <position>4</position>
    </component>
  </components>
</page>
```

## Ustawienie komponentu w layoutcie

Każdy komponent ma swoje ściśle określone miejsce w layoutcie strony. Informację o tym, w którym konkretnie miejscu layoutu znajduje się dany komponent możemy sprawdzić w pliku **pages.xml**. Tam możemy również wskazać inne miejsce wyświetlania się komponentu.

Przykład: Decydujemy się przenieść komponent z aktualnościami w zupełnie inne miejsce na stronie głównej.

1. Otwieramy plik **pages.xml**
2. Szukamy sekcji **main**, w której zawarte są komponenty znajdujące się na stronie głównej.
3. Odczytujemy nazwę layoutu z pola **layout**. Domyślna nazwa layoutu dla strony ma wartość **home\_layout**.
4. W sekcji **main** odszukujemy interesujący nas komponent z aktualnościami **pl.psync.dlibra.web.comp.pages.components.NewsComponent**, któremu chcemy przypisać nowe położenie w layoutcie **home\_layout**. Miejsce dla naszego komponentu z aktualnościami odczytujemy z pola **place**. Domyślnie będzie to wartość **topSplit**. Kolejność sprawdzamy w polu **position** i domyślnie będzie to wartość **2**. Pierwszym komponentem wyświetlanym w miejscu **topSplit** jest komponent z ostatnio dodanymi obiektami.
5. Modyfikujemy wartość pola **place** wewnątrz danych komponentu **pl.psync.dlibra.web.comp.pages.components.NewsComponent**. Wartość **place** zmieniamy na **myPlace**, a wartość **position** zmieniamy na **1**.

## pages.xml

```
<component name="pl.psync.dlibra.web.comp.pages.components.NewsComponent">
  <place>myPlace</place>
  <position>1</position>
</component>
```

6. Otwieramy plik **/WEB-INF/layout/templates/home\_layout.vm** i w nim umieszczamy poniższy kod w dowolnym miejscu.

### home\_layout.vm

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

Spowoduje to, że na stronie głównej wyświetlą się wszystkie komponenty, których pola **place** mają wartość **myPlace**. W tym nasz komponent z aktualnościami.

## Modyfikacja oraz dodawanie komunikatów językowych

Każdy komunikat w aplikacji czytelnika jest wyświetlany w formie wielojęzycznej. Oznacza to, że w zależności od wyboru języka strony, komunikat będzie się wyświetlać w innym języku.

Przykład: W stopce chcemy zawrzeć klauzulę: "Wszelkie prawa zastrzeżone".

1. Szukamy pliku, w którym umieszczone są informacje na samym dole strony. Analizując plik **/WEB-INF/layout/templates/home\_layout.vm** dowiemy się, że za wyświetlanie stopki jest odpowiedzialny plik **/WEB-INF/layout/templates/parts/footer.vm**.
2. W pliku **/WEB-INF/layout/templates/parts/footer.vm** dodajemy nowy wpis

### footer.vm

```
<span class="footer__bottom--text">$res.get('Home.Copyright.clause')</span>
```

Odwołanie **\$res.get('Home.Copyright.clause')** będzie wyświetlać naszą klauzulę w różnych językach. Niezbędne jest ich zdefiniowanie.

3. Otwieramy plik **/WEB-INF/layout/resources/layout\_pl.xml** i dodajemy tłumaczenie dla języka polskiego

### layout\_pl.xml

```
<entry key="Home.Copyright.clause">Wszelkie prawa zastrzezone</entry>
```

4. Otwieramy plik **/WEB-INF/layout/resources/layout\_en.xml** i dodajemy tłumaczenie dla języka angielskiego

### layout\_en.xml

```
<entry key="Home.Copyright.clause">All rights reserved</entry>
```

5. Komunikaty, które chcielibyśmy wyświetlać w komponentach zapisujemy do plików plików **/WEB-INF/components/resources/WEBAPP\_{język}.xml**. Następnie w plikach komponentów (np. **MyComponent.vm**) odwołujemy się do tłumaczeń jak poniżej:

### MyComponent.vm

```
$res.get('MyComponent.MyInfo')
```

## Podmiana istniejących obrazków (logo)

Aplikacja czytelnika oprócz obrazków dynamicznych (np. miniaturki obiektów cyfrowych) wyświetla obrazki statyczne (np. logo). Te drugie można wymieniać na swoje. Zaleca się by wymienić obrazki na obrazki o takich samych wymiarach. Jeśli nowe obrazki mają inne wymiary należy przetestować wyświetlanie takiego obrazka w różnych widokach i w różnych przeglądarkach: dla komputera stacjonarnego, tabletu i telefonu komórkowego.

Przykład: Wymieniamy domyślne logo dlibry na własne.

1. Otwieramy plik **/WEB-INF/layout/custom\_templates/custom\_layout\_library.vm** i dodajemy wpis ze ścieżką do naszego nowego logo:

```
#macro(logoPath)
    ${homepageUrl}/style/dlibra/${styleVariant}/my/logo/path/logo.svg
#end
```

Przykład: Wymieniamy domyślne logo dlibry na własne z uwzględnieniem wersji językowej logo.

1. Otwieramy plik **/WEB-INF/layout/custom\_templates/custom\_layout\_library.vm** i dodajemy wpis ze ścieżką do naszego nowego logo. Korzystając ze zmiennej `{userLanguage}` uwzględniamy logo w różnych językach:

```
#macro(logoPath)
    #if ( ${userLanguage} == "en" )
        ${homepageUrl}/style/dlibra/${styleVariant}/my/logo/path/logo_en.svg
    #elseif ( ${userLanguage} == "de" )
        ${homepageUrl}/style/dlibra/${styleVariant}/my/logo/path/logo_de.svg
    #else
        ${homepageUrl}/style/dlibra/${styleVariant}/my/logo/path/logo_pl.svg
    #end
#end
```

Przykład: Wymieniamy dowolny obrazek na własny.

1. W celu dokonania podmiany wystarczy, że namierzmy ścieżkę pod którą leży nasz stary obrazek i na jego miejscu wkleimy inny. Aby sprawdzić ścieżkę do obrazka najlepiej jest się posłużyć przeglądarką **Google Chrome**.
2. Otwieramy interesującą nas stronę w naszej bibliotece cyfrowej. Klikamy prawym przyciskiem myszy w obrazek i z menu kontekstowego wybieramy **zbadaj**. Wyświetli się nam okienko z zaznaczonym elementem w kodzie html. Będzie to element **img**. Sprawdzamy jaką wartość ma atrybut **src** tego elementu i w ten sposób odczytujemy ścieżkę do pliku na naszym serwerze. Np wartość `src="http://demo.dl.psync.pl/dlibra/default/img/pictures/my-picture.svg"` wskazuje na katalog `/webapp/style/dlibra/default/img/pictures/my-picture.svg`

## Modyfikacje css

Style, z których korzysta aplikacja czytelnika znajdują się w katalogu `/styles/dlibra/default/css/`. Istnieje możliwość nadpisania istniejących styli lub dodania nowych. Najodpowiedniejszym do tego miejscem jest plik `/styles/dlibra/default/css/custom.css`