DocTemplates2Pdf - DOKUMENTACJA

PRZYGOTOWANIE MODUŁU DO DZIAŁANIA

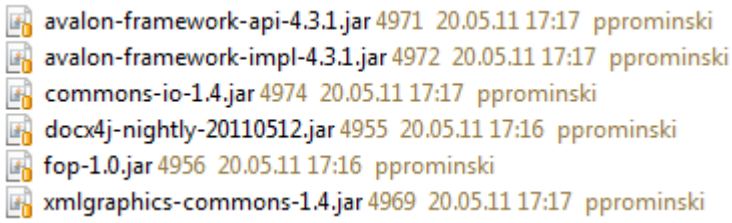
Moduł DocTemplates2Pdf znajduje się w bibliotece CUF, w ścieżce:

`com.plusmpm.CUF.util.extension.DocTemplates2Pdf`

Zadaniem modułu jest przetworzenie szablonu dokumentu w formacie *.docx, na wypełniony zmiennymi z procesu PlusWorkflow szablon *.pdf, który zostanie podłączony do wybranej klasy dokumentów w archiwum systemu. Pierwszym krokiem koniecznym przy korzystaniu z modułu jest podłączenie do projektu odpowiednich bibliotek. Niezbędne biblioteki znajdują się w projekcie CommonUsedFunctions, w ścieżce:

`trunk/WebRoot/WEB-INF/lib`

a wymagane biblioteki to:



* przy czym starsza wersja biblioteki commons-io-1.1.jar znajduje się już w systemie i dodanie wyżej wymienionej, nowszej wersji biblioteki (w wersji 1.4), bez usunięcia starsze wersji powoduje błędy podczas generowania dokumentu, co uniemożliwia prawidłowe działanie modułu

Generowanie dokumentu najłatwiej wywołać zadaniem automatycznym, w którym zostanie wykorzystana metoda

```
public static Long createPdfAndSaveInArchive(
                            String sInputFilePathOldDocx,
                            String sOutputDocClassName,
                            String sProcessId,
                            Map<String, Object> mMappings,
                            Map<String, Object> mOptionalParameters
                            )
```

gdzie:

sInputFilePathOldDocx - to ścieżka do pliku z szablonem wejściowym („*.docx”)
sOutputDocClassName - nazwa klasy dokumentów, do której zostanie podłączony dokument w archiwum systemu PlusWorkflow.
sProcessId - identyfikator procesu z którego mają zostać pobrane wartości zmiennych, które mają uzupełnić odpowiednie pola z szablonu wejściowego.

mMappings - ten parametr pozwala dynamicznie oprogramować dodawanie do szablonu wartości, które nie są pobierane bezpośrednio z procesu. Jeśli chcemy uzupełniać w szablonie jakąś zmienną wartość, która nie jest zmienną procesu, wówczas należy nadać jej jakiś znacznik (@zmienna_spoza_procesu@) i dodać do mapy dodatkowych mappingów parę: `"zmienna_spoza_procesu", "Wartość zmiennej spoza procesu"`. Ten parametr występuje w jednym z przeciążeń metody `createPdfAndSaveInArchive`, jeśli wykorzystamy przeciążenie bez tego parametru, nie możemy uzupełniać w szablonie zmiennych spoza procesu.
mOptionalParameters – mapa przechowująca dodatkowe parametry. Aby prawidłowo wywołać metodę generującą dokument konieczne jest dodanie do tej mapy pary: `"variablesMarker", "@"`, lub w miejsce znaku małpki dowolnego innego (jednoznakowego) znacznika wyróżniającego zmienne w szablonie wejściowym, które mają zostać zastąpione wartościami z procesu. Inne parametry opcjonalne to:

`"whichTables", List<Boolean>` - jeśli w dokumencie znajdują się tabele statyczne, które nie mają być przetwarzane (uzupełniane wartościami ze znadującej się w procesie tabeli dynamicznej), należy stworzyć listę obiektów Boolean rozmiarze równym ilości wszystkich tabel szablonie wejściowym, i tabele statyczne na liście powinny posiadać wartość - **`false`**, a dynamiczne, uzupełniane wartościami z procesu - wartość **`true`**. Jeśli ten parametr nie został dodany do mapy parametrów opcjonalnych, to wszystkie tabele znajdujące się w pliku (szablonie) wejściowym będą przetwarzane.

`"whichHeaders", List<Boolean>` - rozwiązanie analogiczne do poprzedniego, z tym, że tabele oznaczone wartością **`true`** na tej liście, będą posiadały nagłówki, tabele odpowiadające indeksom z listy, gdzie wartość to **`false`**, nie będą posiadały nagłówków (pierwszy wiersz to od razu będą wypełnione wartości z wnętrza tabeli dynamicznej w systemie). Ta lista też koniecznie musi mieć rozmiar równy liczbie wszystkich tabel z szablonu wejściowego, lecz możliwość usuwania nagłówka odnosi się jedynie do tabel przetwarzanych (nie do statycznych), gdyż w tabelach nieprzetwarzanych nagłówek można usunąć bezpośrednio w szablonie. Jeśli ten parametr nie został dodany do mapy parametrów opcjonalnych, to wszystkie tabele przetwarzane, znajdujące się w pliku (szablonie) wejściowym, będą posiadały nagłówki.

`"documentDescription", String` - ten parametr przechowuje wartość z postaci łańcucha znaków, który w momencie podłączania wygenerowanego dokumentu do archiwum zostanie ustawiony jako opis dokumentu.

`"documentIndices", String[]` – w tym parametrze podłączona zostaje tablica typu String, w której znajdują się wartości indeksów, które mają być przypisane do wygenerowanego pliku, podczas podłączania do archiwum.

Na wyjściu metoda zwraca identyfikator dokumentu podpiętego do archiwum. Ten identyfikator można wykorzystać, dodając go do DocIds danego procesu, wtedy wygenerowany dokument będzie podłączony danego procesu. Jest to przedstawione w przykładowej klasie wywołującej metodę poprzez zadanie automatyczne:

```
RunDocTemplates2PdfExample
```

która znajduje się folderze Dokumentacji projektu CommonUsedFunctions:

```
trunk/Dokumentacja
```

PRZYGOTOWANIE SZABLONU

Szablon dokumentu można dodać procesu, podczas konfiguracji systemu, a ścieżkę do pliku z tym szablonem pobierać w implementacji, w ten sposób można ją łatwo zmieniać (nie jest wpisana na sztywno). To też jest przedstawione w klasie z przykładowym wywołaniem.

Zmienne w szablonie która mają być dynamicznie podmieniane powinny muszą być oznaczone dodanym przy wywołaniu metody separatorem (np. @):

@id_zmiennej_1@

jeśli w mapie przekazanej do metody generującej plik znajduje się wpis „zmienna1”, „wartość”, wówczas w dokumencie wynikowym zamiast powyższego wpisu będziemy widzieli wartość. Jeśli korzystamy z przeciążenia, w którym nie przekazujemy dodatkowych mappingów, tylko identyfikator procesu, zastąpnie zmiennej ma miejsce jeśli w procesie jest zmienna o identyfikatorze „zmienna1”.

Jeśli chcemy wstawić dynamicznie dane z tabelki, wypełnionej na formularzu, musimy wtedy stworzyć w szablonie tabelę z co najmniej jednym wierszem. W tym wierszu wpisujemy w poszczególnych kolumnach identyfikatory zmiennych, analogicznie jak przy zwykłych zmiennych:

@id_zmiennej_tabelkowej1@

wtedy w kolumnie zostaną dodane kolejne wiersze z kolejnymi wartościami przechowywanymi w zmiennej procesu o id: „id_zmiennej_procesu1”. Standardowo nagłówki tabeli zostaną zamienione z identyfikatorów zmiennych między separatorami, na nazwy tych zmiennych z systemu. Jeśli nadać własne nagłówki w tabelce, inne niż te systemowe musimy wykorzystać następującą notację:

@id_zmiennej_tabelkowej1|Moja nazwa zmiennej@

Nagłówki w tabelach, jak i inne zmienne zachowują w wynikowym pliku formatowanie, zgodne z tym ustawionym w szablonie. Inne wartości uzupełniane w tabelach mają standardowe formatowanie narzucone przez program Word. Jeśli chcemy dodać jakieś inne formatowanie do wartości wewnątrz tabelki, musimy dodać drugi wiersz, poza tym nagłówkowym. Wszystkie wartości w tabelce będą sformatowane w taki sam sposób jak ten drugi wiersz. Tekst wpisany wewnątrz komórek w tym dodatkowym wierszu formatującym jest nieistotny, gdyż wszystkie wiersze poza pierwszym nagłówkowym zostaną usunięte (chyba, że parametr opcjonalny `"whichHeaders"` dla danej tabelki, będzie ustawiony na **`false`**, wtedy nagłówek też zostanie usunięty i tabelka będzie posiadał tylko wartości przepisane z tabeli na formularzu.

@ilosc@	@jednostka|J.@	@opis|Opis@	@konto_kosztowe@	@wartosc|Wartość@
T	T	T	T	T

Przykładowy szablon także znajduje się w folderze z dokumentacją

DocTemplates2Pdf.docx

przedstawione zostały w nim podstawowe funkcjonalności modułu.