

Генератор документов



Антон Горбылёв,
Разработчик 1С-Битрикс



Обзор модуля

Состав модуля



- 1 Парсер docx-файлов
- 2 Провайдеры данных для документов
- 3 Шаблоны документов - файл + набор настроек (модуль, связанные провайдеры, права доступа и пр.)
- 4 Сложные типы полей для работы модификаторов
- 5 Настройки прав доступа
- 6 Региональные настройки

Настройки шаблона

- + Активность — видимость в списке
- + С подписью и печатью. Вставлять ли по умолчанию печати и подписи в документ
- + Привязка к разделам. В этих разделах шаблон будет использоваться
- + Кто будет работать с шаблоном
- + Привязать к стране — региональные настройки
- + Используемый нумератор.

Изменить шаблон Коммерческое предложение (Великобритания)

[ОБРАТНАЯ СВЯЗЬ](#) Коммерческое пр...

19.92 КБ

[Восстановить версию шаблона по умолчанию](#)[Скачать шаблон](#)[Загрузить новый](#)

Название

 Активность С подписями и печатью

Привязать к разделам CRM

Компания ×

Контакт ×

Сделка (Общее) ×

Счет ×

Лид ×

Заказ ×

Предложение ×

Кто будет работать с шаблоном

Всем сотрудникам ×

[+ Добавить еще](#)

Привязать к стране

Великобритания ▾

[создать](#)

Используемый нумератор

Коммерческое предложение (Великобритания) ▾

[настроить](#) [создать](#)

СОХРАНИТЬ

ОТМЕНИТЬ

Региональные настройки

- + Формат даты и времени по умолчанию
- + Формат имени по умолчанию
- + Некоторые языковые фразы, которые генерируются динамически поставщиками данных (например, фразы налогов)

Добавление страны

Использовать настройки для страны

Не выбрано

Название

Формат даты

DD.MM.YYYY

Формат времени

HH:MI:SS

Формат имени

Иван Петров

Языковые фразы

НДС включен в цену

НДС не включен в цену

Налог (не НДС) включен в цену

СОХРАНИТЬ

ЗАКРЫТЬ

Настройки нумератора

- + Шаблон — по нему будет формироваться сам номер
- + Набор настроек, в зависимости от вставленных в шаблон полей
 - величина смещения
 - период работы (когда сбросится счетчик)
- + По умолчанию для каждой моей компании используется отдельный счетчик. С помощью чекбокса можно изменить это поведение - номер будет сквозной для всех моих компаний.

Подробнее про нумерацию документов будет в следующих слайдах

Редактирование нумератора

Название нумератора
Коммерческое предложение (Великобритания)

Шаблон номера
{NUMBER} ?

Последовательный номер Текущий день

Текущий месяц Текущий год Случайный номер

Префикс ID ID моей компании ID клиента

Увеличивать последовательный номер на
1

Период работы нумератора
Постоянно ▾

Использовать сквозную нумерацию на все мои компании ?

Текущий номер - 3
[Изменить текущий номер](#)

СОХРАНИТЬ

ОТМЕНИТЬ

Настройки прав доступа

- + Настройки — права доступа, региональные настройки;
- + Шаблоны - изменение шаблонов. Есть разные уровни доступа (свои, свои и отдела, все).
- + Документы - изменение даёт возможность изменять документы (с учетом привязки используемого шаблон к пользователю)
- + Документы - просмотр даёт возможность просматривать документы, созданные по тем шаблонам, к которым у пользователя есть доступ.

Настройки прав доступа

		Роль		Список ролей:	
Группа	Администраторы	Администратор	×	Администратор	✎ ✕
Отдел	Моя компания: Все сотрудники отдела с подотделами	Менеджер	×	Менеджер	✎ ✕
Добавить право доступа					
Добавить					

Настроить права доступа сотрудников к документам CRM можно в коммерческих тарифах.

Редактирование роли Менеджер

Роль

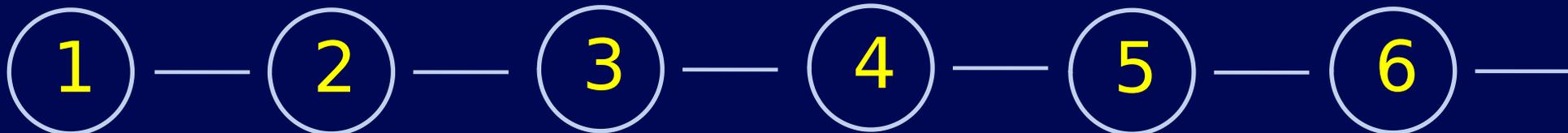
Менеджер

Сущность	Действие	Право
Настройки	Изменение	Есть доступ
Шаблоны	Изменение	Любые
Документы	Изменение	Есть доступ
	Просмотр	Есть доступ

Конвертация в PDF



Цикл генерации документа



1
Файл шаблона
скачивается и
распаковывается

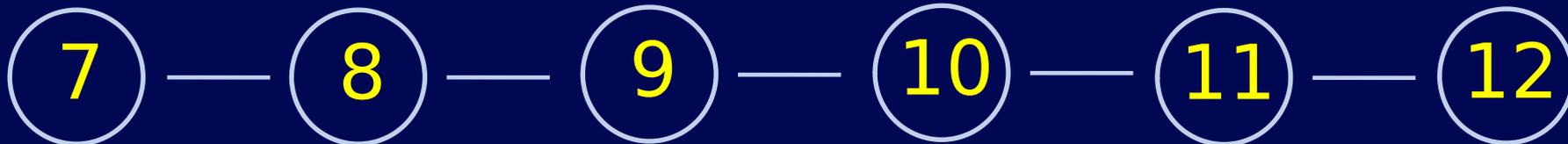
2
Из файла
извлекается
список полей

3
Сопоставление
полей и
способов
получения их
значений

4
Получение
данных из
провайдера

5
Парсер
вставляет
текстовые
значения

6
Изображения
скачиваются и
вставляются в
файл



7
Файл
документа
закачивается
в облако и
сохраняется

8
**В этот момент
можно
работать с
docx файлом**

9
Добавление
команды на
конвертацию
в очередь на
сервере

10
После
ожидания в
очереди файл
скачивается
на сервер

11
Производится
конвертация
в PDF с
помощью
LibreOffice

12
Сервер
закачивает PDF
на портал —
документ
сконвертирован

Особенности работы конвертера файлов



- ① Асинхронная операция - конвертация производится на другом сервере
- ② Очередь на выполнение - файл не конвертируется сразу же
- ③ Непосредственно конвертацию выполняет пакет LibreOffice версии 6.2

За отображение PDF отвечает LibreOffice



 RabbitMQ by Pivotal.

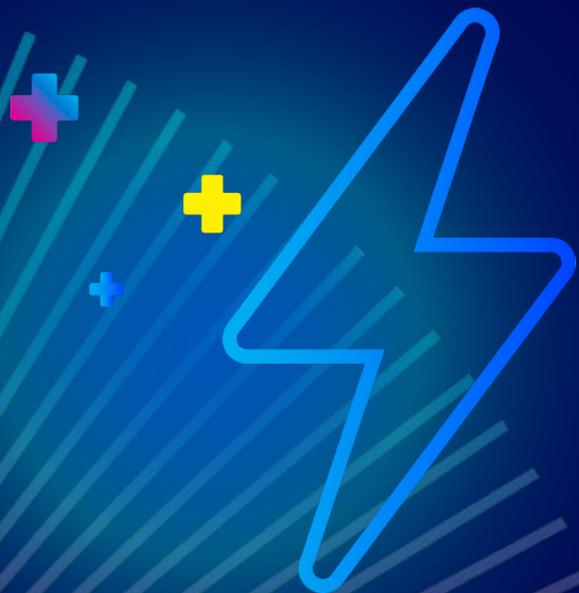


LibreOffice
The Document Foundation



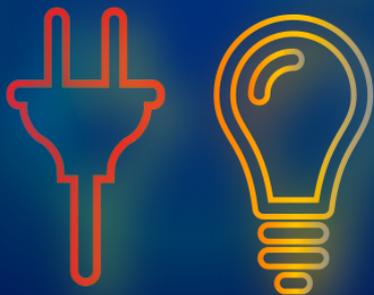
Как добиться лучшего соответствия DOCX и PDF

- ⚡ Исходный документ — уже существующий экземпляр, созданный в Microsoft Word
- ⚡ Файл открывается в LibreOffice. Проверяется соответствие внешнего вида в двух редакторах. Если есть расхождения - они устраняются
- ⚡ Не используйте векторные изображения. Чем проще разметка — тем лучше
- ⚡ В результате должен получиться файл, который выглядит одинаково в Microsoft Word и LibreOffice
- ⚡ В файл вставляются {поля}. После этого можно ещё раз посмотреть в двух редакторах.
- ⚡ Шаблон загружается на портал. Обязательно сделать тестовый документ



Типы полей и модификаторы

Какие поля вставлять в шаблон



- ⦿ Интерфейс поиска полей неудобный. Меньше полей — проще найти ;-)
- ⦿ Не все поля из списка работают :-)
- ⦿ Чем короче название поля, тем лучше
- ⦿ “Умные” поля для реквизитов. {Requisite...} - реквизиты клиента. {BankDetail...} - банковские реквизиты клиента
- ⦿ Контакт и компания - собственные реквизиты;
- ⦿ Лид, сделка, заказ - это реквизиты компании, если она привязана;
- ⦿ Если не привязана - реквизиты основного (первого) контакта;
- ⦿ В счете и предложении - выбранные реквизиты контрагента.

Модификаторы

- + Некоторые поля по своей природе являются сложными.
- + Необходимо управлять представлением таких полей на уровне шаблона
- + Для этого используются модификаторы.
- + Вставляются внутрь поля после знака тильда (~)
- + Для каждого типа поля есть свои модификаторы.
- + Модификаторов может быть несколько. Пишутся в формате название=значение, разделяются запятой
- + Например, {Address~format=3, separator=3}
- + Названия параметров внутри модификатора регистронезависимы

Типы полей

- + Дата/время
- + Номер телефона
- + Имя
- + Изображение
- + Печать / подпись
- + Текст / html
- + Деньги
- + Адрес
- + Список
- + Множественные значения



Дата / время



- ◉ По умолчанию используется формат из региональных настроек шаблона
- ◉ Широкий выбор вариантов отображения



```
{DocumentCreateTime~format=d.m.Y H:i:s}
```

```
04.10.2019 17:00:00
```

Номер телефона



⦿ Доступно три формата:

E.164	+79002191234
International	+7 900 219-12-34
National	8 (900) 219-12-34

⦿ По умолчанию используется формат E.164

```
{AssignedMobilePhone~format=International}
```

```
+7 555 111-66-44
```

Имя

BITRIX
POWER

- Любые поля типа «Имя» содержат в себе всё имя целиком, но для некоторых полей предустановлены модификаторы
- К примеру, {ContactName} - это то же самое, что {ContactLastName~format=#NAME#}
- В шаблонах можно использовать любую часть имени;
- Склонение имён работает только для русского языка и если заполнено отчество.
- Для пользователей пол может быть указан в явном виде
- Для склонения имён используется open-source библиотека Petrovich

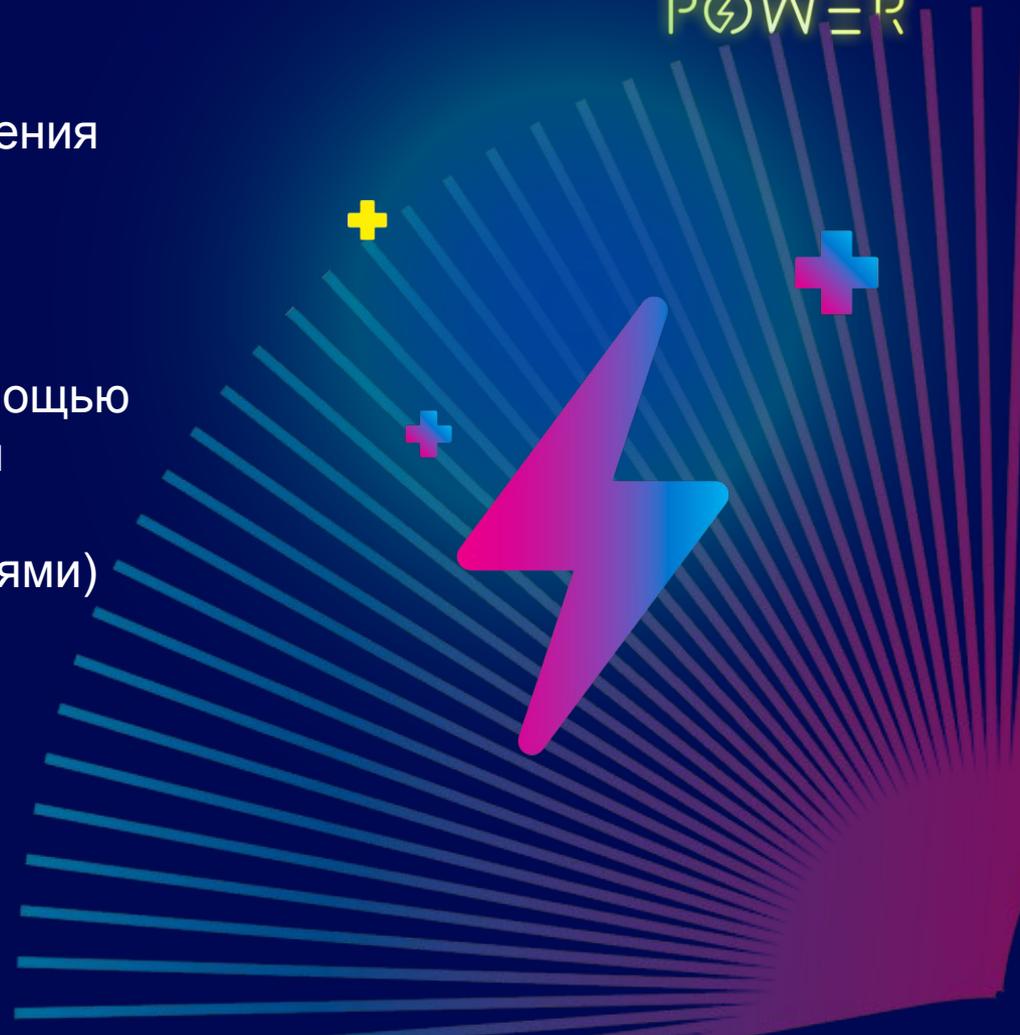


для {AssignedLastName~format=#NAME#,case=0}
для Антона



Изображение

- + Все поля типа “Файл” считаются — изображения
- + Вставка производится через изображение-заглушку
- + Если надо вставить одно поле дважды, то в параметрах надо указать разные поля с помощью модификаторов (из-за особенностей работы LibreOffice - он не допускает наличие двух и более изображений с одинаковыми названиями)
- + Не используйте векторные форматы



Печать / подпись

- + Как изображение, но вставляется в документ только при установленном чекбоксе “С печатью и подписью”
- + Лучше использовать .png-файл с прозрачным фоном
- + Используйте LibreOffice



Текст / html



- ⦿ Переносы строк сохраняются
- ⦿ Если в значении поля есть хотя бы один тег, то он считается html
- ⦿ При вставке html сохраняются переносы строк через тег `
` и переносы блочных элементов
- ⦿ Сохраняются свойства шрифта - **жирный**, *курсив*, подчеркнутый, ~~зачеркнутый~~
- ⦿ Нумерованные и ненумерованные списки вставляются как списки
- ⦿ **Пока что нет поддержки уровней вложенности списков**
- ⦿ **Всё остальное не работает: таблицы, отступы, изображения и пр.**

Деньги



- ⦿ Это поле доступно только при работе с CRM
- ⦿ По умолчанию используется формат валюты из настроек CRM, но без знака валюты
- ⦿ Для итоговых сумм по умолчанию выводятся незначащие нули и знак валюты
- ⦿ Можно вывести сумму прописью для рублей, белорусских рублей, гривен и тенге.
- ⦿ Язык суммы прописью зависит только от валюты

```
{TotalSum~wz=y, ns=y}  
155 000.00
```

Адрес

- + Доступно только при работе с CRM
- + По умолчанию берется формат адреса из настроек CRM

Адрес

По умолчанию, Битрикс24 использует формат адреса страны текущего шаблона.

Мы можем изменить тип формата даты и разделитель строк. Зададим тип адреса Северной Америки с переводом строки между частями адреса.

```
{Address-Format=3, Separator=3}
```

Форматы адресов и разделители

Часть модификатора **Format** отвечает за тип формата:

- 1 - Европа;
- 2 - Великобритания;
- 3 - Северная Америка;
- 4 - Россия (улица -> страна);
- 5 - Россия (страна -> улица).

Часть **Separator** отвечает за разделитель строк:

- 1 - запятая;
- 2 - без разделителя;
- 3 - перевод строки.

Выбираемые значения

- + Некоторые поля могут принимать несколько значений. Например, у вас может быть несколько моих компаний.
- + У каждой компании может быть несколько наборов реквизитов
- + При редактировании документа можно будет выбрать конкретное значение
- + Все поля будут заменены автоматически, если они не были введены вручную

Реквизиты клиента

Реквизиты клиента

ИП Иванов Петр Сергеевич

Название

ИП Иванов Петр Сергеевич

Сокращенное наименование организации

ИНН

2354234

КПП

СПИСКИ

- + В качестве значения поля выступает набор (список, массив) значений другого провайдера.
- + Товары, налоги, контакты лида/сделки, компании контакта.
- + Множественные значения полей типа “Привязка к CRM”
- + Такие поля можно вставить в документ несколькими способами:
 - + Таблицы
 - + Повторяющиеся блоки
 - + Если вставить поле в произвольное место документа, то будет выведено значение первого элемента списка
 - + С помощью модификатора index можно указать порядковый номер элемента списка, из которого должно быть вставлено значение поля
 - + С помощью модификатора all можно вставить значение одного поля всех элементов списка. В этом случае эти значения будут вставлены как множественное значение одного поля.
- + Сейчас списки не работают, если они не находятся на первом уровне вложенности.

Множественные значения

- + Некоторые поля содержат в себе множественные значения
- + Например, телефон / почта контакта. Для сохранения обратной совместимости для них по умолчанию установлен модификатор `mfirst=y`
- + Поддерживаются множественные значения пользовательских полей
- + Можно выводить множественные значения из списков
- + Для таких полей есть несколько модификаторов
 - + `mfirst=y/n`. По умолчанию `n`. Выводить все значения или только первое
 - + `mseparator=1/2`. Разделитель множественных значений. По умолчанию 1 - запятая, 2 - перевод строки
- + Множественные значения типа файл не поддерживаются, в них всегда берется первое значение.

Множественные значения списков

- + Модификаторы полей можно комбинировать.
- + Например, список контактов сделки. Контакт содержит в себе поле “телефон”. Это поле множественное, и его тип тоже имеет свои модификаторы
 - + Вывести все номера телефонов второго контакта через запятую в международном формате `{ContactsContactPhone~index=1,mseparator=1,format=International}`
 - + Вывести все номера телефонов всех контактов, каждый номер на новой строке в формате E.164
`{ContactsContactPhone~mseparator=2,all=y,mseparator=2,format=E.164}`
 - + Вывести все номера телефонов всех контактов, каждый контакт с новой строки, а его номера - через запятую
`{ContactsContactPhone~mseparator=2,all=y,mseparator=1}`
 - + Первый модификатор влияет на разделение в рамках элементов списка, а второй - на разделение значений элемента списка

Вставка списков в таблицы

BITRIX
POWER

- ⦿ Если в строку таблицы вставлено хотя бы одно поле, которое является частью списка, то эта строка будет размножена столько раз, сколько есть элементов в списке.
- ⦿ Этот режим работает только для простых таблиц.
- ⦿ Не надо вставлять в строку таблицы поля двух списков
- ⦿ В таблицу можно вставить изображение
- ⦿ Если список пустой (нет ни одного значения) и в параметрах поля указано `HIDE_ROW=Y`, то строка этой таблицы будет вырезана из документа (например, налоги в стандартных шаблонах)



Вставка списков с помощью повторяющихся блоков

- + Таблица может не устроить — можно использовать блоки
- + В начале области ставится специальное поле вида {#ПОЛЕ#.BLOCK_START}, а в конце {#ПОЛЕ#.BLOCK_END}. Где #ПОЛЕ# - это название поля-списка в верхнем регистре. Для контактов это будет CONTACTS, для налогов TAXES, для товаров PRODUCTS, для пользовательских полей код поля UF_CRM_D34M2L331
- + Маркеры начала и конца будут стерты
- + В повторяющихся блоках можно использовать поля-изображения
- + Если список пустой, то повторяющаяся область будет вырезана из документа.

Что будет, если таблицу
поместить внутрь
повторяющихся блоков?



Таблицу можно скопировать при
использовании PHP / Rest API, про это
будет в следующих слайдах

Управление значениями полей



Способ сопоставления имен полей и способ получения их значений

- + Любое поле может быть другим провайдером, в том числе список
- + Модуль строит карту таких полей, где путь для получения значения - это цепочка ключевых слов, разделенных точкой (например, “Имя контакта” - это `this.SOURCE.CONTACT.NAME`)
- + Для получения доступа к элементам списков добавляется “Промежуточное” ключевое слово. Для списков из пользовательских полей это `ITEM` (например, “Название товара” \leftrightarrow `this.SOURCE.PRODUCTS.PRODUCT.NAME` \leftrightarrow `{ProductsProductName}`)
- + Для лучшей читаемости этот путь преобразовывается в простую строку (`this.SOURCE.CONTACT.NAME` \leftrightarrow `{ContactName}`)
- + Полностью алгоритм поиска значений полей выглядит так:
 - + из файла шаблона извлекаются названия полей;
 - + производится поиск по провайдеру и его полям, чтобы найти соответствие путей и читаемых названий;
 - + когда путь найден, модуль вычисляет значение поля для этого пути.

Переопределение значений полей

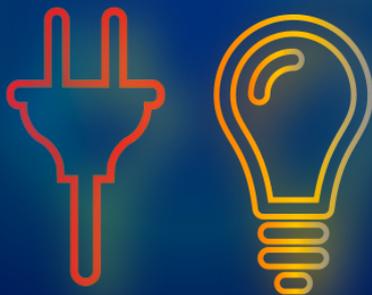
- + Значение каждого поля может быть переопределено.
- + Кроме списков и множественных значений. Сейчас API модуля не поддерживает возможность изменения значений таких полей;
- + В шаблон документа можно вставить произвольные поля, а значения для них передать самостоятельно. Через REST можно передать и тип поля;
- + В качестве значения поля можно передать ссылку на любое поле, доступное в шаблоне;
- + Передать значения полей можно несколькими способами:
 - + Через интерфейс (“редактировать документ”)
 - + Робот / бизнес-процесс
 - + PHP API
 - + REST API
- + Сначала рассмотрим несколько служебных полей документа

{DocumentCreateTime}

BITRIX
POWER



© Содержит дату и время создания документа



{DocumentTitle}

- ⦿ Это поле всегда есть в документе
- ⦿ По умолчанию формируется как “#Название шаблона#
#Номер#”
- ⦿ Название файла документа при скачивании формируется как {DocumentTitle}.docx
- ⦿ В текст шаблона можно вставить другое поле, а это использовать только для формирования имени файла при скачивании



Например, ваш шаблон называется “Клиентская документация (версия 4)”. В тексте вы хотите иметь заголовок вида “Документация от #Текущая дата#”. А файл должен называться “Документация для клиентов.docx”. Для этого в текстовый заголовок вместо #Текущая дата# вставляете поле {DocumentCreateTime}, а в поле {DocumentTitle} вставляете текст “Документация для клиентов”

{DocumentNumber}



- ⦿ Это поле всегда есть в документе
- ⦿ По умолчанию значение получает от нумератора
- ⦿ Если переопределить это поле, то документ не обращается к нумератору для получения следующего номера. Соответственно, состояние нумератора не меняется

Например, вы не хотите использовать нумератор для документов определенного типа клиентов. В этом случае перед созданием документа вы проверяете тип клиента. Если для него не нужен нумератор, то вы передаете в это поле сформированный снаружи номер. А иначе передаете туда “системное” значение этого поля - “this.DOCUMENT.NUMBER” (в этом случае будет использован нумератор)

Сквозная нумерация при синхронизации из нескольких источников



Всегда использовать нумератор шаблона. Перед созданием нового документа во внешней системе производится rest-запрос на создание нового документа в битрикс24, получается номер этого документа, удаляется документ, номер используется во внешней системе.



Не использовать нумератор шаблона. В этом случае должен быть внешний синхронизируемый нумератор, а номер документа при создании в битрикс24 всегда передается вручную.

Автоматизация



Робот «Создание документа CRM»

- + Сейчас работает только в рамках CRM
- + В настройках можно выбрать нужный шаблон
- + Можно указать значение любого поля
- + В результатах доступны:
 - + Идентификатор созданного документа (можно использовать в REST-запросах)
 - + Номер документа
 - + Идентификатор docx-файла (можно скопировать на диск)
 - + Публичная ссылка на документ
- + Если в настройках стоит чекбокс “Ожидать окончания конвертации”, то полностью робот будет выполнен только после окончания конвертации, и к результатам добавится идентификатор pdf-файла (можно скопировать на диск)
- + Если по какой-то причине результат конвертации не был получен, то робот “зависнет”
- + При конвертации может произойти ошибка, и в этом случае pdf файла не будет - надо это учитывать

Порядок вычисления значений полей в бизнес-процессах



Для всех полей принцип одинаковый:

- 1 Сначала работают вычисления бизнес-процессов
- 2 Вычисленные значения передаются в генератор документов
- 3 Генератор документов обрабатывает полученную строку
- 4 Если она является ссылкой на какое-то поле, то значение будет вычислено по переданному пути
- 5 Если это не путь - строка будет вставлена как есть

Подстановка значения по умолчанию с помощью робота

У вас есть поле в шаблоне, куда вы хотите подставлять значение по умолчанию, если оно не заполнено. Например, это строковое пользовательское поле.

- + С помощью функций бизнес-процессов проверяете, что поле заполнено
- + Если оно заполнено - передаете в значение поля его "путь"
- + Если поле не заполнено - вставляете значение (например, "Неизвестно")
- + В бизнес-процессах можно вставить значение глобальной константы

```
{{= if ({{Строковый параметр}},  
'this.SOURCE.UF_CRM_1528115104210',  
'Строковый параметр не заполнен') }}
```

Документ, если поле не заполнено

Выполнить: [сразу](#) после предыдущего робота

Условие: [\[+\]](#)

Шаблон: Демонстрация_робот

Ожидать конвертации PDF файла

Поля документа

Добавить поле: [Добавить](#)

UfCrm1528115104210 [Удалить](#)

Документ -> Строковый параметр

`{{= if ({{Строковый параметр}}, 'this.SOURCE.UF_CRM_1528115104210', 'Строковый параметр не заполнен') }}`

[СОХРАНИТЬ](#) [ОТМЕНИТЬ](#)

Триггеры документов



- Генерация документа
Срабатывает при создании нового документа по выбранному шаблону



- Просмотр документа
Срабатывает при просмотре публичной страницы документа по выбранному шаблону
Триггер срабатывает при просмотре любым пользователем



REST API

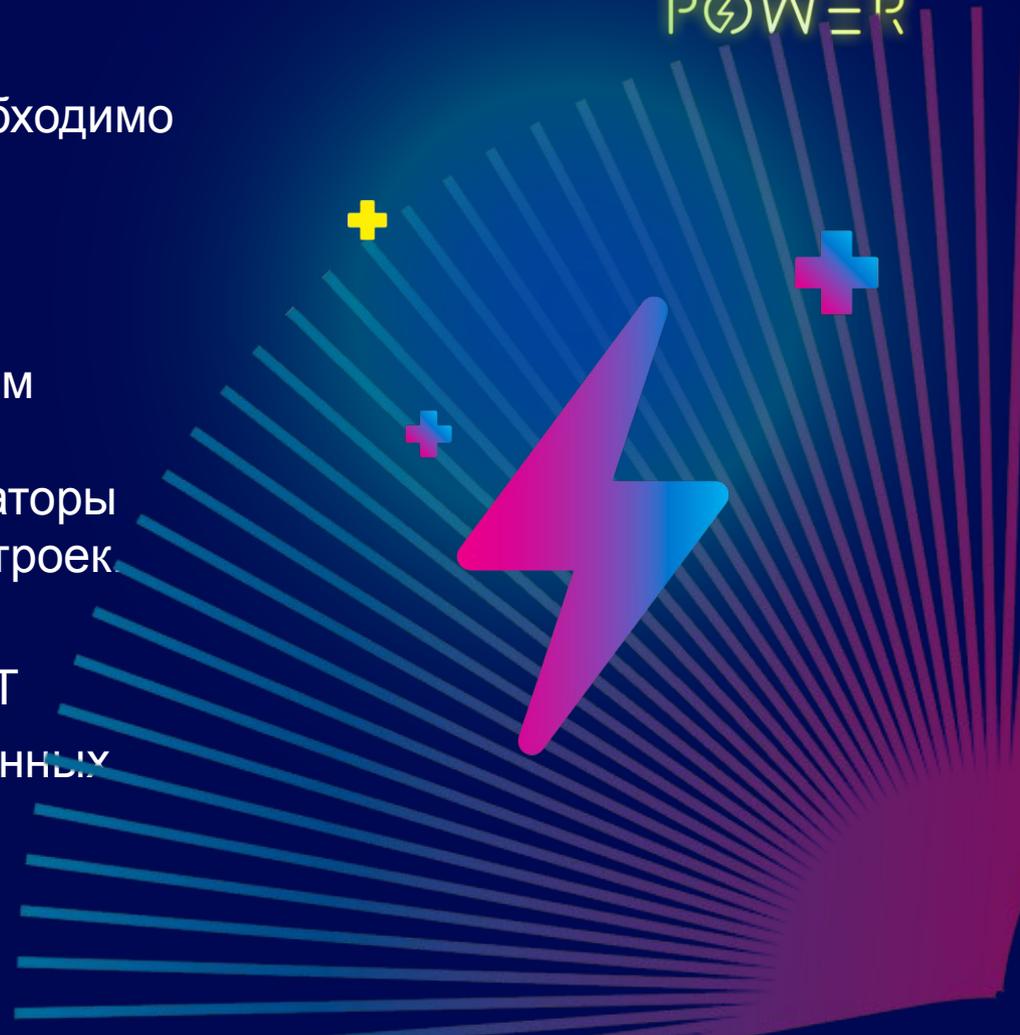
Особенности REST генератора документов

- + REST API модуля охватывает весь доступный функционал
- + Документация по REST модуля достаточно подробна
- + На данный момент есть два scope для работы с генератором документов
 - + Методы `crm.documentgenerator.*`. Результаты работы этих методов отображаются в интерфейсе CRM.
 - + Методы `documentgenerator.*`. Результат работы этих методов доступен только на уровне REST.
- + Из методов одного scope нельзя получить доступ к данным другого
 - + Нельзя создать документ CRM с шаблоном для REST
 - + Нельзя использовать данные CRM при работе с методами `documentgenerator.*`
- + Дальше все примеры будут рассматриваться на методах `documentgenerator.*`

Связанные сущности

Перед началом создания документов необходимо создать ряд дополнительных сущностей

- + Нумератор
 - + Подробной документации по настройкам нумераторов нет. Если есть необходимость управления - можно создать разные нумераторы через интерфейс и посмотреть наборы настроек.
 - + Обновить / удалить можно только тот нумератор, который был создан через REST
- + Регион можно взять один из предустановленных



Загрузка шаблона документа

documentgenerator.template.add

BITRIX
POWER



- ⦿ Контент файла (параметр file) можно передать двумя способами:
 - ⦿ В post-запросе закодированным в base64 (fields[file])
 - ⦿ Без кодировки в multipart/form-data (просто file)
- ⦿ Т.к. интерфейс должен быть реализован самостоятельно, то настройки видимости (параметр fields[users]) нужны только самому приложению.
- ⦿ Аналогично индекс сортировки (параметр fields[sort]) и активность (параметр fields[active])
- ⦿ Все созданные через этот метод шаблоны привязаны к модулю rest и единственному провайдеру `\Bitrix\DocumentGenerator\DataProvider\Rest`.
- ⦿ Можно передать параметр fields[code] - символьный код шаблона.

Просмотр полей шаблона

documentgenerator.template.getfields



- ⦿ На вход получает идентификатор
 - ⦿ шаблона
- В результате отдается набор полей шаблона

Удаление шаблона

documentgenerator.template.delete



- ① Если есть хотя бы один документ, созданный по этому шаблону, то он не удаляется на самом деле. Запись остается для сохранения привязки документа к модулю
- ② Если нет ни одного документа, то запись о шаблоне удаляется полностью
- ③ При получении списка шаблонов их можно отфильтровать по параметру `isDeleted=Y`

Простое создание документа

documentgenerator.document.add

- ⦿ Параметр `value` нужен только для интерфейса приложения, как идентификатор внешнего источника. Это строковый параметр, сюда можно передать полноценный внешний код. Например, `"PARTNER_APP_10_BILL_133"`
- ⦿ `values` - это набор значений полей документа. В простом случае это одномерный массив, где ключ - название поля, а значение - строка для вставки в документ.
- ⦿ `fields` - это описание полей документа.



Описание простых полей при создании документа

Параметр `fields` — массив, где ключ — название поля, а значение - описание

- + `TYPE` — тип поля
- + `FORMAT` — формат
- + `PROVIDER` — провайдер
- + `TITLE` — заголовок

Коды типов полей

- + `IMAGE` - изображение
- + `STAMP` - печать / подпись
- + `DATE` - дата / время
- + `NAME` - имя
- + `PHONE` - номер телефона

Передача значений полей при создании документа

В массиве `values` ключ — это имя поля, а значение, как правило, строка. Для сложных типов полей надо формировать значения по-другому

- + Значение поля типа “Изображение” — ссылка на файл
- + Значение поля типа “Имя” — массив, где `GENDER` — это пол (M/F)

```
$values['SpeakerPhoto'] = 'https://mysite.site/anton.png';  
$values['SpeakerName'] = [  
    'NAME' => 'Антон',  
    'LAST_NAME' => 'Горбылёв',  
    'SECOND_NAME' => 'Анатольевич',  
    'GENDER' => 'M',  
];
```

Значения списков

○ Список - это одно поле, которое является провайдером

○ `ArrayDataProvider`

Этот провайдер содержит в себе список других

○ провайдеров (в ресте доступен только базовый

`HashDataProvider`

Пример описания и значения поля

```
$fields = [  
    'Speakers' => [  
        'PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\  
ArrayDataProvider',  
        'OPTIONS' => [  
            'ITEM_NAME' => 'Speaker',  
            'ITEM_PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\  
HashDataProvider',  
        ],  
    ],  
    'SpeakersSpeakerPhoto' => [  
        'TYPE' => 'IMAGE',  
    ],  
];  
$values = [  
    'Speakers' => [ [  
        'Name' => 'Антон Горбылев',  
        'Position' => 'Разработчик',  
        'Company' => '1С-Битрикс',  
        'Photo' => 'https://my.site/anton.png',  
    ],  
    /*[], следующие элементы списка /  
],  
    'SpeakersSpeakerName' => 'Speakers.Speaker.Name',  
    /* "" => "" остальные поля аналогично  
];
```

Передача списков

- + `fields[Speakers][PROVIDER] = Bitrix\DocumentGenerator\DataProvider\ArrayDataProvider`
- + `fields[Speakers][OPTIONS][ITEM_NAME] = Speaker` - Код элемента списка
- + `fields[Speakers][OPTIONS][ITEM_PROVIDER] = Bitrix\DocumentGenerator\DataProvider\HashDataProvider`
- + `values[Speakers]` - простой массив, где каждое значение - это набор значений для одной строки таблицы.
- + Кроме непосредственно значений списка необходимо передать пути к этим значениям в явном виде. `values[SpeakersSpeakerName] = Speakers.Speaker.Name` и т.д.

Если какое-то из полей элемента списка является сложным, то передаем его описание отдельно

```
fields[SpeakersSpeakerPhoto][TYPE] = IMAGE
```

Названия полей списков можно использовать любые, но лучше придерживаться следующего:

Имя поля элемента списка снаружи = `#Название поля-списка###Код элемента списка###Название поля списка#`

Значение для этого поля = `#Название поля-списка#.#Код элемента списка#.#Название поля списка#`

Пример содержимого шаблона

1 Повторяющиеся блоки

```
{Speakers.BLOCK_START}  
Speaker: {SpeakerSpeakerName}  
{SpeakersSpeakerPosition},  
{SpeakersSpeakerCompany}  
{Speakers.BLOCK_END}
```

2 Таблица

Name	Position
{SpeakersSpeakerName}	{SpeakersSpeakerPosition}, {SpeakersSpeakerCompany}

Копирование таблиц



- ⦿ С помощью REST API можно организовать копирование таблицы.
- ⦿ Для этого её нужно поместить внутрь повторяющихся блоков.
- ⦿ “Внешний” список должен в качестве значений отдавать названия полей для внутренних списков. Все описания и значения полей должны быть переданы сразу.
- ⦿ В массиве описаний полей внешний список должен идти первым (т.к. поля обрабатываются в том же порядке, в котором они переданы в описании).

Значения при копировании таблиц

```
$values = [ /* значения простых полей*/
  'Events' => [
    [
      'Title' => 'Автоматизация',
      'SpeakerName' => '{Event1SpeakersSpeakerName}',
      /* аналогичные значения полей - вставляются поля другого списка */
    ], /* [] аналогичные значения второго элемента, только с {Event2SpeakersSpeakerName}*/
  ],
  'EventsEventTitle' => 'Events.Event.Title',
  /* остальные пути к полям Events */
  'Event1Speakers' => [
    [
      'Name' => 'Антон Янжула', 'Company' => '1С-Битрикс', 'Position' => 'ведущий разработчик',
    ], /* [] следующие элементы списка */
  ],
  'Event1SpeakersSpeakerName' => 'Event1Speakers.Speaker.Name',
  /* остальные пути к полям Event1Speakers */
  'Event2Speakers' => [
    [
      'Name' => 'Антон Горбылёв', 'Company' => '1С-Битрикс', 'Photo' => 'http://my.site/anton_gorbylev.png'
    ], /* [] следующие элементы списка */
  ],
  'Event2SpeakersSpeakerName' => 'Event2Speakers.Speaker.Name',
  /* остальные пути к полям Event2Speakers */
];
```

Описания полей при копировании таблиц

```
$fields = [  
  'Events' => [  
    'PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\ArrayDataProvider',  
    'OPTIONS' => [  
      'ITEM_NAME' => 'Event',  
      'ITEM_PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\HashDataProvider',  
    ],  
  ],  
  'Event1Speakers' => [  
    'PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\ArrayDataProvider',  
    'OPTIONS' => [  
      'ITEM_NAME' => 'Speaker',  
      'ITEM_PROVIDER' => 'Bitrix\\DocumentGenerator\\DataProvider\\HashDataProvider',  
    ],  
  ],  
  'Event2Speakers' => [  
    /* структура аналогичная Event1Speakers */  
  ],  
  'Event1SpeakersSpeakerName' => ['TITLE' => 'Event1SpeakersSpeakerName'],  
  'Event1SpeakersSpeakerPhoto' => ['TITLE' => 'Event1SpeakersSpeakerPhoto', 'TYPE' => 'IMAGE'],  
  /* "" => "", остальные поля первого события аналогично */  
  'Event2SpeakersSpeakerName' => ['TITLE' => 'Event2SpeakersSpeakerName'],  
  /* "" => "", остальные поля второго события аналогично */  
];
```

Пример содержимого шаблона для копирования таблиц

{Title}

{Description}

{EVENTS.BLOCK_START}

Event {EventsEventTitle}

{EventsEventDescription}

Name	Position
{SpeakersSpeakerName}	{SpeakersSpeakerPosition}, {SpeakersSpeakerCompany}

{EVENTS.BLOCK_END}

Почему в результате `document.add` нет ссылки на pdf?



Как говорилось выше, конвертация в pdf - операция асинхронная. На момент окончания генерации документа ещё нет pdf-файла



Если для документа очень нужен pdf, то сейчас единственный вариант - сделать повторный запрос `documentgenerator.document.get` через 20-30 секунд, чтобы считать ссылку на pdf. Если она там не появилась - попробовать повторить.

Отличие параметров методов разных scope

- ⦿ “Изнутри” методы идентичны. По факту методы `crm.documentgenerator.*` после пред-обработки параметров вызывают методы `documentgenerator.*`. Но есть ряд отличий
- ⦿ На вход методов `crm.documentgenerator.*` необходимо вместо имен провайдеров передавать ид типа сущности CRM (параметр `entityTypeId`)
- ⦿ На вход методов `crm.documentgenerator.*` необходимо вместо параметра `value` передавать параметр `entityId` - ид сущности CRM



Итого: что сделать можно

-  Создание документов на основе шаблонов в формате .docx файлов
-  В шаблон можно вставить списки с произвольным количеством элементов через таблицы или повторяющиеся блоки
-  В шаблон можно вставить изображения, в том числе из списков
-  Вставить поля в виде html с частичным сохранением форматирования
-  Создавать документы, отправлять их и отслеживать просмотр без участия пользователя (через роботы)

Итого: что сделать нельзя

-  Вставить множественное значение поля типа «файл»
-  Вставка таблиц и изображения из html
-  Вставка векторных изображений
-  Не полностью передается форматирование

Планы развития

Наследование

Убрать final и
сделать
возможность
использования
своих наследников

Поиск полей

Новый интерфейс

Управление значениями списков

API и интерфейс

Редактор

Редактирование
файла документа
онлайн

Бизнес-процесс на произвольных данных

Топ запросов



- 1 Реестр документов
- 2 Сохранение форматирования из Microsoft Word
- 3 Формулы в шаблонах
- 4 Локальный конвертер файлов в коробке

Спасибо всем!

АНТОН ГОРБЫЛЁВ
1С-Битрикс

@keymnsk

