

ОфисГен

Генерация Open XML документов по XML-схемам

ООО «Интеграл»

26 мая 2026 г.

Содержание

1	Введение	2
2	Структура ZIP-пакета	2
2.1	Файл манифеста	2
3	Создание схемы данных (schema.xml)	2
3.1	Пример схемы	2
3.2	Атрибуты привязки	3
3.3	Поддерживаемые типы колонок	3
4	Настройка шаблона документа	4
4.1	Для DOCX (Microsoft Word / LibreOffice Writer)	4
4.2	Для XLSX (Microsoft Excel / LibreOffice Calc)	4
5	Штрихкоды (Barcode)	4
5.1	Доступные форматы штрихкодов	5
5.2	Дополнительные атрибуты штрихкодов	5
6	Репликация данных (повтор блоков)	5
7	Пример данных (data.xml)	6
8	Форматы вывода	6
9	Обработка ошибок	6
10	Требования к среде выполнения	7
11	Пример использования (клиент)	7

1 Введение

Система «ОфисГен» предназначена для автоматической генерации офисных документов в форматах Open XML (например, `.docx`, `.xlsx`) на основе:

- Шаблона документа;
- XML-схемы с атрибутами привязки;
- XML-данных, соответствующих схеме.

Все компоненты упаковываются в ZIP-архив, который отправляется в систему. На выходе формируется готовый документ с заполненными полями, повторяющимися блоками и штрихкодами.

2 Структура ZIP-пакета

Архив должен содержать как минимум три файла, а также файл манифеста `manifest.json`, описывающий их расположение:

```
.zip
manifest.json
template.docx ( template.xlsx)
schema.xml
data.xml
```

2.1 Файл манифеста

Пример `manifest.json`:

```
{
  "template": "./template.docx",
  "schema": "./schema.xml",
  "data": "./data.xml"
}
```

- Ключи `template`, `schema`, `data` — обязательны.
- Пути указываются относительно корня архива, допускается префикс `./` или `/`.

3 Создание схемы данных (`schema.xml`)

Схема представляет собой обычный XSD-файл, дополненный атрибутами из **вашего пространства имён** (далее используется префикс `tpl` для краткости).

3.1 Пример схемы

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:tpl="http://mycompany.com/template">

  <xs:element name="Report" type="tpl:ReportType"/>
```

```

<xs:complexType name="ReportType">
  <xs:sequence>
    <xs:element name="Title" type="xs:string" tpl:binding="DocTitle"/>
    <xs:element name="Date" type="xs:date" tpl:binding="ReportDate" tpl:format="
      dd.MM.yyyy"/>
    <xs:element name="Rows" type="tpl:RowType" maxOccurs="unbounded" tpl:binding=
      "TableRow"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RowType">
  <xs:sequence>
    <xs:element name="Code" type="xs:string" tpl:binding="CodeCell"/>
    <xs:element name="Amount" type="xs:decimal" tpl:binding="AmountCell"
      tpl:format="N2"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

3.2 Атрибуты привязки

Все атрибуты должны быть объявлены в том же пространстве имён, что и префикс `tpl`.

Атрибут	Обязательность	Описание
<code>tpl:binding</code>	Да	Идентификатор элемента в шаблоне документа.
<code>tpl:type</code>	Для BARCODE	Тип колонки (см. раздел 3.3).
<code>tpl:format</code>	Нет	Формат вывода (для чисел, дат, булевых значений).
<code>tpl:multiply</code>	Нет	Если <code>true</code> , колонка будет размножена по числу значений.
<code>tpl:sort</code>	Нет	Задел под сортировку (пока не реализовано).

3.3 Поддерживаемые типы колонок

Если атрибут `tpl:type` не указан, тип определяется по XSD-типу элемента.

Тип	XSD-тип / атрибут	.NET тип
STRING	string, token, normalizedString	System.String
INT	int, integer	System.Int32
UINT	unsignedInt, unsignedShort	System.UInt32
DOUBLE	double, float	System.Double
DECIMAL	decimal	System.Decimal
<i>Продолжение на следующей странице</i>		

Тип	XSD-тип / атрибут	.NET тип
DATE	date	System.DateOnly
TIME	time	System.TimeOnly
DATETIME	dateTime	System.DateTime
BOOLEAN	boolean	System.Boolean
DURATION	duration	System.TimeSpan
BARCODE	(обязателен tpl:type="BARCODE")	генерирует PNG

4 Настройка шаблона документа

4.1 Для DOCX (Microsoft Word / LibreOffice Writer)

1. Вставьте в шаблон элемент управления содержимым (Content Control / Sdt).
2. Откройте свойства элемента:
 - **Alias** = значение `tpl:binding` из схемы.
 - **Tag** = `template`.
3. Для секций (повторяющихся строк таблицы) поместите всю строку в `SdtRow` с `Tag = template` и `Alias =` значению `binding` секции.

4.2 Для XLSX (Microsoft Excel / LibreOffice Calc)

- **Лист**: имя листа должно быть указано в фигурных скобках, например `{ws}` (значение `binding` страницы по умолчанию).
- **Ячейка**: внутри ячейки напишите `{{binding}}` (двойные фигурные скобки).
- **Секция (строка)**: создайте *именованный диапазон* (Named Range) с именем, равным значению `binding` секции. Диапазон должен охватывать ровно одну строку полностью.

5 Штрихкоды (Barcode)

Для генерации штрихкода в схеме укажите:

```
<xs:element name="BarcodeValue" type="xs:string"
  tpl:binding="BarcodeCell"
  tpl:type="BARCODE"
  tpl:format="QR_CODE"
  tpl:width="300"
  tpl:height="300"
  tpl:color="#000000"
  tpl:error_correction="H"/>
```

5.1 Доступные форматы штрихкодов

`tpl:format` может принимать следующие значения (регистронезависимо):

- AZTEC
- CODABAR
- CODE_39
- CODE_93
- CODE_128
- DATA_MATRIX
- EAN_8
- EAN_13
- ITF
- PDF_417
- QR_CODE
- RSS_14
- RSS_EXPANDED
- UPC_A
- UPC_E
- MSI
- PLESSEY
- PHARMA_CODE

5.2 Дополнительные атрибуты штрихкодов

Атрибут	Значение по умолчанию	Описание
<code>tpl:width</code>	400	Ширина изображения в пикселях
<code>tpl:height</code>	70	Высота изображения в пикселях
<code>tpl:color</code>	#0000C8 (синий)	Цвет штрихов (HEX)
<code>tpl:margin</code>	-	Внутренний отступ (пиксели)
<code>tpl:error_correction</code>	L (для QR)	Уровень коррекции ошибок
<code>tpl:character_set</code>		Кодировка символов

6 Репликация данных (повтор блоков)

- Если в XML элемент имеет `maxOccurs="unbounded"` или просто повторяется несколько раз, соответствующая секция или колонка будет автоматически продублирована.
- Для Excel: строки сдвигаются вниз, объединённые ячейки и формулы корректируются.

- Для Word: строки таблиц (SdtRow) копируются вместе с содержимым.

7 Пример данных (data.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<Report>
  <Title> </Title>
  <Date>2025-03-31</Date>
  <Rows>
    <Code> </Code>
    <Amount>1250.50</Amount>
  </Rows>
  <Rows>
    <Code> </Code>
    <Amount>890.30</Amount>
  </Rows>
  <Rows>
    <Code> </Code>
    <Amount>3400.00</Amount>
  </Rows>
</Report>
```

8 Форматы вывода

Система генерирует стандартные Open XML документы:

- .docx — документы Word (совместимы с Microsoft Word, LibreOffice Writer, Google Docs)
- .xlsx — электронные таблицы Excel (совместимы с Microsoft Excel, LibreOffice Calc, Google Sheets)

Технически формат является **Open XML** (ECMA-376), что позволяет в будущем расширить поддержку на .odt и .ods (Open Document Format) с минимальными изменениями.

9 Обработка ошибок

Возможные проблемы при генерации:

- **Не найден файл в архиве** — проверьте пути в manifest.json.
- **Нет привязки в шаблоне** — убедитесь, что в документе присутствует элемент с соответствующим Alias и Tag=template.
- **Неверный формат данных** — проверьте соответствие data.xml схеме schema.xsd.
- **Некорректный штрихкод** — данные не соответствуют требованиям формата (длина, допустимые символы).

10 Требования к среде выполнения

Для работы системы на стороне сервера необходимо:

- Операционная система Windows (из-за аутентификации Negotiate)
- Установленный .NET 8 Runtime
- Доступ к файловой системе для временных файлов (генерируемые документы отдаются в потоке)

11 Пример использования (клиент)

1. Подготовьте ZIP-архив согласно структуре из раздела [2.1](#).
2. Отправьте его HTTP POST-запросом на соответствующий эндпоинт:
 - `/Wordprocessing` для документов Word
 - `/Spreadsheet` для электронных таблиц
3. В ответе получите готовый файл документа.