

ТЗ на разработку шаблонизатора документов

Требования

- 1) **Задача.** Требуется разработать веб-сервис, который будет генерировать документы (word, excel) на основе шаблонов (word, excel) и подставляемых пользовательских данных.
- 2) **Стек технологий.** Веб-сервис должен быть написан на node.js
- 3) **Входные данные.** Шаблонизатор через собственный HTTP API получает запрос на генерацию документа в формате json с двумя обязательными параметрами:

```
{
  template: "url к файлу шаблона",
  data: {json-объект с переменными}
}
```
- 4) **Шаблон.** Представляет собой сверстаный документ в формате docx/dotx или xlsx/xltx. Шаблонизатор должен скачать шаблон.
- 5) **Данные.** Пользовательские данные (data) — объект с вложенными объектами и массивами.
- 6) **Шаблонизация.** В шаблоне в местах, куда нужно вставлять пользовательские данные, указаны переменные в формате `{{переменные из data}}`.
Примеры: `{{invoice_number}}`, `{{executor.bank.title}}`.
- 7) **Множимые данные.** В шаблонах бывают таблицы и списки. Количество элементов в пользовательских данных заранее неизвестно. Поэтому шаблонизатор должен уметь генерировать строки в таблицах (word/excel) и элементы в нумерованных и не нумерованных списках в итоговом документе. Такие значения в шаблоне указываются также: `{{order.items.title}}`. Шаблонизатор, парся выражение в `{{...}}`, должен понимать, что переменная `items` — это массив и множить верстку в шаблоне и подставлять данные. Готовы рассмотреть другое решение через вычисление конструкций `{{}}` через `vm / vm2`.
- 8) **Формат вывода.** Приложение в случае успешной генерации возвращает содержимое сгенерированного файла в формате .docx или .xlsx в зависимости от формата шаблона. После этого удаляет созданный документ и шаблон. Таймаут на генерацию 60 секунд. Если ответ не успешный, то возвращает 400 ошибку и ответ:

```
{
  error: "содержание ошибки от vm2 или внутренней"
}
```
- 9) **Бонус.** При возможности предусмотреть возможность экспорта итогового файла в виде PDF.

Пример

Документ (template)

Пример содержание документа-шаблона в word:

Счет на оплату **{{invoice}}** от **{{date}}**

Банк получателя	
{{executor.bank.name}}	{{executor.bank.bik}}
	{{executor.bank.account}}

Получатель		
{{executor.company}}	{{executor.account}}	
	{{executor.inn}}	{{executor.kpp}}

Поставщик				
{{executor.company}}, {{executor.inn}}, {{executor.kpp}}, {{executor.address}}, {{executor.tel}}				
Заказчик				
{{Customer.company}}, {{customer.inn}}, {{customer.kpp}}, {{customer.address}}				

№	Товар	Кол-во	Цена	Сумма
{{table.number}}	{{table.product}} • {{table.specs}}	{{table.quantity}}	{{table.price}}	{{table.summ}}
			Итого:	{{total_summ}}
Всего наименований {{total_count}}, на сумму {{total_summ}}				

Товары в заказе:

- **{{table.product}}** – **{{table.quantity}}**

Данные (data)

```
{
  invoice: "КП-45",
  date: "31.12.2018",
  executor: {
    bank: {
      name: "Банчище Интернешнл",
      bik: "7770555",
      account: "57567456454254564564"
    },
    company: "ООО «Продавище»",
    account: "4567894564876415678",
    inn: "987564321897",
    kpp: "9875643",
    address: "г. Городще, пр. Проспектов, д.100",
    tel "+7-987-654-32-10":
  },
  customer: {
    company: "ООО «Покупалище»",
    inn: "123456789123",
    kpp: "1234567",
    address: "г. Городок, Ул. Уличная, д.1"
  },
  table: [
    {
      number: 1,
      product: "Продукт 1",
      specs: ["новый", "красный"],
      quantity: 5,
      price: 100,
      summ: 500
    },
    {
      number: 2,
      product: "Продукт 2",
      specs: [],
      quantity: 2,
      price: 2000,
      summ: 4000
    },
    {
      number: 3,
      product: "Продукт 3",
      specs: ["большой", "круглый", "прозрачный"],
      quantity: 10,
      price: 10,
      summ: 100
    }
  ],
  total_count: 3,
  total_summ: 4600
}
```