

ГЕНЕРИРАНЕ НА СПРАВКИ ЧРЕЗ JAVASCRIPT

Теодора Гарджева, Николай Павлов, Асен Рахнев

Резюме. Тази статия представя употребата на библиотеката jsreport за генериране на персонализирани XLSX и PDF справки. Имплементираният интерфейс може да бъде използван за генерирането на разнообразни справки чрез предаването на метаданни на колоните. Решението поддържа неограничен брой колони и редове, както и основните познати типове данни. Приложението приема JSON обект като входящ параметър, което го прави независимо от клиентската платформа или програмен език. Данните в справките могат да бъдат групирани и сортирани по множество критерии. Приложният интерфейс цели да предостави лесен за употреба начин за генериране на справки във формат, подходящ за нуждите на клиентите му.

Ключови думи: *jsreport, xlsx, pdf, handlebars, open office*

1. Въведение

С развитието на бизнеса и интегрирането на информационните технологии в ежедневните му задачи нараства употребата на онлайн справки [5, 6, 7]. Ето защо създадохме абстрактни шаблони за генериране на справки в едни от най-използваните формати: PDF и XLSX. Решението предлага генериране на справки с набор от колони, дефинирани от клиентското приложение. На клиентите на програмния интерфейс се дава възможност да дефинират как да изглеждат техните справки и от каква функционалност се нуждае техният бизнес.

2. Реализация

За реализацията на проекта е използвана библиотеката jsreport [1], която е базирана на javascript инструменти за генериране на съдържание. За да бъде генерирано съдържанието на една справка в jsreport [1], то трябва да бъде избрана библиотека за дефиниране на стиловете и разположението му, както и библиотека за конвертиране на резултата в желанния формат (PDF

или XLSX). За стилизиране на съдържанието е използвана библиотеката `handlebars` [2], която дава възможност за свързване с резултатните данни, итерирането им и създаване на условни функции. За конвертиране на информацията във файл в PDF или XLSX са използвани съответно `chrome-pdf` и `xlsx` библиотеки. Първата позволява интеграция на HTML синтаксис и `handlebars` [2] свързване с данни, докато `xlsx` следва Open Office XML [3] синтаксис с `handlebars` [2] свързване на данни. `Handlebars` [2] позволява обхождане на масиви от данни, което прави възможно прилагането на една и съща стилизация на различни данни. Създадени са два абстрактни шаблона за генериране на PDF и XLSX справки. Имплементацията и на двата шаблона работи с метаданни. Чрез един и същ шаблон могат да бъдат генерирани справки с различни на брой и тип колони. Шаблоните са независими от данните, които ще показват. Програмният интерфейс очаква да получи метаданни на колоните от клиентските приложения. Подаваните параметри са в JSON формат, което не обвързва клиентите с определен език за имплементация.

```
let requestParam = {
  data: [
    {
      name: 'John Doe',
      age: 30,
      children: 2,
      birthDate: '1988-03-07[T]10:00:00',
      salary: 1235.66
    }
  ],
  columns: [
    {key: 'name', type: 7, name: 'Fullname'},
    {key: 'age', type: 3, name: 'Age'},
    {key: 'birthDate', type: 5, name: 'Birth date'},
    {key: 'birthDate', type: 4, name: 'Birth time'}
  ]
};
```

Фигура 1. Примерни входящи параметри

Всяка колона може да бъде описана със следните свойства: име на полето в JSON обекта, тип на данните и текст за визуализация на колоната. Данните могат да бъдат форматиращи в някой от изброените типове: два или един знак след десетичната запетая, цяло число, час, дата, дата и час, текст.

Заявката за генериране на персонализирана заявка може да съдържа също опционални допълнителни параметри за групиране и сортиране на резултатните данни.

```
let requestParam = {
  data: [...],
  columns: [
    {key: 'name', type: 7, name: 'Fullname'},
    {key: 'age', type: 3, name: 'Age'},
    {key: 'birthDate', type: 5, name: 'Birth date'},
    {key: 'birthDate', type: 4, name: 'Birth time'}
  ],
  groupBy: [2, 0],
  sortBy: [2]
};
```

Фигура 2. Опционални параметри

Поддържа се сортиране и групиране по множество колони. Като се подават индекси на колоните, по които ще се сортира/групира.

Справките поддържат също форматиране на данни за различни часови зони. При добавяне на опционален параметър “timezone: ‘Europe/Sofia’” колоните дата и час се форматира за зададената часова зона.

3. Шаблон за генериране на PDF справки

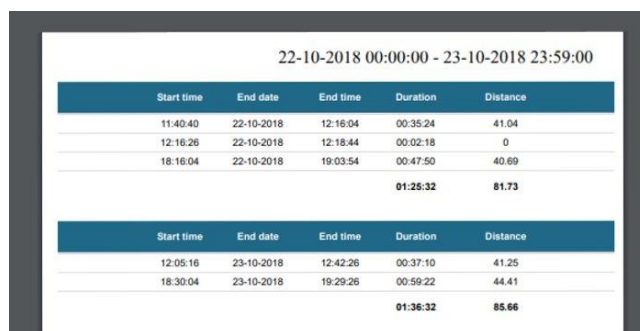
Чрез синтаксиса на handlebars.js [2] за свързване с данни, генератора на PDF chrome-pdf и HTML тагове е създаден шаблон за генериране на справки в табличен вид. Входящите данни се трансформират в данни за header и body на таблицата. Като в прилежащия код стойностите на колоните се форматира спрямо зададените метаданни на колоните. Данните биват сортирани и групирани и се подават на шаблона като обикновени масиви с текстове, за да бъдат показани в този вид.

Ако са зададени условия за групиране, в резултатната справка се изобразяват данни в табличен вид за всяка отделна група: по една таблица за една група. Сортировката се прилага на всяка таблица.

```
{{#each group.values as |cells itemInd|}}
  <tr>
    {{#each cells as |vals valInd|}}
      <td>{{this}}</td>
    {{/each}}
  </tr>
{{/each}}
{{#if group.totalRow}}
  <tr>
    {{#each ../gridConfig.columns as |col colInd|}}
      <td>
        {{getTotalCellValue group.totalRow colInd}}
      </td>
    {{/each}}
  </tr>
{{/if}}
```

Фигура 3. Фрагмент код за генерация на справка

В метаданните на колоните може да бъде добавена опцията “total: true”, която активира сумиране по тази колона. При наличие на поне една сумираща колона, се добавя допълнителен ред към таблицата с резултатните суми. Ако справката показва повече от една таблица, се добавя сумиращ ред за всяка таблица. Може да бъде сумирано по колони от тип: число и час.



22-10-2018 00:00:00 - 23-10-2018 23:59:00

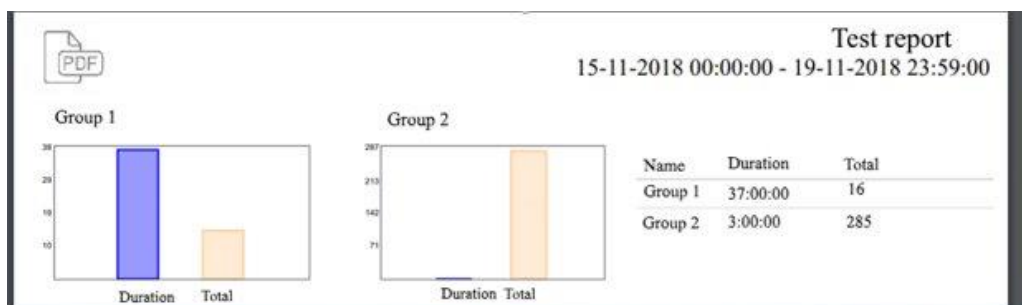
Start time	End date	End time	Duration	Distance
11:40:40	22-10-2018	12:16:04	00:35:24	41.04
12:16:26	22-10-2018	12:18:44	00:02:18	0
18:16:04	22-10-2018	19:03:54	00:47:50	40.69
			01:25:32	81.73

Start time	End date	End time	Duration	Distance
12:05:16	23-10-2018	12:42:26	00:37:10	41.25
18:30:04	23-10-2018	19:29:26	00:59:22	44.41
			01:36:32	85.66

Фигура 4. Резултатна PDF справка

PDF справките дават допълнителни опции за стилизиране на съдържанието. С входящите параметри могат да бъдат изпратени данни за визуализиране на заглавна шапка (header). Шаблонът за генериране на PDF справки приема лого в base64 формат или URL и име на генерираната справка. За всяка справка автоматично се добавя footer с информация за дата, час и странициране.

Абстрактният шаблон за генериране на персонализирани PDF репорти поддържа също генериране на специфични данни с графики. Към входящите параметри за генериране на справка може да бъде добавен опционален параметър “graphic: [column1, column2, ...]”. За подадените колони ще бъде генерирана графика, която сравнява сумарните стойности на тези колони. Поддържат се графики за сравнение на стойности по колони от тип число и час.



Фигура 5. Генериране на справка с интегрирани графики

4. Шаблон за генериране на XLSX справки

За генерирането на XLSX справки са създадени два шаблона: шаблон със стилове и абстрактен шаблон за визуализация на данните.

Jsreport [1] позволява качване на готов XLSX файл, който да бъде използван от шаблоните за генериране на съдържание. Чрез тази функционалност се дава възможност за улеснен начин за дефиниране на стилове в XLSX [4]. Абстрактният шаблон за генериране на справки използва дефинираните стилове в style.xml на този файл. Така се постига генериране на функционални справки: такива, които поддържат валидни типове на колоните, по които клиентите могат да сортират, сумират и филтрират. Дефинирани са стилове за текст, дата, час, дата и час, цяло число, число с една и два символа след десетичната запетая.

Абстрактният шаблон дефинира визуализацията на данните. Използва се Open Office XML SpreadsheetXML File Format [3] и синтаксиса на handlebars.js [2]. В прилежащия код данните се трансформират в конфигурация с колони и масив от данни. Създадени са форматиращи функции, които трансформират дата и час в стойности валидни за Excel.

```

<row>
  {{#each item as |cellItem cellId|}}
    {{#addCell @root.gridConfig.reportViewColumns cellId 1}}
      <t="inlineStr"><is><t>{{cellItem}}</t></is></c>
    {{/addCell}}
    {{#addCell @root.gridConfig.reportViewColumns cellId 2}}
      <c s="{{getStyle @root.gridConfig cellId}} t="n"><v>{{cellItem}}</v></c>
    {{/addCell}}
    {{#addCell @root.gridConfig.reportViewColumns cellId 3}}
      <c s="{{getStyle @root.gridConfig cellId}}"><v>{{cellItem}}</v></c>
    {{/addCell}}
    {{#addCell @root.gridConfig.reportViewColumns cellId 4}}
      <c t="inlineStr" s="{{getStyle @root.gridConfig cellId}}"><is><t>{{cellItem}}</t></is>
    {{/addCell}}
  {{/each}}
</row>

```

Фигура 6. Фрагмент код за генериране на XLSX справка

XLSX справките поддържат сортиране и сумиране по колони. При зададени критерии за сумиране по колони се добавя функционален сумиращ ред. За всяка колона за сумиране се използва Excel функцията SUM(), която остава активна при интеракция от страна на крайния потребител. Имплементирани са функции, които изчисляват начало и край на сумиращия регион.

XLSX дават възможност за добавяне на допълнителен заглавен ред в параметъра “groupHeaders”. Поддържа се полето “colspan”, който чрез форматираща функция обединява клетки в заглавните редове.

	A	B	C	D	E	F	G
1	Start date	Start time	End date	End time	Duration	Distance	
2	22-10-2011	11:40:40	22-10-2011	12:16:04	0:35:24	41.04	
3	22-10-2011	12:16:26	22-10-2011	12:18:44	0:02:18	0.00	
4	22-10-2011	18:16:04	22-10-2011	19:03:54	0:47:50	40.69	
5	23-10-2011	12:05:16	23-10-2011	12:42:26	0:37:10	41.25	
6	23-10-2011	18:30:04	23-10-2011	19:29:26	0:59:22	44.41	
7							
8							
9							

Фигура 7. Резултатна XLSX справка

Реализираният проект предлага генериране на разнообразни справки. Имплементацията е лесна за надграждане при разрастване на нуждите на клиентите. Дава се възможност клиента да използва само тази част от функционалността, която му е нужна. Реализирания проект е интегриран и успешно използван в ежедневните дейности на фирма за проследяване на автопарк.

Благодарности

Авторите изказват благодарност към научен проект ФП17-ФМИ-008 „Иновационни софтуерни инструменти и технологии с приложения в научни изследвания по математика, информатика и педагогика на обучението“ към Фонд „НИ“ на ПУ „Паисий Хилендарски“, за частичното финансиране на настоящата работа

Използвана литература

- [1] jsreport сървър за справки, <https://jsreport.net/>
- [2] handlebars, <https://handlebarsjs.com/>
- [3] Apache Open Office, <http://www.openoffice.org/xml/general.html>
- [4] Excel дефиниране и добавяне на стилове,
<https://www.excel-easy.com/examples/cell-styles.html>
- [5] Pavlov, N., *Object-Oriented Framework for Development of Distributed Business Applications*, Ph.D. Thesis, Plovdiv University, Plovdiv, Bulgaria, 2011, (in Bulgarian).
- [6] Pavlov, N., Dobрева M., Rahnev A., Spasov G., Automatic Report Generation, *Сборник с доклади на научна конференция „Иновационни ИКТ: Изследвания, разработка и приложения в бизнеса и обучението“*, Пампорово, 24–25 ноември, 2016, pp. 21–28, ISBN: 978-954-8852-72-2.
- [7] Pavlov, N., Dobрева M., Pivot Reporting Tool, *Сборник с доклади на научна конференция „Иновационни ИКТ: Изследвания, разработка и приложения в бизнеса и обучението“*, гр. Хисар, 11–12 ноември 2015 г., pp. 21–30, ISBN: 978-954-8852-56-7.

Факултет по математика и информатика

Пловдивски университет „Паисий Хилендарски“

бул. „България“ № 236, Пловдив 4003, България

e-mails: teodoragardjeva@gmail.com, nikolayp@uni-plovdiv.bg,

asen@uni-plovdiv.bg

JAVASCRIPT REPORT GENERATOR

Teodora Gardjeva, Nikolay Pavlov, Asen Rahnev

Abstract. This paper presents how we can use jsreport library to generate personalized reports in XLSX and PDF formats. The implemented interface can be used to generate various reports by supplying metadata for report columns. The system supports unlimited number of rows and columns and the most common data types. The input is a JSON object, which makes it independent from the client platform and programming language. Data can be grouped and sorted by multiple criteria. The application programming interface aims to present an easy to use method to generate reports.