

Реализация библиотеки для поточковой обработки .xlsx файлов

Свитков Сергей

группа 344

научный руководитель Ю.В. Литвинов

консультант М.В. Заведеев

СПбГУ

кафедра системного программирования

4 мая 2018 г.

Введение

- ▶ Веб-приложения в сфере биллинга, телекоммуникаций
- ▶ Различные отчеты, статистика
- ▶ Формат .xlsx
- ▶ Поточковая обработка для экономии памяти

Существующие решения

- ▶ Apache POI
 - ▶ До версии 3.8 — только работа in-memory
 - ▶ Начиная с 3.8 — Stream-API для записи, Event-API для чтения
 - ▶ Часть операций всё равно только in-memory
 - ▶ Отсутствие полных и подробных примеров работы с Stream-API
- ▶ SJXLSX
 - ▶ Не поддерживается с 2015 года
- ▶ Excel Streaming Reader
 - ▶ Разработка коммьюнити
 - ▶ Обертка над POI

Постановка задачи

- ▶ Изучить существующие решения и формат `xlsx`
- ▶ Реализовать библиотеку для потоковой обработки `xlsx` файлов
- ▶ Опубликовать библиотеку в Maven
- ▶ Провести апробацию
- ▶ Сравнить полученную реализацию с существующими

XLSX

- ▶ zip-архив с xml файлами
- ▶ Структура:
 - ▶ Content_Types.xml — типы контента в архиве и пути к ним
 - ▶ _rels — зависимости между файлами внутри архива
 - ▶ docProps — метаданные: имя автора, дата создания, ...
 - ▶ xl — директория с основными файлами архива: workbook, страницы, стили, таблицы

XLSX

- ▶ Workbook:
 - ▶ Метаданные
 - ▶ Ссылки на страницы с данными
- ▶ Worksheet:
 - ▶ Содержат данные
 - ▶ 3 формата представления данных: Grid, Chart, Dialog
- ▶ Grid:
 - ▶ Данные разбиты на ряды;
 - ▶ Каждый ряд состоит из ячеек, в которых хранятся значения
 - ▶ В каждой ячейке хранятся номер ряда и тип значения

Реализация: подход

► Запись:

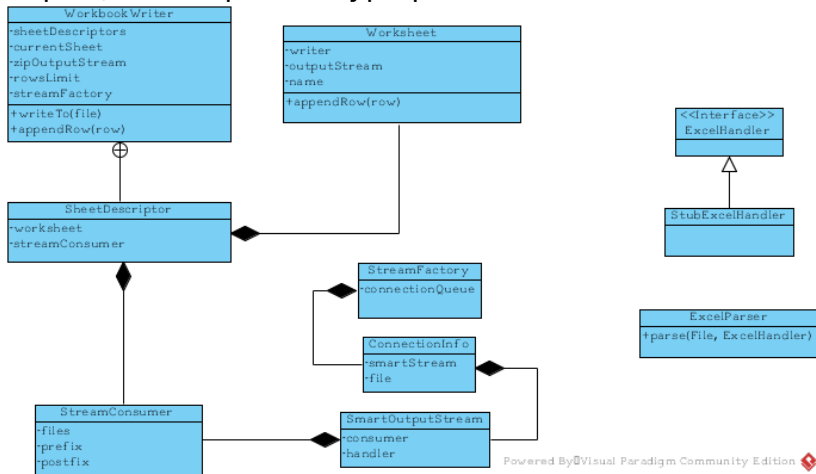
- Для каждой страницы создавать временный файл
- Хранить в RAM только один ряд (во время создания)
- После создания добавлять ряд во временный файл страницы
- После завершения формирования документа — записывать данные из временных файлов в основной файл
- Для экономии дискового пространства сжимать временные файлы

► Чтение:

- Использовать парсер POI
- Реализовать механизм передачи прочитанных данных пользовательскому обработчику

Реализация: архитектура

Упрощенная архитектура решения:



Текущие результаты

- ▶ Проведён анализ существующих решений и формата xlsx
- ▶ Разработана архитектура библиотеки
- ▶ Почти закончена реализация библиотеки