**Обзор библиотек для работы с JSON файлами**

**Jackson**

Jackson имеет три метода обработки формата json, каждый из них имеет свои «за» и «против»:

1. Потоковый API: читает и записывает данные json как последовательность событий (events);
2. Модель дерева: формирует представление json файла в виде дерева (вложенность одних объектов в другие), как DOM в формате xml;
3. Конвертирование json в POJO (plain old java object) и наоборот;
4. Высокая производительность при работе с большими файлами.

Плюсы Jackson заключаются в том, что он содержит в себе много функций и является хорошим инструментом для чтения и записи json файлов различными способами. Но в то же время из-за большого количества функций его размер становится недостатком, если нужно выполнить какие-то простые действия с форматом json (например, преобразовать java object в json string).

**GSON**

1. Данная библиотека преобразует Java объекты в JSON строки и наоборот. Ставить аннотации в классах при этом не надо;
2. Также имеет функцию «красивой записи» («Pretty printing») в json файл: каждый элемент в новой строчке, ставятся отступы;
3. Имеет полную поддержку Java Generics;
4. Низкая производительность при работе с большими файлами.

**Json-simple**

1. Простая и легковесная библиотека, имеющая всего 12 классов;
2. Поддержка Java IO readers и writers;
3. Модель дерева: формирует представление json файла в виде дерева (вложенность одних объектов в другие), как DOM в формате xml;
4. Высокая производительность при работе с любыми файлами;
5. Не имеет зависимостей от внешних библиотек.

**JSON-P**

1. Средняя производительность (по сравнению с библиотеками выше);
2. Создание и использование данных json в потоковом режиме (как SAX и StAX в формате xml)
3. Модель дерева: формирует представление json файла в виде дерева (вложенность одних объектов в другие), как DOM в формате xml;

Для выполнения задания была взята библиотека Json-simple, так как она наиболее производительная из всех перечисленных, наиболее проста в работе и имеет модель представления json файла в виде дерева, что облегчает работу с данным типом файлов. Однако при записи в json файл, после создания json-строки (при помощи библиотеки Json-simple), она была записана при помощи «красивой записи» из библиотеки GSON.