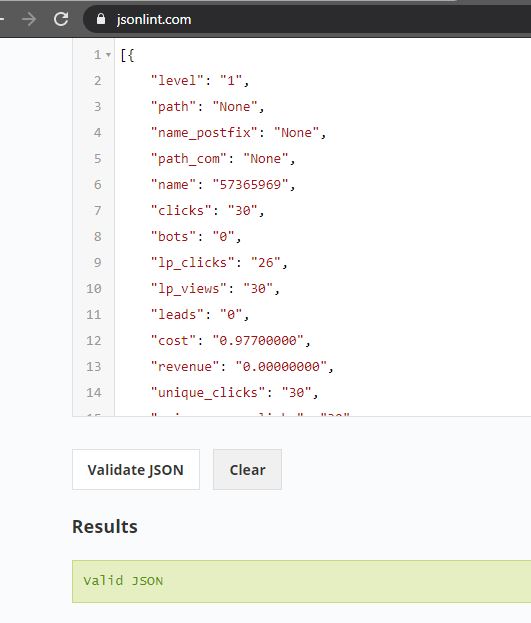
***Инструкция по преобразованию файлов JSON в единый JSON и Excel***

1. Нужно подготовить файлы JSON

Файлы JSON требуют двойных кавычек, нужно везде в файлах с помощью редактора VS Code или другого выполнить замену ‘ на “. Также может быть вариант, что в json файле будут данные 'tags': ['{, в них встречается комбинация символов ‘{ и }’, такую комбинацию надо заменить на { и } соответственно. Слово None надо взять в двойные кавычки, если встречается None заменить на “None”. Также может быть вариант, что будут слова без кавычек как значение name, то есть, обычно name сопровождается значением 'name': '57084429s4217144646', но может быть, что вместо номера будет слово без кавычек (например, browser, widget\_id), тогда данные слова browser или widget\_id надо взять в двойные кавычки.

Чтобы быстро проверить себя на валидность каждого отдельно взятого файла JSON нужно использовать сервис <https://jsonlint.com/>

Вот пример рандомного файла с правильной валидацией



1. Нужно добавить в скрипте все имена файлов в массив arrayOfJsonsFiles в начале скрипта. Это можно сделать быстро, если прописать в терминале команду ls > all.txt (любое название файла для вывода имен). Далее открыть этот файл в VS Code или другом редакторе и сделать замены:

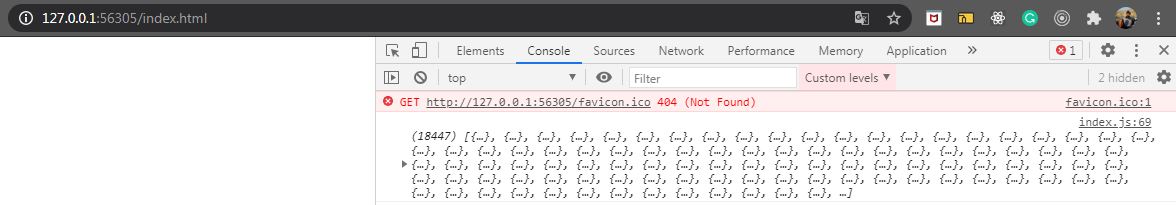
.txt на пустую строку

(json(\d){1,3}) на "$1", (берет названия файлов в кавычки, после “$1” идея запятая и пробел)

\n заменяем на пустую строку (убирает новые строки для каждого отдельного файла и делает все в одну строку)

Вставляем результат в массив.

1. Если все файлы после замен будут валидными, то функция dataStructuring выведет в console.log без ошибок массив с объектами и количеством данных объектов в нем (строка 69 в скрипте)



1. Если после данного этапа, для проверки себя на финише, нужно получить полный json файл единый, то нужно сделать следующие шаги в коде:

* В функции dataStructuring закомментировать строки

// const uniqueArray = findUniqueName(response);

// const fullArray = SummaryOfSameElements(response, uniqueArray);

// const readyArray = RemoveOfSameElements(fullArray, uniqueArray);

(70-72 строки)

* В 73 строке передать параметром функции toJson() переменную response

const json = toJson(response);

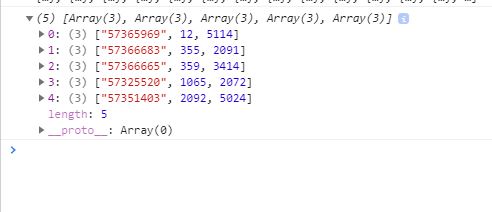
* В функции putToJson расскоментировать a.click() – это сохранит файл в браузере
* После этого вернуть все изменения обратно, чтобы дальше работать с данными в скрипте

1. Функция RemoveOfSameElements на строке 163 делал console.log массива с объектами уже после работы с данными – это конечный массив с объектами. На данном этапе нужно проверить себя, что мы удалили все повторяющиеся строки, кроме первых строк с одинаковыми именами name. Это делается так:

* Нужно раскомментировать строку 164 с

console.log(findUniqueName(fullArray));

Это даст нам лог



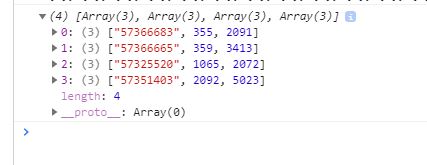
Если массив будет пустой, то значит все строки уникальные, если же массив вернет дополнительные массивы, как тут, то есть еще повторяющиеся строки

В данном примере 5 массивов (первое значение в каждом – имя уникальной строки, второй значение – индекс строки в массиве, где она впервые встречается, последующие значения – индекс строки в массиве, где она встречается повторно, в данном примере, таких повторных появлений всего по одному для каждого name, нужно удалить эти строки по их индексам.

* В строках 161 и 162 меняем значения (добавляем стоки, если потребуется):

fullArray.splice(5114, 1); Удалям индекс в массиве с номером 5514

Важно понимать, что splice после удаления индекса, сдвигает все индексы на одно значение, поэтому снова смотрим лог



Теперь массивов 4 и мы удаляем индекс 2091

fullArray.splice(2091, 1);

Повторяем действия пока не получим лог с пустым массивом

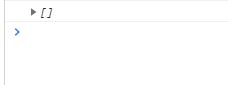
fullArray.splice(5114, 1);

fullArray.splice(2091, 1);

fullArray.splice(3413, 1);

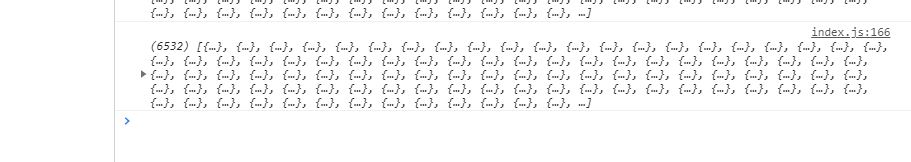
fullArray.splice(2072, 1);

fullArray.splice(5021, 1);



1. Теперь мы имеем массив с объектами в законченном виде

Функция RemoveOfSameElements на строке 166 делает вывод лога



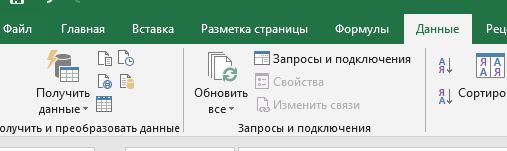
Таким образом, было 18447 строк, получилось 6532 строки с уникальными name.

1. Переходим к этапу импорта данных из JSON в Excel

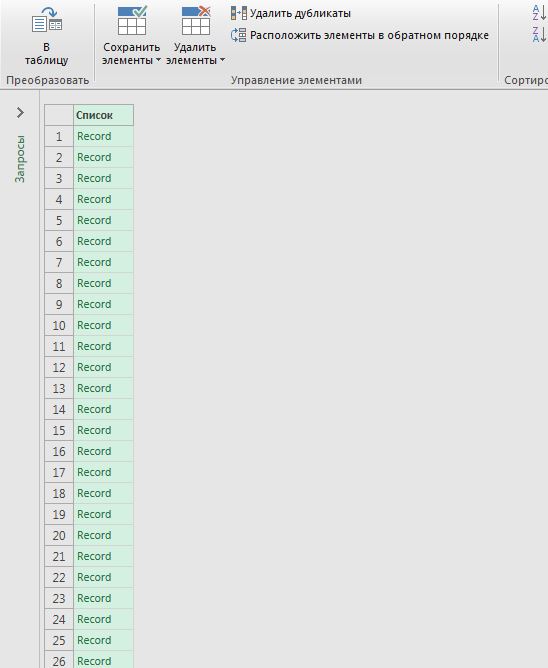
* Снова расскоментируем строку 87 с a.click();

Это сохранит файл в браузере, после сохранения закомментируйте ее снова, чтобы файлы не сохранялись повторно.

* В пустой книге Excel переходим в Данные - Получить Данные – Из файла – Формат JSON

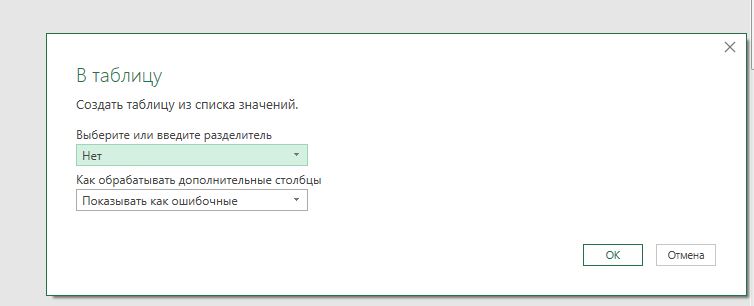


* Получаем промежуточную таблицу с данными, где каждый Record – это наша строка



Нажимает кнопку В таблицу

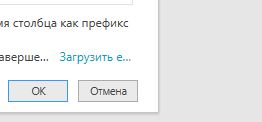
* Далее просто жмем ок



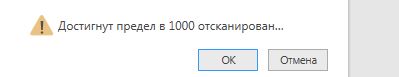
* Нажимаем кнопку разворачивания данных



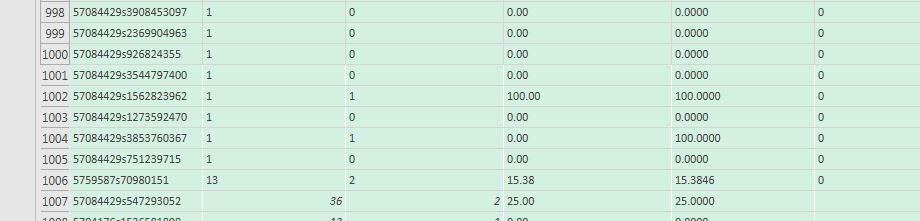
* Выбираем загрузить еще



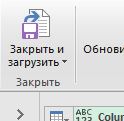
* Не обращаем внимание на ограничение в 1000 строк, данные выведутся все, нажимаем ок



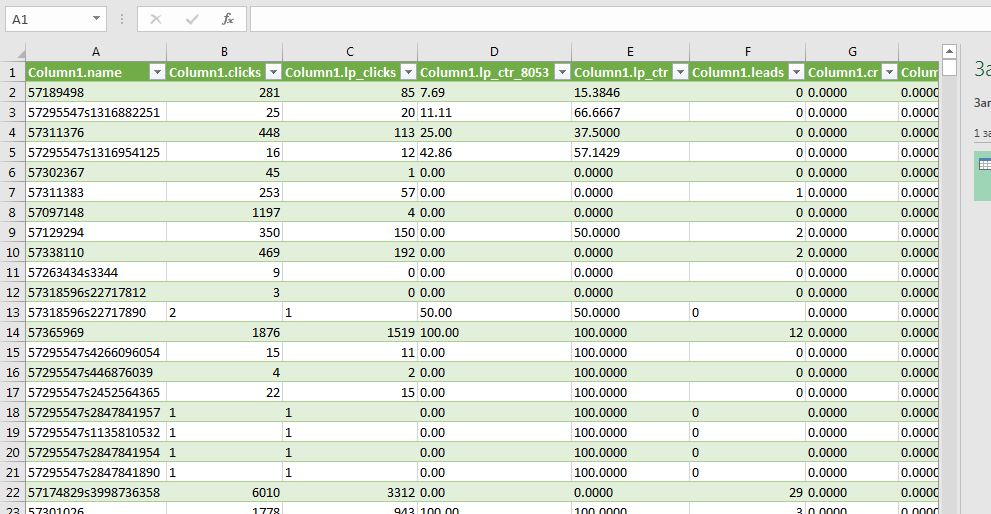
* Получаем данные, проверяем, что строк больше 1000



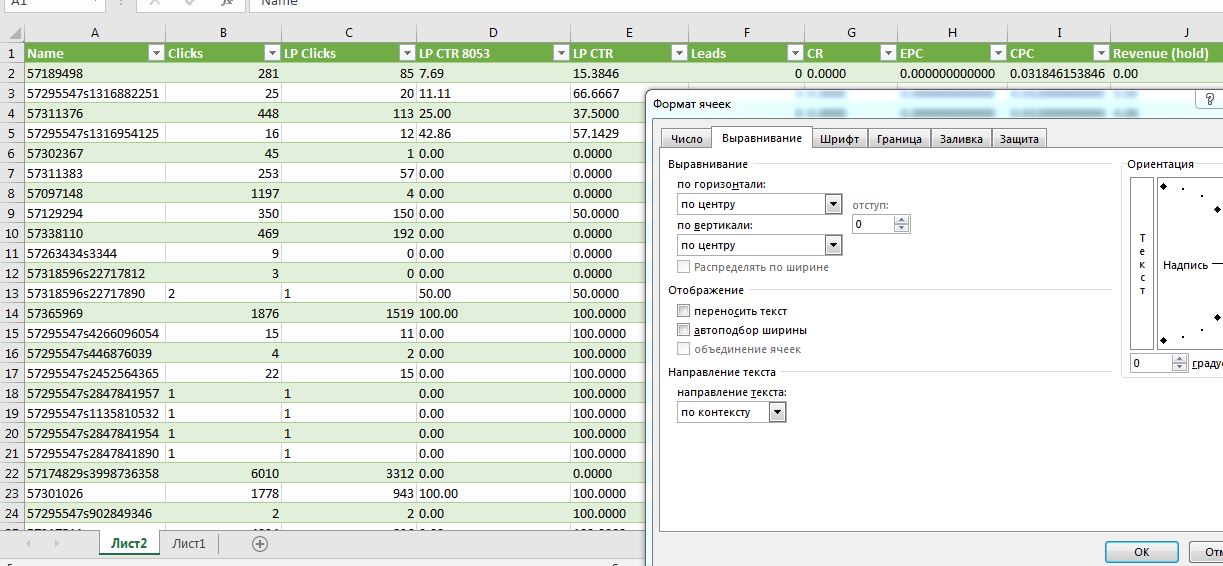
* Нажимаем кнопку Закрыть и загрузить



* Получаем готовую таблицу, которую нужны привести в нормальный вид путем выставления формата ячеек и редактирования названия столбцов



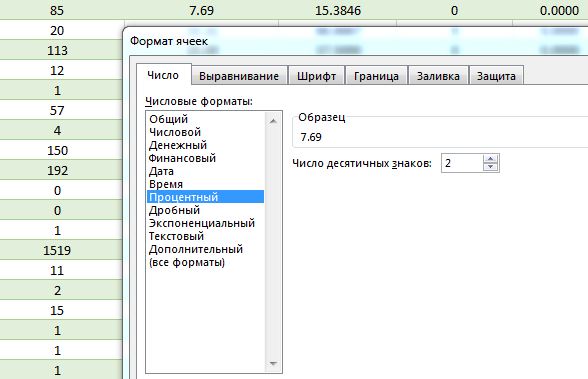
* Если на этапе 4 делали промежуточный Excel файл, то можно проверить себя, выбрать поле name, которое повторяется, например, 57351403 и посмотреть, что значения по полям clicks, lpClicks, leads, revenue, cost, profit будет равняться сумме значения по данным полям для всех одинаковых name таблица Excel, полученной на этапе 4.
* Форматируем названия Столбцов – убираем слово Column, оставляем только слово после точки с большой буквы и выравниваем все значения по центру в ячейках, для этого выделяем все ячейки и выбираем формат ячеек

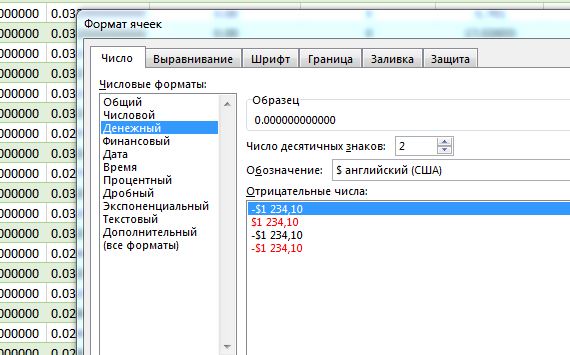


* Выставляем форматы данных для ячеек в таблице

Для LP CTR 8053, LP CTR, CR, ROI - %

Для EPC, CPC, Revenue (hold), Revenue, Cost, Profit, Optimal Bid – доллары США





* Получаем таблицу в финальном виде:

