

ВАРЛАМОВА С. А., МЯЛИК П. Н.
ИСПОЛЬЗОВАНИЕ API ПРИ СОЗДАНИИ ВЕБ-ПРИЛОЖЕНИЯ ПО
ОЦЕНКЕ РЕКЛАМНЫХ ВЛОЖЕНИЙ

УДК 338.24+004.9, ВАК 05.13.11, ГРНТИ 50.01.14

Использование API при создании веб-приложений по оценке рекламных вложений

Using the API when creating a web-applications for advertising investments evaluating

С. А. Варламова, П. Н. Мяслик

S. A. Varlamova, P. N. Myalik

Пермский национальный исследовательский политехнический университет, Березниковский филиал; г. Березники

Perm National Research Polytechnic University, Berezniki branch, Berezniki

Цель исследования - разработать веб-приложение, консолидирующее в себе данные о расходах на рекламные источники (такие как, Яндекс Директ и GoogleAds) и данные о их финансовой целесообразности. Это решение позволит владельцам коммерческих организаций, или специалистам, находящимся на должности «маркетолог», наиболее оперативно оценивать целесообразность денежных вложений в рекламные источники трафика, которые приводят пользователей на «продающую» страницу на сайте. Новизна работы заключается в обобщении и упорядочивании всех необходимых материалов для разработки подобного приложения в одной работе.

The purpose of the study is to develop a web application that consolidates data on the costs of advertising sources (such as Yandex Direct and GoogleAds) and data on their financial viability. This solution will allow owners of commercial organizations, or specialists in the position of "marketer", to most quickly assess the feasibility of money investments in advertising sources of traffic that lead users to the "selling" page on the site. The novelty of the work lies in the generalization and ordering of all the necessary materials for the development of such an application in one work.

Ключевые слова: API, Яндекс.Метрика, Яндекс.Директ, GoogleAds, JSON.

Keywords: API, Yandex.Metrica, Yandex.Direct, GoogleAds, JSON

Введение

На сегодняшний день большинство коммерческих организаций, имеют свой сайт в интернет, который чаще всего необходим для привлечения клиентов. Но само по себе наличие «продающей страницы» в интернет не гарантирует сколько-то стабильный поток клиентов, и большинство организаций применяют

один или несколько рекламных инструментов для продвижения сайта в интернет. Наиболее распространенными являются инструменты по настройке контекстной рекламы: Яндекс.Директ и GoogleAds [1, 2].

При этом специалисту организации (в данном случае маркетологу) крайне необходимо оценивать, насколько привлеченная аудитория была «целевой». Для проведения аналитической работы используют такие инструменты, как, например, Яндекс.Метрика.

Яндекс.Метрика – это инструмент веб-аналитики, который помогает получать наглядные отчеты, видеозаписи действий посетителей, отслеживать источники трафика и оценивать эффективность онлайн- и офлайн-рекламы. Позволяет получить и проанализировать данные о пользователях, посетивших сайт в разрезе типа устройства, пола, возраста, географического местоположения, и многих других срезах, но для нас необходимы данные об источнике трафика (из какой рекламной системы был переход, и по какому запросу на поиске), и о совершении им на сайте целевого действия (отправки заявки на товар, заказ обратного звонка и пр. далее «обращение»). В этом сервисе видны все источники трафика на сайт, объем этого трафика, и его качество (в том числе количество обращений, подробная информация о подключении к Яндекс.Метрика в официальной документации [3]).

Аналитика, наиболее эффективного рекламного источника, проводится регулярно большинством специалистов, но основная техническая проблема состоит в том, что Яндекс Метрика не позволит получить данные о прямых денежных затратах на эти источники трафика, и соответственно, сопоставить их с количеством обращений, полученных по источникам трафика, и вывести расчетное значение «стоимости обращения». Которое, в конечном счете, и является основным показателем финансовой целесообразности того или иного источника рекламного трафика. В связи с этим, скорость принятия решения о необходимости перераспределения бюджетов крайне низкая, и в таких случаях финансовые потери могут составлять до 100% всего месячного рекламного бюджета компании.

Для получения стоимости обращения необходимо вывести ее вручную (с помощью excel таблиц, на что будет уходить до нескольких часов ежедневно), либо воспользоваться существующими решениями сквозной аналитики, например, Roistat, Alytics, CoMagic. Эти системы также используют API для получения данных, консолидируют их и представляют в окне браузера.

Достоинства этих систем широко представлены на официальных сайтах, однако есть и некоторые минусы. Во-первых, они все являются платными (от 4500 руб/месяц, до тарифов, где только подключение будет обходиться 60000 руб. – по состоянию на май 2021 года). А также важной особенностью, например, Alytics, является то, что система не хранит никаких данных в собственной БД. Соответственно, в ситуации, когда подключенные системы станут недоступны (отключены), невозможно будет получить данные даже за прошлые периоды.

В данной статье рассмотрена разработка собственного веб-приложения, использующего API различных систем размещения рекламы и аналитических систем, консолидирующего в себе данные о расходах и данные о их финансовой

целесообразности. Это решение позволит владельцам коммерческих организаций, или специалистам, находящимся на должности «маркетолог», наиболее оперативно оценивать целесообразность денежных вложений в рекламные источники трафика.

Алгоритм работы приложения представлен на Рисунке 1.

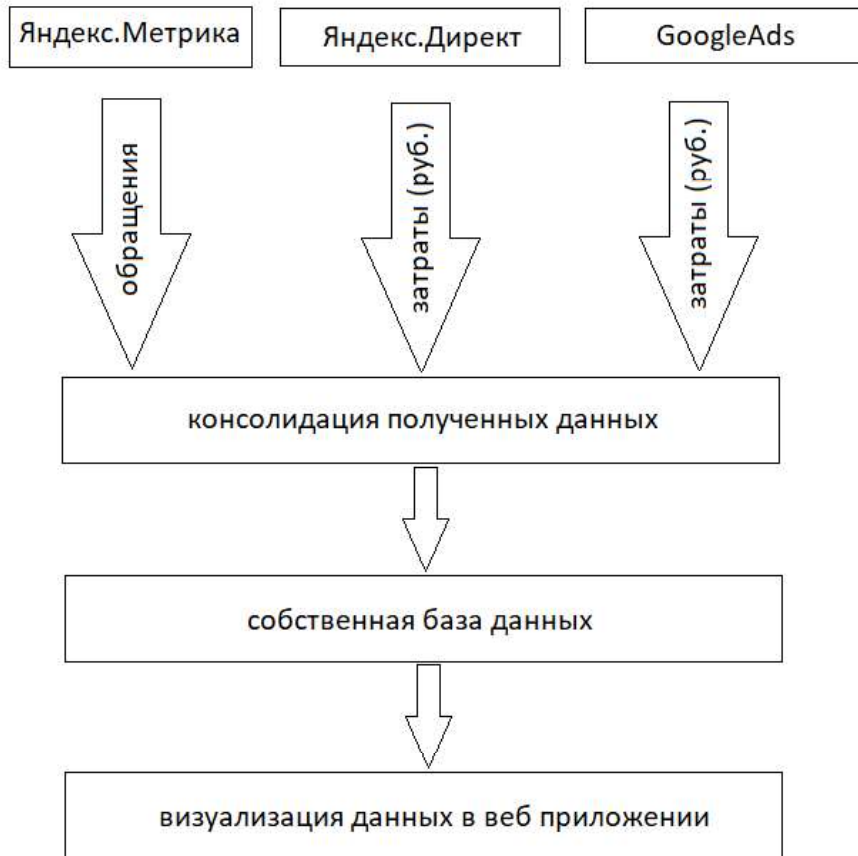


Рисунок 1. Схема получения, обработки и хранения данных

Приложение должно выполнять получение и обработку данных, кроме того необходимо организовать хранение и возможность представлять эти данные в визуально-понятном виде. Для этих целей приложение будет использовать СУБД MySQL, языки программирования SQL (для обращения к собственной базе данных), PHP (для написания основного алгоритма работы приложения), а также HTML, CSS и JavaScript для визуализации данных в браузере.

Что такое API?

Аббревиатура API расшифровывается как «Application Programming Interface» (интерфейс программирования приложений, программный интерфейс приложения). Большинство крупных компаний разрабатывают API для клиентов или для внутреннего использования.

Для реализации нашей цели необходимы API систем: Яндекс Метрика, Яндекс Директ, GoogleAds.

Регистрация приложения

Чтобы приложение могло взаимодействовать с API Яндекс.Метрика, ему необходимо пройти процедуры OAuth авторизации [4] и получить авторизационный токен. (ключ доступа). Предварительно необходимо зарегистрировать свое приложение [5] на OAuth-сервере Яндекса и получить для него персональный идентификатор. После этого можно отправлять запрос для получения токена:

```
https://oauth.yandex.ru/authorize?response_type=token&client_id=<id приложения>
```

Сервер возвращает ответ на этот запрос в виде ссылки с параметрами:

```
https://oauth.yandex.ru/verification_code?dev=True#access_token=<авторизационныйтокен>&token_type=bearer&state=&expires_in=31536000
```

Этот токен (набор символов) необходимо сохранить и использовать далее при создании запросов к API Яндекс.Метрика.

Похожим образом необходимо получить токены для авторизации в Яндекс.Директ и GoogleAds [6].

Получение данных

Получение данных из Яндекс Метрики с помощью API состоит, условно, из двух частей: конструирование ссылки со множеством параметров, и обработка возвращенного JSON массива в обычный массив PHP [7].

Перед созданием кода запроса, необходимо в коде PHP подготовить переменную для номера счетчика Яндекс.Метрики, (8 цифр, выдается при регистрации счетчика); один или несколько идентификаторов «целей метрики», таким образом, в Яндекс.Метрике называются совершение действий на сайте, в том числе обращения.

Листинг кода PHP для создания ссылки-запроса:

```
$sURL = 'https://api-metrika.yandex.net/stat/v1/data?ids='.$sauth_counter.'&metrics=ym:s:visits'.goals_metrics($sauth_goal).'&date1='.$last_date1.'&date2=yesterday&dimensions=ym:s:date,ym:s:from&attribution=last_yandex_direct_click&limit=100000';  
$aHTTP = array('http' => array('method' => 'GET', 'header' => 'Authorization: OAuth <токен авторизации>'));
```

После этого необходимо сделать непосредственно сам запрос и преобразовать данные в пригодные для использования. А также сразу «очистить» массив от «нулевых» строк. Т.е. тех строк, где количество обращений будет 0.

Листинг кода для осуществления запроса и обработки данных:

```
$context = stream_context_create($aHTTP); // авторизация  
$data = json_decode(file_get_contents($sURL, false, $context)); //сам запрос  
$old_data = $data -> data;  
$metrika = json_decode(json_encode($old_data), true); //декодирование полученных данных  
$new_goals = goals_clear($metrika); //собственная функция «очистки» массива
```

Таким образом мы получили данные от Яндекс.Метрики в виде массива, который состоит из строк, каждая из которых содержит:

- дату
- «идентификатор группы_поисковую фразу» (уникальный идентификатор группы фраз и фраза, записанная пользователем в поисковую строку Яндекс, непосредственно перед переходом по рекламному объявлению). Эта информация передается при переходе по рекламному объявлению, в котором добавлены параметры url в виде: ?from={gbid}_{keyword}. В этом случае «gbid» передает идентификатор группы фраз, а «keyword» передает поисковую фразу [8].

- количество обращений.

Далее необходимо получить данные от Яндекс.Директ. В целом процедура схожа с получением данных от Яндекс.Метрики. Параметры будут следующими: Дата «с..», дата «по..», и условие: «количество кликов – больше 0». А поля (возвращаемые данные) необходимо задать следующие: "Date" (дата), "AdNetworkType" (тип рекламной кампании), "CampaignName" (название кампании), "AdGroupName" (название группы объявлений), "Criterion" (поисковая фраза), "Clicks" (количество кликов), "Cost" (стоимость) и "AdgroupId" (идентификатор группы) [9].

Листинг кода формирования ссылки запроса:

```
$url = 'https://api.direct.yandex.com/json/v5/reports';
$token = '<токен авторизации>';
$clientLogin = $auth_login;
$params=["params"=>["SelectionCriteria" => ["DateFrom" =>
"$last_date1", "DateTo" => "$yesterday", "Filter" => [["Field" => "Clicks", "Operator" => "GREATER_THAN", "Values" => ["0"]]]], "FieldNames" => ["Date", "AdNetworkType", "CampaignName", "AdGroupName", "Criterion", "Clicks", "Cost", "AdgroupId", "CriterionType"],,];
$body = json_encode($params);
```

Таким образом сохранены все необходимые параметры для осуществления запроса. Непосредственно запрос осуществляет функция «curl_exec()».

Листинг кода выполнения запроса:

```
$result = curl_exec($curl);
$responseHeadersSize = curl_getinfo($curl, CURLINFO_HEADER_SIZE);
$responseHeaders = substr($result, 0, $responseHeadersSize);
$responseBody = substr($result, $responseHeadersSize);
$httpCode = curl_getinfo($curl, CURLINFO_HTTP_CODE);
$requestId = preg_match('/RequestId: (\d+)/', $responseHeaders, $arr) ?
$arr[1] : false;
$retryIn = preg_match('/retryIn: (\d+)/', $responseHeaders, $arr) ? $arr[1] : 60;
curl_close($curl);
$direct = array();
$strings = explode("\n", $responseBody);
for ($s=0; $s<count($strings)-1; $s++) {
$direct[$s] = explode("\t", $strings[$s]); //разбор ответа в массив
```

После правильного выполнения этого запроса, в переменной «\$direct» будет сохранен двумерный массив данных, в виде строк, каждая из которых содержит:

- дату (столбец 0)
- тип трафика (столбец 1)
- название кампании (столбец 2)
- название группы объявлений (столбец 3)
- ключевой запрос (столбец 4)
- количество кликов (столбец 5)
- стоимость этих кликов (столбец 6)
- идентификатор группы (столбец 7)

Консолидация и запись в базу данных

После корректного получения вышеописанных данных, необходимо их передать в собственную базу данных. Но, перед этим надо осуществить консолидацию данных.

Алгоритм представляет собой цикл, где каждая строка формируется, присваивая значения переменных от массива Яндекс.Директ, а при обработке значения «обращения» вызывается функция, которая возвращает количество обращений, «предназначенное» именно этой строке. В эту функцию передается одна строка из массива Яндекс.Директ, и весь массив Яндекс.Метрика. Алгоритм ее работы показан на схеме (Рисунок 2):

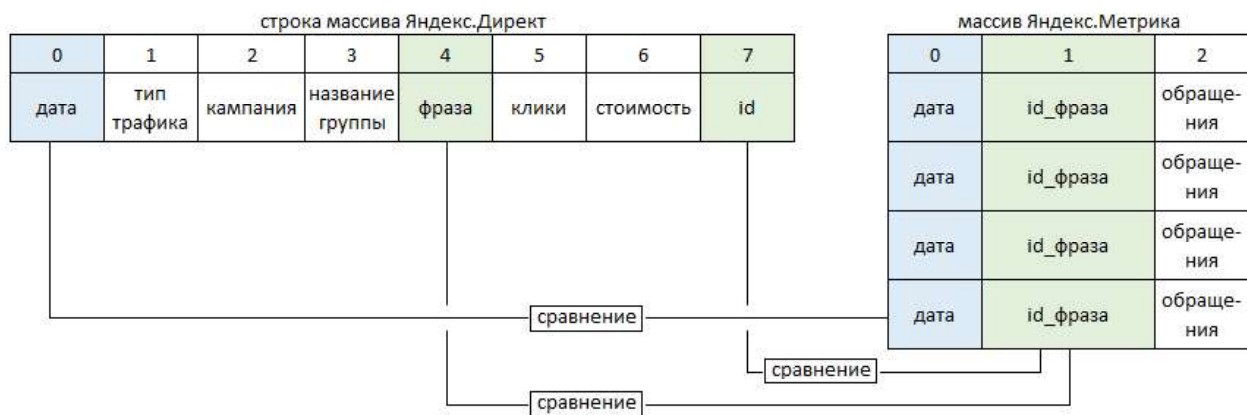


Рисунок 2. Схема консолидации данных

Для того, чтобы однозначно соотнести количество обращений, достаточно сделать одновременное сравнение даты, идентификатора группы и фразы. Это будет означать, что количество обращений относится именно к этой строке из массива Яндекс.Директ.

Листинг кода функции присвоения количества обращений:

```
function goals_fusion($base_string, $metrika_arr){
    $goals = 0;
    if ($metrika_arr !== NULL) {
        if (isset($metrika_arr[0]['metrics'])) {
```

```

        $z = count($metrika_arr[0]['metrics']); }else{
        $z = 2; }
    foreach ($metrika_arr as $q) {
/*непосредственно сравнение данных*/
        if ($q['dimensions'][0]['name'] == $base_string[0] & groups_compare($q['dimensions'][1]['name'], $base_string[7]) & key_compare($q['dimensions'][1]['name'], $base_string[4])) {
            for ($p=1;$p<$z; $p++){ $goals += $q['metrics'][$p];}}}}
        return $goals;}

```

Для наиболее корректного сравнения используются вспомогательные функции «groups_compare()», «key_from_str()», «key_clear()», «key_full_clear()», «key_compare()». Последние 4 осуществляют «очистку» фраз разной степени. Это необходимо потому, что фразы, записанные в Яндекс.Директ часто имеют некоторые вспомогательные операторы, например, знаки «+», «!» и другие. И для корректности, перед сравнением они удаляются.

После корректного срабатывания всех функций, в переменной \$db_goals сохранится количество обращений (в эту дату и по этой фразе).

Итак, мы получили 9 переменных с корректными значениями, которые готовы к записи в собственную базу данных.

Листинг кода записи в БД :

```

/* процедура отправки данных в БД при помощи SQL */
mysql_query($conn, "INSERT INTO `login_query` (`date`, `source`, `medium`, `campaign`, `content`, `term`, `cost`, `clicks`, `goals`) VALUES ('$db_date', '$db_source', '$db_medium', '$db_campaign', '$db_content', '$db_term', '$db_cost', '$db_clicks', '$db_goals');");
}

```

Таким образом, консолидированные данные массивов Яндекс.Директ и Яндекс.Метрика будут записаны в собственную базу данных. И будут доступны для получения и проведения аналитики в любое время.

Визуализация данных в приложении

Полученные данные, при просмотре в СУБД MySQL [10], выглядят, как таблица с большим количеством строк (Рисунок 3).

id	date	source	medium	campaign	content	term	clicks	goals	cost
2	2020-03-22	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории перми	6	0	248.65
3	2020-03-22	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории пермского края	3	0	96.67
4	2020-03-22	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории цена	3	0	85.33
5	2020-03-24	yandex	SEARCH	Лагерь Поиск	Лагерь	лагерь летний оздоровительный	1	0	56.94
6	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории закамскя	1	0	45.97
7	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории перми	11	0	656.95
8	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории пермского края	6	0	157.74
9	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории цена	8	0	491.9
10	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий	санаторий заболевания бронхов	1	0	50.36
11	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий 55+	санатории перми	2	0	50.16
12	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий 55+	санатории пермского края	9	0	319.61
13	2020-03-24	yandex	SEARCH	Санаторий Поиск	Санаторий 55+	санатории цена	9	1	618.06
14	2020-03-25	yandex	AD_NETWORK	Лагерь РСЯ	Лагерь	лагерь летний	1	0	59
15	2020-03-25	yandex	AD_NETWORK	Лагерь РСЯ	Лагерь	лагерь летний детский	2	0	41.12
16	2020-03-25	yandex	AD_NETWORK	Лагерь РСЯ	Лагерь	лагерь пермь летний	2	0	23.26
17	2020-03-25	yandex	AD_NETWORK	Лагерь РСЯ	Ретаргет	Ретаргет Лагерь	7	0	145.75
18	2020-03-25	yandex	SEARCH	Лагерь Поиск	Лагерь	лагерь летний детский	1	1	59.74
19	2020-03-25	yandex	SEARCH	Санаторий Поиск	Санаторий	санатории закамска	1	0	46.51

Рисунок 3. Вид таблицы в базе данных

В таблице видны 5 полей (столбцов), которые на самом деле имеют иерархическую структуру. Это означает, что трафик имеет такое понятие, как «источник» (поле `source`), в источнике могут быть кампании двух типов (поиск и сети; поле `medium`), каждого типа может быть одна или несколько кампаний (поле `campaign`), каждая кампания содержит одну или несколько групп (поле `content`), и каждая группа содержит определенный список фраз, по которым осуществляется показ объявлений (поле `term`).

Необходимо их представить в более понятном и простом виде. Наилучшим образом для этой цели подойдет «древовидная» таблица, которая будет получена за выбранный период (в приложении должен быть выбор периода данных, т.е. даты «с:» и «по:»), а также все числовые данные в рамках выбранного периода будут суммированы. Соответственно, необходим алгоритм, который будет получать данные из нашей БД.

Листинг кода получения данных:

```
$select = mysqli_query($conn, "SELECT * FROM `login_query` WHERE date>='$date1' AND date<='$date2'");
```

```
while ($result = mysqli_fetch_array($select)) {
        $actuals[0][$i] = $result['source'];
        $actuals[1][$i] = traffic_type($result['medium']); //функция переименования типа трафика
        $actuals[2][$i] = $result['campaign'];
        $actuals[3][$i] = $result['content'];
        $actuals[4][$i] = $result['term'];
        $actuals[5][$i] = $result['clicks'];
        $actuals[6][$i] = $result['goals'];
        $actuals[7][$i] = $result['cost'];
}
```



```
    $i++;  
}  
mysqli_close($conn);
```

После его исполнения в переменной будет сохранен массив необходимых нам данных, за выбранный период.

Непосредственно визуализация данных может выполняться различными способами, наиболее удобными для каждого пользователя или разработчика подобного приложения. В данном приложении, как было упомянуто ранее, визуализация осуществлена с помощью «древовидной» таблицы. Способ довольно ёмкий с точки зрения написания кода, но при этом, наиболее удобен для изучения полученных данных. У данной работы нет цели глубоко изучить методы визуализации больших табличных данных, поэтому листинг кода самой древовидной таблицы, со всеми вспомогательными функциями будет опущен. Подобные методы описаны в источнике [11].

Но, самое главное, при построении такого метода, учесть необходимость вывода такого показателя, как «стоимость обращения». Его необходимо рассчитать непосредственно перед визуализацией, т.к. такой показатель не может храниться в нашей БД, ведь он всегда рассчитывается на весь массив полученных данных.

В итоге правильного написания кода визуализации, получается таблица, в которой на первом этапе видны источники рекламного трафика, с количеством кликов, потраченной суммой, количеством обращений (или «заявок»), и стоимостью обращения (или «ценой заявки»). При нажатии на название одного из источников (или на «+»), таблица раскроется, и станут видны один или несколько типов трафика, и них будут также видны собственные показатели. Далее, при нажатии на название любого из типов трафика, таблица снова раскроется и станут видны кампании со своими показателями. Таким же образом можно углубиться до групп. И также можно углубиться до конкретной поисковой фразы, по которой был переход пользователя. И на каждом этапе раскрытия этой таблицы можно увидеть, и проанализировать обобщенные данные по этой категории (Рисунок 4).

Свернуть всё	Развернуть всё							
Источник рекламы	Тип трафика	Компания	Группа	Фраза	Клики	Заявки	Стоимость	Цена заявки
ИТОГО					1893	20	33135.63	1656.78
+ google					56	4	193.02	48.26
- yandex					1837	16	32942.61	2058.91
	+ поиск				443	8	17279.8	2159.98
	- semu				1394	8	15662.81	1957.85
		+ Creta РСЯ			262	3	2812.4	937.47
		+ H1 РСЯ			44	0	68703	--
		+ Santa Fe РСЯ			230	0	2975.18	--
		- Solaris РСЯ			251	0	2818.65	--
			+ Купить		34	0	353.48	--
			+ Общие		196	0	2276.34	--
			+ Рестораны		1	0	9.37	--
			- Широкие		20	0	179.46	--
				hyundai solaris	2	0	12.2	--
				хендай солярис	18	0	167.26	--
		- Tucson РСЯ			242	1	2814.27	2814.27
			+ Купить		110	0	1306.49	--
				hyundai tucson крегу	69	0	804.07	--
				hyundai tucson купим	5	0	62.7	--
				tucson наличие	3	0	54.23	--
				купить туссан автогу	2	0	23.69	--

Рисунок 4. Древоподобная таблица

Заключение

В статье рассмотрены возможности применения API Яндекс.Метрики, Яндекс.Директ и GoogleAds. Рассмотрена процедура регистрации приложения и получения токенов авторизации, и описана разработка приложения, которое способно получать наиболее важные для аналитики данные, консолидировать их, сохранять в собственной БД и далее визуализировать в максимально удобном и структурированном виде. Такое приложение способно экономить время специалиста (до нескольких часов ежедневно на составление подобных таблиц вручную [12]); экономить на тарифах существующих систем сквозной аналитики; и максимально быстро принимать решения о перераспределении денежных вложений в тот или иной источник рекламного трафика, и повышать эффективность этих вложений.

Список использованных источников и литературы:

1. Документация Яндекс.Директ [Электронный ресурс]. – Режим доступа: <https://yandex.ru/support/direct/> (дата обращения: 16.04.2021).
2. Документация GoogleAds [Электронный ресурс]. – Режим доступа: <https://support.google.com/google-ads/answer/7635053?hl=ru> (дата обращения: 17.04.2021).
3. Документация Яндекс.Метрика [Электронный ресурс]. – Режим доступа: <https://yandex.ru/support/metrika/index.html> (дата обращения: 17.04.2021).
4. Документация по OAuth-авторизации для API-сервисов Яндекс [Электронный ресурс]. – Режим доступа: <https://yandex.ru/dev/oauth/doc/dg/concepts/about.html> (дата обращения: 17.04.2021).

5. Регистрация приложения на OAuth-сервере Яндекса [Электронный ресурс]. – Режим доступа: <https://yandex.ru/dev/direct/doc/dg/concepts/register.html> (дата обращения: 18.04.2021).
6. Регистрация приложения на OAuth-сервере Google [Электронный ресурс]. – Режим доступа: <https://developers.google.com/google-ads/api/docs/first-call/oauth-cloud-project> (дата обращения: 18.04.2021).
7. Описание JSON [Электронный ресурс]. – Режим доступа: <https://www.json.org/> (дата обращения: 20.04.2021).
8. Параметры, передаваемые в URL объявлений [Электронный ресурс]. – Режим доступа: <https://yandex.ru/support/direct/statistics/url-tags.html?lang=ru> (дата обращения: 21.04.2021).
9. Допустимые поля API запросов [Электронный ресурс]. – Режим доступа: <https://yandex.ru/dev/direct/doc/reports/fields-list.html> (дата обращения: 21.04.2021).
10. Справочное руководство по MySQL [Электронный ресурс]. – Режим доступа: <http://www.mysql.ru/docs/man/What-is.html> (дата обращения: 21.04.2021).
11. Как создать сворачивающуюся древовидную таблицу в html/css/js? [Электронный ресурс]. – Режим доступа: <https://coderoad.ru/5636375/> (дата обращения: 21.04.2021).
12. Затонский А. В., Варламова С. А. Информационное обеспечение поддержки принятия решений на примере составления расписания занятий образовательной организации // Вестник Южно-Уральского государственного университета. Серия : Компьютерные технологии, управление, радиоэлектроника. 2018. – Т. 18. № 3. – С. 88-106.

List of references

1. Yandex.Direct documentation, <https://yandex.ru/support/direct/>, accessed April 16, 2021.
2. GoogleAds documentation, <https://support.google.com/google-ads/answer/7635053?hl=ru>, accessed April 17, 2021.
3. Documentation Yandex.Metrica, <https://yandex.ru/support/metrica/index.html>, accessed April 17, 2021.
4. Documentation on OAuth authorization for Yandex API services, <https://yandex.ru/dev/oauth/doc/dg/concepts/about.html>, accessed April 17, 2021.
5. Registering the application on the Yandex OAuth server, <https://yandex.ru/dev/direct/doc/dg/concepts/register.html>, accessed April 18, 2021.
6. Registering the application on the Google OAuth server, <https://developers.google.com/google-ads/api/docs/first-call/oauth-cloud-project>, accessed April 18, 2021.
7. Description of JSON, <https://www.json.org/>, accessed April 20, 2021.
8. Parameters passed to ad URL, <https://yandex.ru/support/direct/statistics/url-tags.html?lang=ru>, accessed April 21, 2021.
9. Allowed fields of API requests, <https://yandex.ru/dev/direct/doc/reports/fields-list.html>, accessed April 21, 2021.

10. MySQL Reference Manual, <http://www.mysql.ru/docs/man/What-is.html>, accessed April 21, 2021.

11. How to create a collapsible tree table in html / css / js? <https://coderoad.ru/5636375/> accessed April 21, 2021.

12. Zatonsky A. V., Varlamova S. A. Information support of decision-making support on the example of scheduling classes for an educational organization // Bulletin of the South Ural State University. Series: Computer technologies, control, radioelectronics. 2018. – T. 18.No. 3. – P. 88-106.