

```
{
  "access_token": "FH6SMCPsP9VghA6TEGwedl...iDxCffeKI12VnIbUe6FEE6o1o2B5cV4TzdVvaRWSxdLagfk
kSvNQ0795FJW4iPtfaK...lc1XQ",
  "expires_in": 3599,
  "refresh_token": "...wPYV6mR0X7K0Zomh-zfSUIhD1f...97M",
  "scope": "https://www.googleapis.com/auth/adsense.readonly",
  "token_type": "Bearer"
}[root@zb scripts]#
```

На днях размотал запутанную головоломку, поэтому ее надо обязательно описать, иначе сам забуду. Я настроил доступ к google api для того, чтобы забирать данные в Zabbix. Покажу это на примере информации из личного кабинета AdSense. Забирать инфу через api будем обычными post и get запросами через curl.

Теоретический курс по основам **сетевых технологий**. Позволит системным администраторам упорядочить и восполнить пробелы в знаниях. Цена очень доступная, есть бесплатный доступ. Все подробности по . Можно пройти тест на знание сетей, бесплатно и без регистрации.

Содержание

Введение

Добавление ресурсов в консоли API

Доступ к Google API

Сбор данных в Zabbix из Google API

Заключение

Помогла статья? Подписывайся на telegram канал автора

Введение

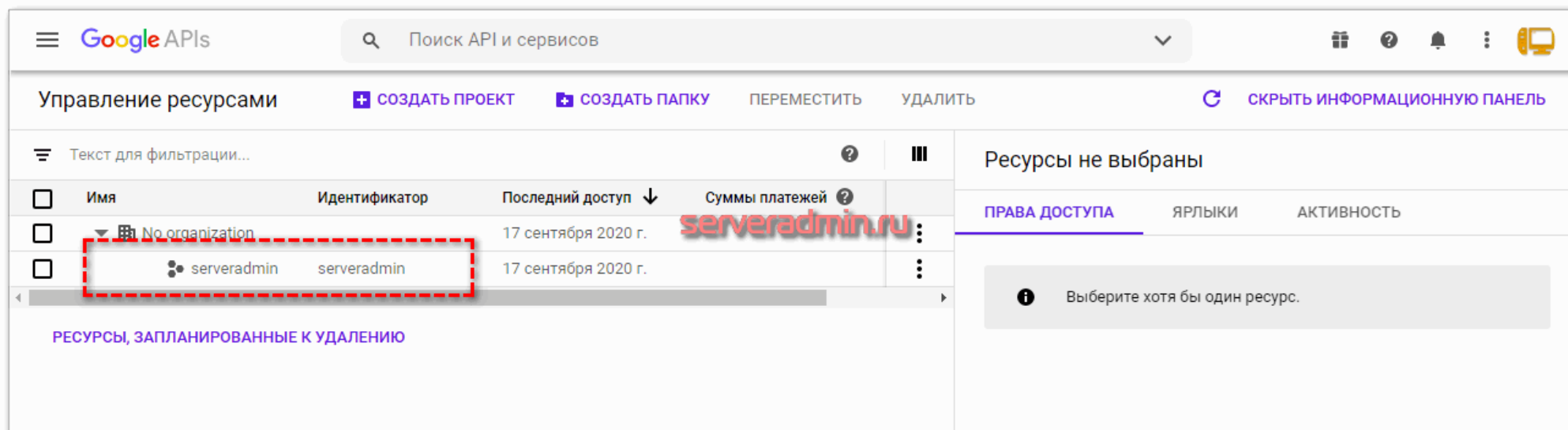
Для начала расскажу, для чего я все буду делать. Глобальная задача стоит сделать дашборд для работы маркетолога или ssm специалиста. Свести в одно место метрики из разных систем - посещаемость сайтов, монетизация, активность в группах vk, telegram и т.д. Часть этого всего я уже сделал. Буду постепенно делиться на страницах сайта. Начать рассказ решил с самого сложного - работа с **google api**.

В целом, работа со всеми api примерно одинаковая и не представляет особой сложности. С google api возникают проблемы с доступом у тех, кто первый раз все это настраивает. Там слишком много всяких сущностей. Надо пройти определенный квест и регулярно его повторять, чтобы иметь доступ к google api. Сделано все это для безопасной работы больших и сложных систем. Ну а обычным юзерам, которые просто хотят дергать curl-ом какие-то метрики, это кажется не очень простым.

В общем, расскажу обо всем по порядку, чтобы было понятно даже новичку, который только начинает работу с api. Покажу на реальном примере моего сайта и кабинета adsense к нему.

Добавление ресурсов в консоли API

Вся настройка начинается с работы в Консоли API. Вам необходимо создать новый проект.



<input type="checkbox"/>	Имя	Идентификатор	Последний доступ ↓	Суммы платежей ?	
<input type="checkbox"/>	No organization		17 сентября 2020 г.		⋮
<input type="checkbox"/>	serveradmin	serveradmin	17 сентября 2020 г.		⋮

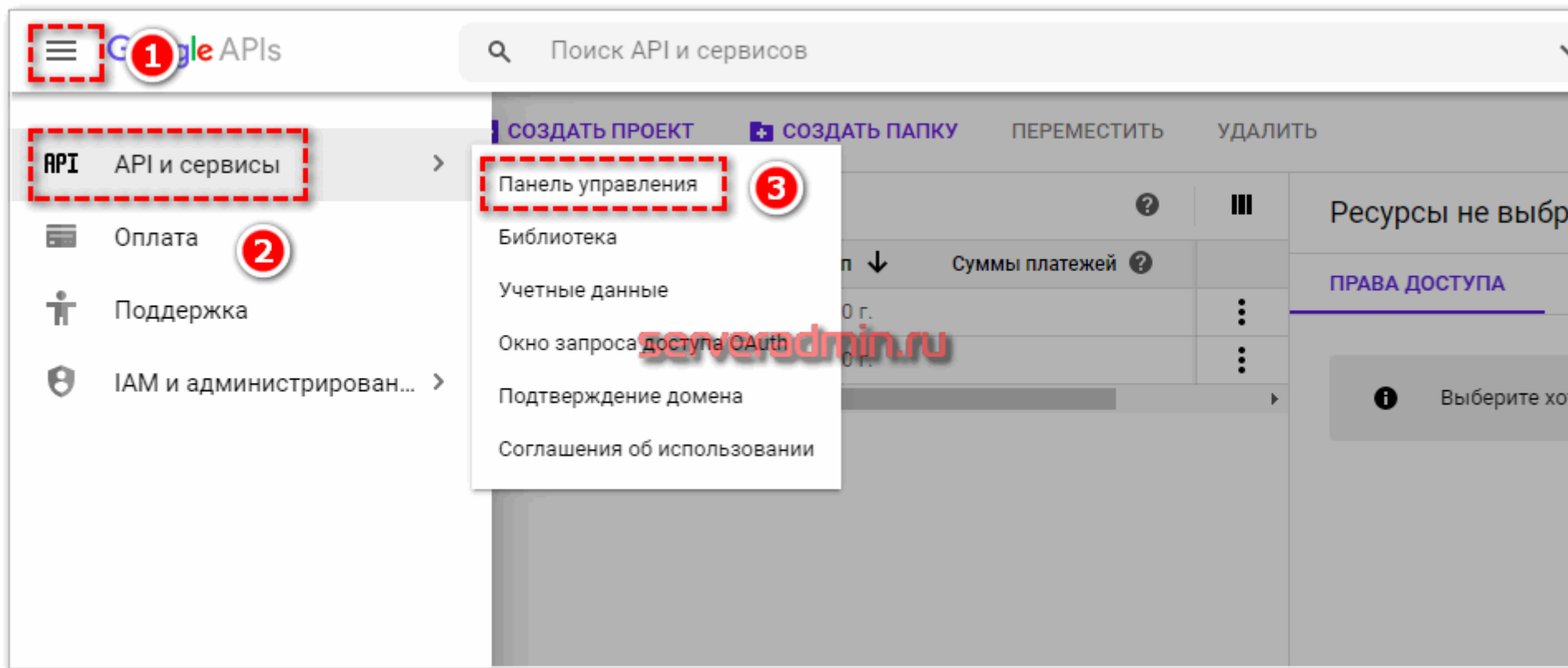
РЕСУРСЫ, ЗАПЛАНИРОВАННЫЕ К УДАЛЕНИЮ

Ресурсы не выбраны

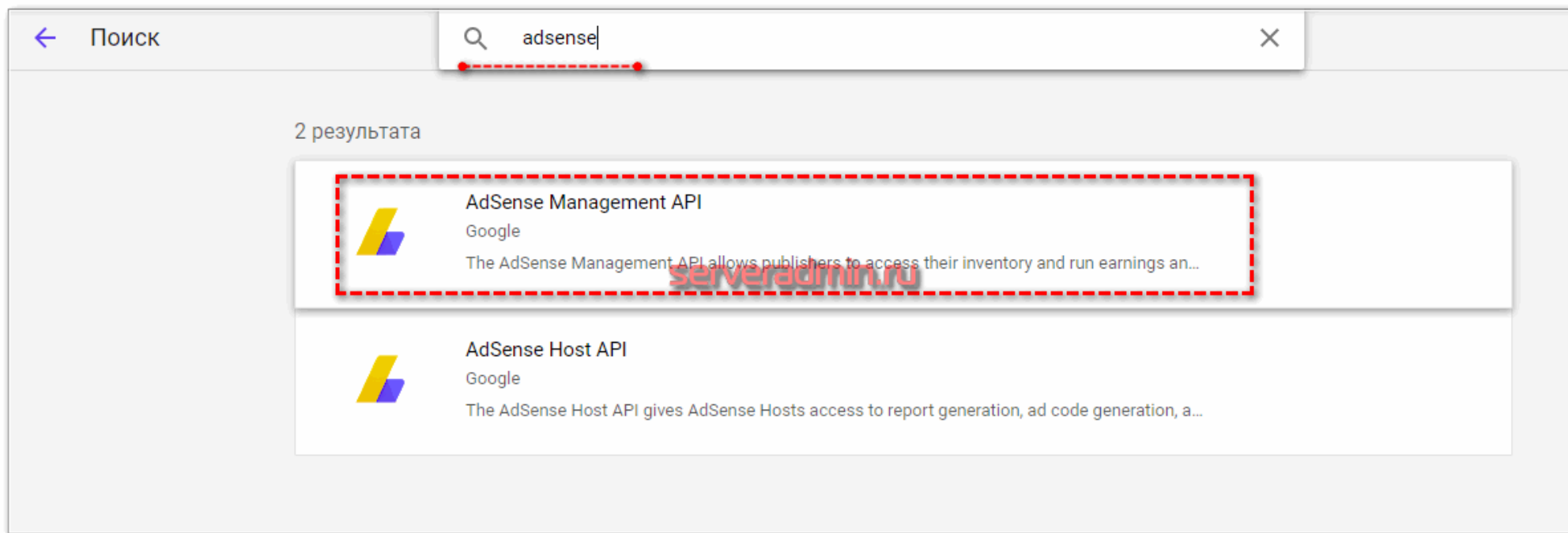
ПРАВА ДОСТУПА | ЯРЛЫКИ | АКТИВНОСТЬ

Выберите хотя бы один ресурс.

Я создал новый проект и назвал его serveradmin. После этого в верхнем левом углу выбирайте меню навигации и переходите в **API и сервисы**.



Далее нажимайте сверху на кнопку [ENABLE APIS AND SERVICES](#). В поиске набирайте adsesne и переходите в **AdSense Management API**.



На следующей странице нажимайте **ВКЛЮЧИТЬ**.

Важно!!! У вас должен быть выключен блокировщик рекламы в браузере. Если он будет включен, то получите ошибку и API adsense не заработает. Тут и далее обязательно все делайте без блокировщика.

API мы запустили, теперь нужно создать для него ключ. С помощью ключа можно обращаться к конкретному API, к которому он привязан. Здесь ожидаешь, что это будет некий токен, с помощью которого можно будет уже получать данные, но нет. Нужно четко понимать терминологию и логику работы google api. Сейчас мы создаем именно идентификатор, который будет привязан к api adsense, чтобы по нему обращаться только конкретно к этим данным.

Итак, идем в раздел **Учетные данные** и создаем новые. Выбираем Ключ API.

API и сервисы

Учетные данные

+ СОЗДАТЬ УЧЕТНЫЕ ДАННЫЕ

Удалить

Ключ API
Используется для доступа и проверки квоты.

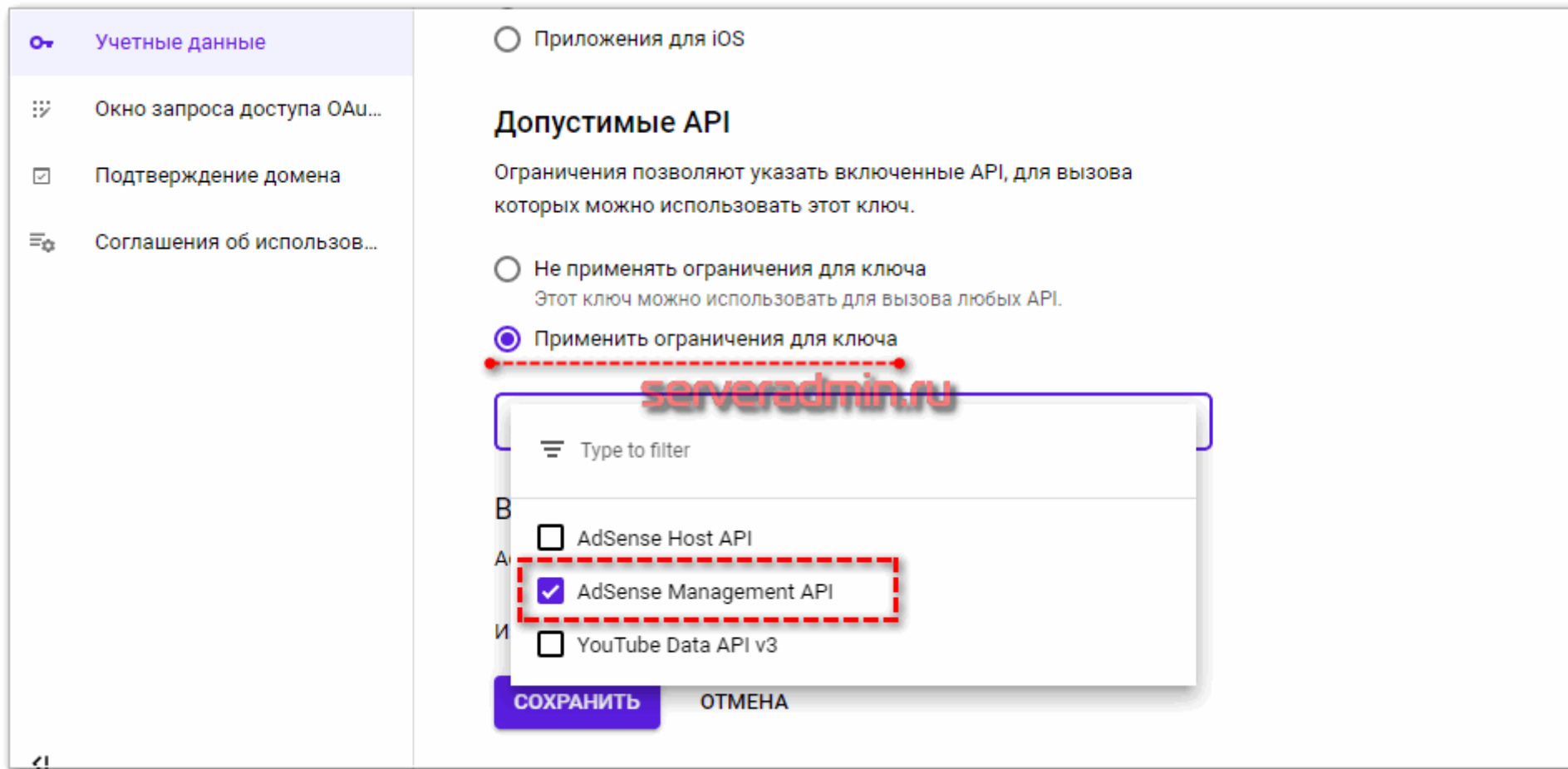
Идентификатор клиента OAuth
Позволяет запросить согласие пользователя на доступ приложения к его данным.

Сервисный аккаунт
Используется аккаунтами роботов для межсерверной аутентификации на уровне приложения.

Мастер создания учетных данных
Ответьте на несколько вопросов, и мы поможем вам определить, какой тип учетных данных нужно использовать.

Имя	Дата создания	Использование всех сервисов за последние 30 дней
	17 сен	527

В настройках ключа можно настроить ограничение доступа на уровне IP или доменного имени сайта, с которого будут идти запросы. Если у вас есть возможность ограничить доступ, рекомендую это сделать сразу же. Я обычно по IP делаю ограничение, так как данные собираю только с конкретного сервера. В списке допустимых API укажите только AdSense Management API, который мы добавили ранее. Для каждой службы лучше делать свой ключ.



Самое простое сделали. Активировали API и добавили ключ, по которому мы будем к нему обращаться. Теперь надо настроить доступ к этому API.

Доступ к Google API

Я не буду подробно описывать теорию работы с google api через OAuth 2.0. Вы можете без проблем нагуглить эту информацию. Опишу своими словами, как это работает.

1. Сначала нужно создать идентификатор клиента OAuth 2.0, который будет состоять из **client_id** и **client_secret**.
2. Далее для этого идентификатора нужно будет запросить **Authentication code**, указав **scope**, на который он будет распространяться. В нашем случае scope для adsense - <https://www.googleapis.com/auth/adsense.readonly>, так как я собираюсь только читать данные через api, не менять их.
3. С помощью полученного кода мы делаем запрос токена. В ответ получаем **access_token** и **refresh_token**. С помощью первого токена мы можем делать запросы на получение данных, с помощью второго - обновлять первый. Проблема в том, что access_token действует только 60 минут. После этого его надо обновлять с помощью refresh_token, у которого нет ограничения по времени жизни.

Вот такая схема доступа к данным через google api. Я не буду часто обращаться за данными, поэтому решил в каждом запросе к api сначала обновлять access_token с помощью refresh_token и подставлять новый токен в запрос. Приступаем к реализации.

Идем снова в раздел *Учетные данные* и добавляем новые, только теперь указываем **Идентификатор клиента OAuth**.

API и сервисы

Учетные данные

+ СОЗДАТЬ УЧЕТНЫЕ ДАННЫЕ

удалить






Ключ API
Используется для доступа и проверки квоты.

Идентификатор клиента OAuth
Позволяет запросить согласие пользователя на доступ приложения к его данным.

Мастер создания учетных данных
Ответьте на несколько вопросов, и мы поможем вам определить, какой тип учетных данных нужно использовать.

Имя	Дата создания	Использование
	17 сен	Использование всех сервисов за последние 30 д
	7 авг. 2019 г.	533

Вам будет предложено создать **Окно запроса доступа OAuth**. Сделайте это, указав любое название и выбрав тип - **Внешнее**. Все остальные поля заполнять не обязательно. После этого снова возвращайтесь к созданию идентификатора oauth. В типе приложения указывайте *Приложение для ПК*. Именно этот тип подходит для запросов через curl.

API API и сервисы	← Создание идентификатора клиента OAuth
 Панель управления	<p>Идентификатор клиента используется для идентификации приложения на серверах Google OAuth. Если приложение запущено на нескольких платформах, то у него должен быть отдельный идентификатор клиента для каждой из них. Подробнее о настройке OAuth 2.0...</p> <p>Тип приложения * Приложение для ПК</p> <p>Подробнее о типах клиентов OAuth...</p> <p>Имя * Клиент для компьютеров 1</p> <p>Название клиента OAuth 2.0. Оно используется только для идентификации клиента в консоли. Конечные пользователи его не увидят.</p> <p>СОЗДАТЬ ОТМЕНА</p>
 Библиотека	
 Учетные данные	
 Окно запроса доступа OAuth...	
<input checked="" type="checkbox"/> Подтверждение домена	
 Соглашения об использо...	

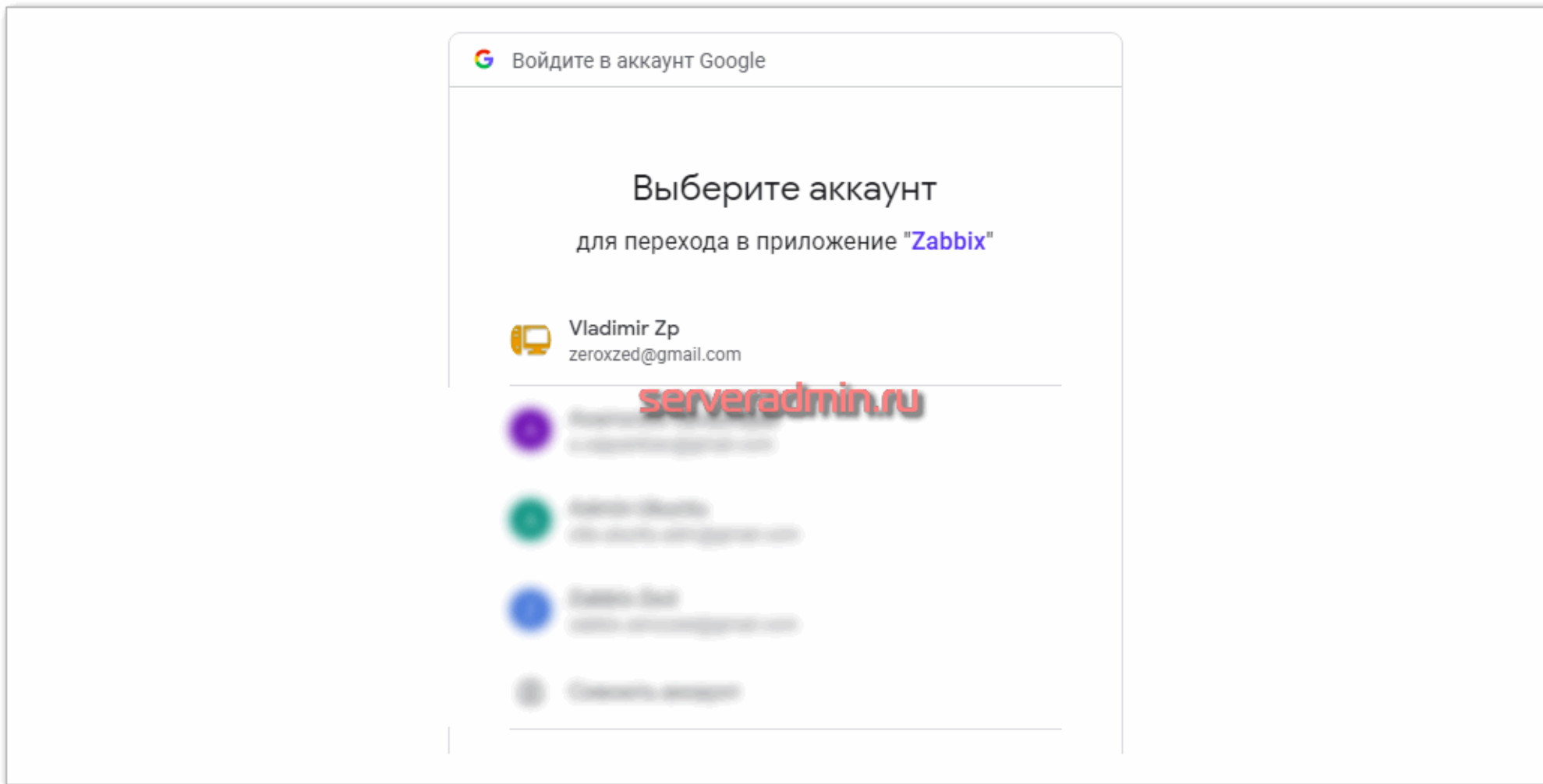
У вас теперь есть Идентификатор клиента (client_id) и Секретный код клиента (client_secret). Далее нам надо получить Authentication code. Для этого переходите по ссылке:

```
https://accounts.google.com/o/oauth2/auth?client_id=[client_id]&redirect_uri=urn:ietf:wg:oauth:2.0:oob&scope=[scope]&response_type=code.
```

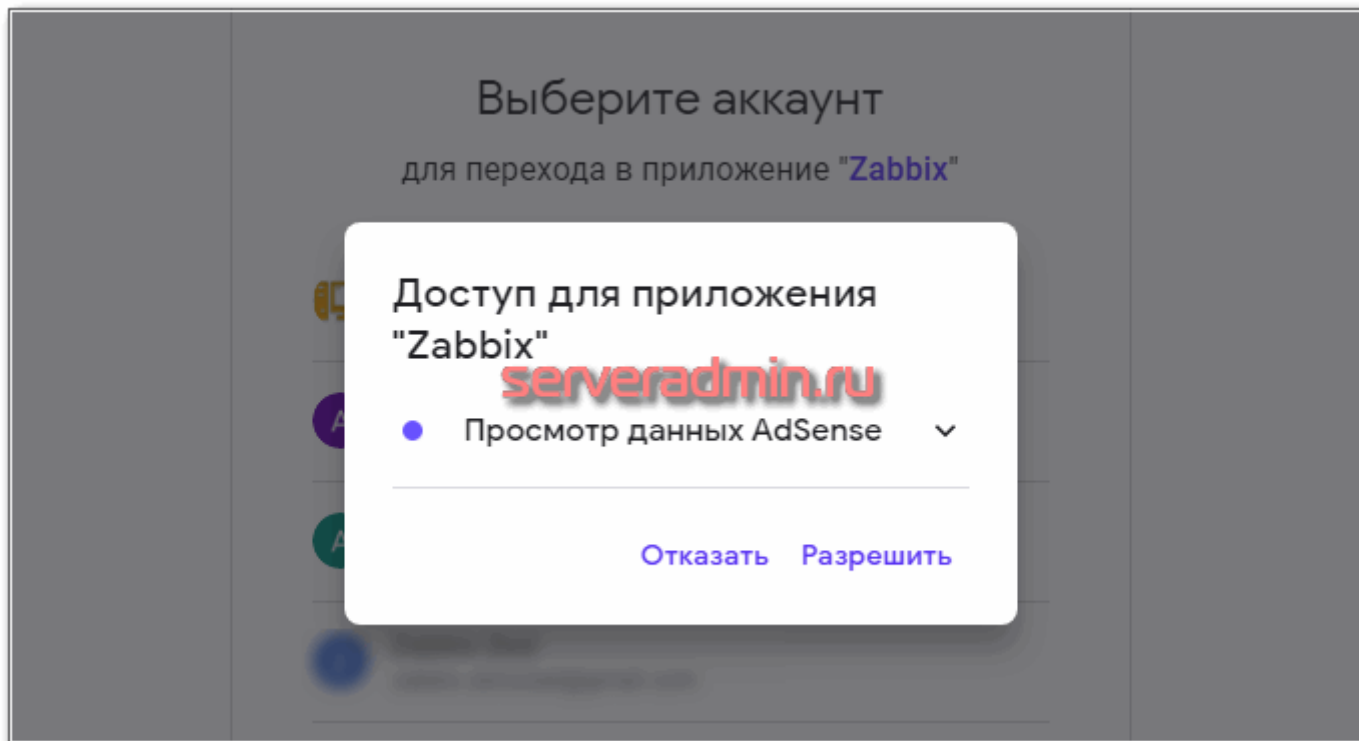
В моем случае ссылка получается такая:

```
https://accounts.google.com/o/oauth2/auth?client_id=782459127764-gu9rr9fo379klsqvmk8b6ubgd6kn.apps.googleusercontent.com&redirect_uri=urn:ietf:wg:oauth:2.0:oob&scope=https://www.googleapis.com/auth/adsense.readonly&response_type=code
```

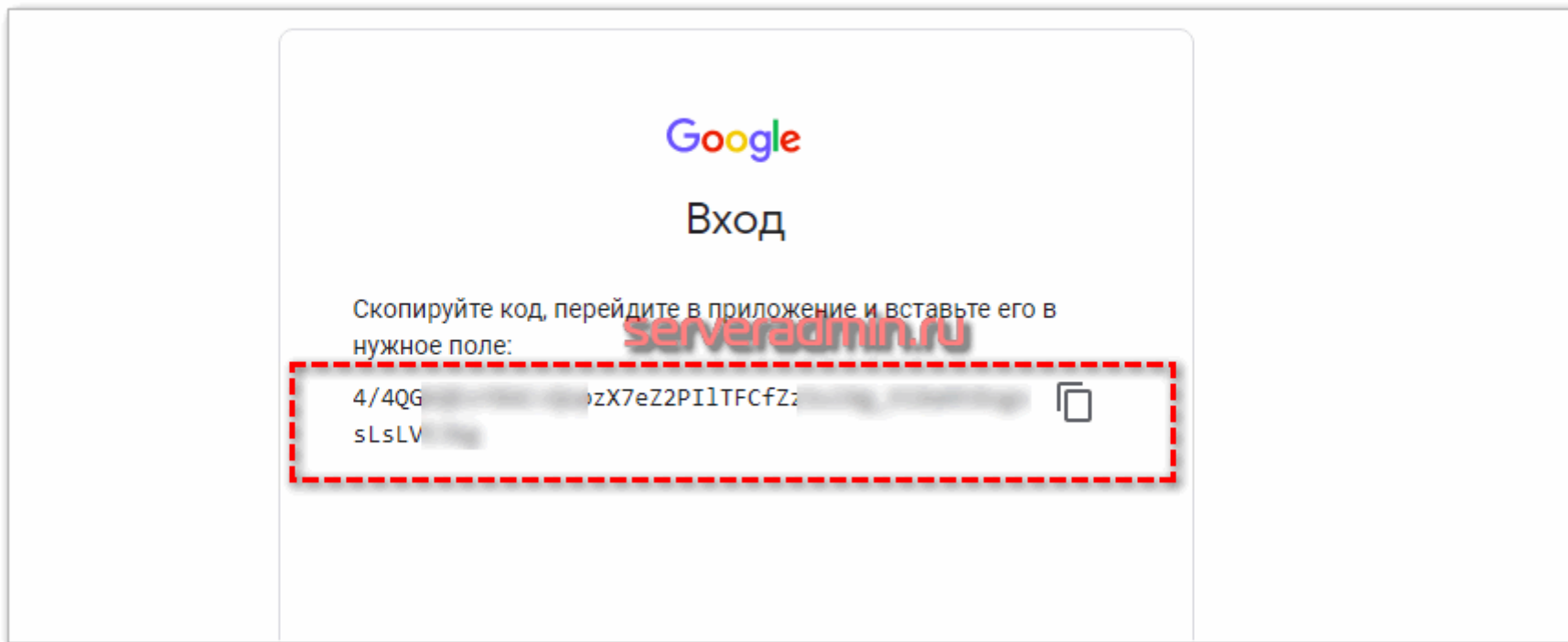
После перехода вас попросят авторизоваться под учетной записью google.



И далее спросят, согласны ли вы ей предоставить доступ к просмотру данных adsense.



После вашего согласия получите необходимый код.



Запомните его, он нам пригодится далее. С его помощью, а так же с помощью client_id и client_secret формируем POST запрос на получение токена.

```
curl --request POST --data \  
"code=[Authentication code]&client_id=[client_id]&client_secret=[client_secret]&grant_type=authorization_code" \  
https://accounts.google.com/o/oauth2/token
```

В моем случае он получился такой:

```
curl --request POST --data \  
"code=4/4QGbQEsY8QwTpEFBeZ2PIlTFCfZzSwlXgwyPrTVEJkg&client_id=782459127764-  
gu9rr9fo379klsqvmk8b6ubgd6kn.apps.googleusercontent.com&client_secret=l7_-  
a3RTnmPkDVL5DwruJfix&redirect_uri=urn:ietf:wg:oauth:2.0:oob&grant_type=authorization_code" \  
https://accounts.google.com/o/oauth2/token
```

После выполнения запроса, вам вернется json с access_token и refresh_token. Это то, что нам нужно.

```
{  
  "access_token": "FH6SMCPsP9VghA6TEGwedl iDxCffeKI12VNbUe6FEE6o1o2B5cV4TzdVvaRWSxdLagfk  
kSvNQ0795FJW4iPtfaK lc1XQ",  
  "expires_in": 3599,  
  "refresh_token": "wPYVnmr0XC7KXIZomh-zfsUihD1f 97M",  
  "scope": "https://www.googleapis.com/auth/adsense.readonly",  
  "token_type": "Bearer"  
}[root@zb scripts]#
```

Теперь у нас есть все для того, чтобы получать данные из кабинета adsense. Вот пример запроса на получение дохода за сегодняшний день.

```
curl \  
'https://www.googleapis.com/adsense/v1.4/reports?startDate=today&endDate=today&accountId=[account  
id]&metric=EARNINGS&useTimezoneReporting=true&key=[api key]' \  
--header 'Authorization: Bearer [access_token]' \  
--header 'Accept: application/json' \  
--compressed
```

В моем случае запрос получился такой.

```
curl
'https://www.googleapis.com/adsense/v1.4/reports?startDate=today&endDate=today&accountId=pub-1320661661569887&metric=EARNINGS&useTimezoneReporting=true&key=AIzaDeTPNM2w6q0PM068vfEH5XTjEsdPEf7K9lpVc' \
--header 'Authorization: Bearer
ya29.a0AfH6l8BRwADYPnpjAFpBEfUykAdMyKXxpnSWaChUBrKJJJDW0PLNTREWfdgfdgcGwPr3UJPOH0tp5VYgT6A0kjVwDxCffeKI12VNBue6FagfkkSvNNQ
0795FJW4iPtfaKFFRcdKailc1XQ' \
--header 'Accept: application/json' \
--compressed
```

Ну и ответ:

```
[
  {
    "kind": "adsense#report",
    "totalMatchedRows": "1",
    "headers": [
      {
        "name": "EARNINGS",
        "type": "METRIC_CURRENCY",
        "currency": "USD"
      }
    ],
    "rows": [
      [
        "9.31"
      ]
    ],
    "totals": [
      "9.31"
    ],
  ],
]
```



```
"averages": [  
  "9.31"  
],  
"startDate": "2020-09-19",  
"endDate": "2020-09-19"  
}  
]
```

Вот и все. Доступ к данным вы получили. Теперь надо научиться обновлять токен, так как этот протухнет через час. Обновляем его с помощью следующего запроса.

```
curl --request POST \  
--data "client_id=[client_id]&client_secret=[client_secret]&refresh_token=[refresh_token]&grant_type=refresh_token" \  
https://accounts.google.com/o/oauth2/token
```

На выходе получите новый токен для доступа к api.

В целом, разобрали механизм доступа и получения данных из google api. Дальше передать это все в Zabbix дело техники. Я уже не стал там сильно заморачиваться и сделал все в лоб через bash скрипты, хотя есть и другие варианты. Рассказываю далее об этом.

Немного полезных ссылок по теме. Для формирования, да и просто проверки запросов к api, удобно пользоваться API Explorer. Его можно вызвать справа в сайдбаре. Слева будет документация по типам ресурсов. Мне нужен только ресурс **report** и метрика *earnings* для запроса различных доходов по датам. А так же ресурс **payments** для информации о текущем балансе аккаунта.

И еще полезная ссылка на [getting_started](#) от самого гугла. Там кратко будет описан принцип работы с google adsense api.

Сбор данных в Zabbix из Google API

Осталось самое простое - настроить сбор данных в Zabbix через Google api. Из-за того, что токен протухает и его надо постоянно обновлять, не получится собирать данные только через HTTP Агент. Я спросил в чате Zabbix, можно ли как-то реализовать сбор данных по такой схеме только в шаблоне. Мне

ответили, что можно, но схема мне не очень понравилась.

Можно создать отдельный item для хранения токена, который будет регулярно обновляться запросом с использованием временного токена. В постобработке этого итема было предложено с помощью JavaScript обновлять макрос для основного айтема, где будет храниться токен, через zabbix api.

Плюс этого метода в том, что все хранится в шаблоне, легко переносится и сохраняется. Нет необходимости что-то делать на хостах. Минус только один - если итемов и проверок много, это все нагружается сервер мониторинга, так как собирает и обрабатывает данные он полностью сам. В целом, я все стараюсь реализовать в шаблонах, так как с большими нагрузками не работаю, а удобство такого подхода налицо.

В данном случае не стал разбираться с JavaScript и Zabbix api, так как банально стало жаль время тратить. Решил отложить этот как-нибудь на потом, а все запросы реализовал по старинке на хосте через bash скрипты. Заняло это у меня буквально 10 минут. Буду передавать готовые цифры сразу в Zabbix через UserParameter.

Если у вас еще нет своего сервера для мониторинга, то рекомендую материалы на эту тему. Для тех, кто предпочитает систему CentOS:

1. Установка CentOS 8.
2. Настройка CentOS 8.
3. Установка и настройка zabbix сервера.

То же самое на Debian 10, если предпочитаете его:

1. Установка Debian 10.
2. Базовая настройка Debian.
3. Установка и настройка zabbix на debian.

Я настроил сбор следующих данных о доходе из adsense по api:

1. Сегодня.
2. Вчера.
3. В текущем месяце.
4. В прошлом месяце.
5. А так же текущий баланс аккаунта.

Отметка времени	Имя	Значение
19.09.2020 18:56:52	Доход сегодня	9.41
19.09.2020 18:47:02	Баланс	286.06
19.09.2020 18:46:59	Доход в прошлом месяце	169.39
19.09.2020 18:46:57	Доход в этом месяце	105.99
19.09.2020 18:46:55	Доход вчера	5.21

Для этого я простым копипастом сделал 5 скриптов с автообновлением токена и 5 итемов. Вот содержимое скриптов.

```
# cat adsense-day.sh
```

```
#!/bin/bash
```

```
client_id='782412765764-gu9dsfsrtyjgd6kn.apps.googleusercontent.com'
```

```
client_secret='l7_-a73uErw532oPer67riuJfix'
```

```
refresh_token='1//0ck3_00Bbay6NQm0yPE_9UGKDFghmEwPuhDDEHMWrjjlWedNceQwepfdkjEvhsadfEWQPPJfgfcmmSwqfgfh'
```

```
api='AIZAswwQQEfjlPewvBnmFDDRTsHYGGUyTjUFqEf7K9lpVc'
```

```
token=`curl -s --request POST --data
```

```
"client_id=${client_id}&client_secret=${client_secret}&refresh_token=${refresh_token}&grant_type=refresh_token"
```

```
https://accounts.google.com/o/oauth2/token | grep access_token | awk '{print $2}' | sed -e 's/^./' | sed 's/..$//'
```

```
curl -s
"https://www.googleapis.com/adsense/v1.4/reports?startDate=today&endDate=today&accountId=pub-13106391315157&metric=EARNIN
GS&useTimezoneReporting=true&key=${api}" \
--header "Authorization: Bearer ${token}" \
--header "Accept: application/json" \
--compressed
```

Я в переменную token записываю токен доступа, полученный с помощью refresh_token, обрезав все лишнее в выводе. А дальше его подставляю в запрос.

В остальных скриптах на запрос дохода меняются только даты:

- adsense-yesterday.sh - startDate=today-1d&endDate=today-1d.
- adsense-month.sh - startDate=startOfMonth&endDate=today
- adsense-lastmonth.sh - startDate=startOfMonth-1m&endDate=startOfMonth-1d

```
# cat adsense-payments.sh
```

```
#!/bin/bash
```

```
client_id='782412765764-gu9dsfsrtyjgd6kn.apps.googleusercontent.com'
client_secret='l7_-a73uErw532oPer67riuJfix'
refresh_token='1//0ck3_00Bbay6NQm0yPE_9UGKDFghmEwPuhDDEHMWrjjlWedNceQwepfdkjEvhsadfEWQPPJfgfcmmSwqfgfh'
api='AIzAswwQEfjlPewvBnmFDDRTsHYGGUyTjUFqEf7K9lpVc'

token=`curl -s --request POST --data
"client_id=${client_id}&client_secret=${client_secret}&refresh_token=${refresh_token}&grant_type=refresh_token"
https://accounts.google.com/o/oauth2/token | grep access_token | awk '{print $2}' | sed -e 's/^./' | sed 's/..$/'`

curl -s "https://www.googleapis.com/adsense/v1.4/accounts/pub-1310661631569687/payments?key=${api}" \
--header "Authorization: Bearer ${token}" \
```

```
--header "Accept: application/json" \  
--compressed
```

Тут другой url запроса, так как поменялся ресурс с reports на payments.

Добавляем UserParameter в конфигурацию zabbix агента.

```
UserParameter=adsense.day, /bin/bash /etc/zabbix/scripts/adsense-day.sh  
UserParameter=adsense.yesterday, /bin/bash /etc/zabbix/scripts/adsense-yesterday.sh  
UserParameter=adsense.month, /bin/bash /etc/zabbix/scripts/adsense-month.sh  
UserParameter=adsense.lastmonth, /bin/bash /etc/zabbix/scripts/adsense-lastmonth.sh  
UserParameter=adsense.payments, /bin/bash /etc/zabbix/scripts/adsense-payments.sh
```

Перезапустите агента и проверьте работу итемов.

```
# zabbix_agentd -t adsense.day  
adsense.day  
[t|{"kind": "adsense#report", "totalMatchedRows": "1", "headers": [{"name": "EARNINGS", "type": "METRIC_CURRENCY", "currency": "USD"}], "rows": [{"9.42"}], "totals": ["9.42"], "averages": ["9.42"], "startDate": "2020-09-19", "endDate": "2020-09-19"}]
```

На выходе должны получить json. Подробно работу с json в Zabbix я разбирал в отдельной статье. Теперь идем на Zabbix сервер и создаем шаблон. Я покажу на примере одного итема, как я предобработкой разбираю приходящий json.

Создаем новый итем.

Элемент данных **Предобработка**

* Имя

Тип

* Ключ

Тип информации

Ед. измерения

* Интервал обновления

Пользовательские интервалы

Тип	Интервал	Период	Действие
<input checked="" type="checkbox"/> Переменный	<input type="text" value="По расписанию 50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Удалить

[Добавить](#)

* Период хранения истории Не хранить историю Период хранения

* Период хранения динамики изменений Не хранить динамику изменений Период хранения

Отображение значения

Новая группа элементов данных

Группы элементов данных

- Нет-
- Adsense

И к нему Предобработку, чтобы получить только число.

Элемент данных Предобработка

Шаги предобработки	Имя	Параметры	Другое при ошибке	Действия
1:	JSONPath	<input type="text" value="\$..rows"/>	<input type="checkbox"/>	Тест Удалить
2:	Обрезать справа	<input type="text" value="serveradmin.ru"/>	<input type="checkbox"/>	Тест Удалить
3:	Обрезать слева	<input "="" type="text" value="["/>	<input type="checkbox"/>	Тест Удалить

[Добавить](#)

Для построения JSONPath рекомендую вот этот онлайн конструктор - <http://jsonpath.com>. В данном случае не очень актуально, так как jsonpath простой. Но в сложных ситуациях сильно выручает. Выгрузил вам для примера мой шаблон с этими пятью айтемами - [zabbix-adsense.xml](#).

Там ничего особенного нет, просто 5 айтемов с настроенными интервалами, чтобы не дергать лишней раз api. Доход прошлого месяца запрашиваем 2-го числа нового месяца. Вчерашний доход забираем сегодня в 17.00 и т.д. Посмотрите сами и настройте так, как вам нужно.

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

Я показал общий принцип работы с google api в Zabbix. Сам ещё подробно не разобрался и дашборды не составлял. Остановился ровно на том, на чем

закончил статью. Дальше буду все это разрабатывать и настраивать под конкретные задачи и нужды. Потом все это уедет вместе с остальными метриками в grafana для удобного дашборда. Возможно покажу все это уже в готовом виде.

Для более сложных и разнообразных запросов надо сделать один скрипт с передачей параметров даты, чтобы не плодить скрипты под каждый айтем. Хотелось бы получить какую-то обратную связь от тех, кто всем этим уже занимался. Тема вообще не гуглится, разобрался во всем сам. Думаю, что в основном для работы с api используются готовые библиотеки под различные языки программирования. Прямые запросы через curl больше на экзотику тянет.

И еще у меня есть некоторые подозрения, что я пошел по сложному пути, и есть способ доступа к api попроще, без протухающих токенов. Но я не понял, как сделать по-другому.

Онлайн курс "DevOps практики и инструменты"

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, научиться непрерывной поставке ПО, мониторингу и логированию web приложений, рекомендую познакомиться с **онлайн-курсом «DevOps практики и инструменты»** в OTUS. Курс не для новичков, для поступления нужны базовые знания по сетям и установке Linux на виртуалку. Обучение длится 5 месяцев, после чего успешные выпускники курса смогут пройти собеседования у партнеров. Проверьте себя на вступительном тесте и смотрите программу подробнее по .

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.