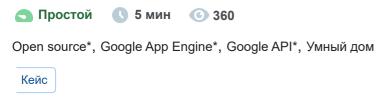
КАК СТАТЬ АВТОРОМ



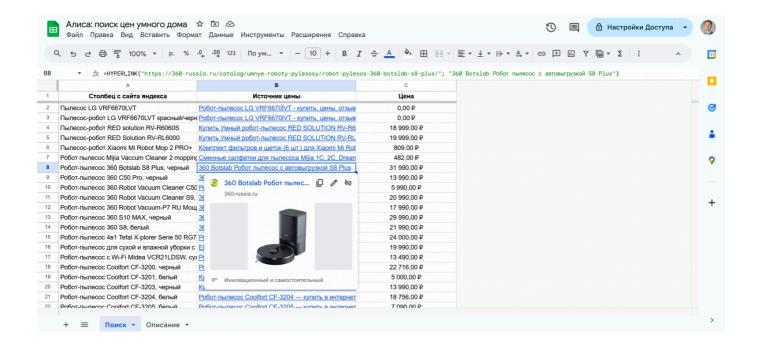
# Как быстро найти цены на совместимую с Home Assistant бытовую технику: мой опыт автоматизации поиска цен



Подбор бытовой техники для своего умного дома может быть сложной задачей. Даже если техника имеет "умную часть", не факт что она подойдёт именно к вашей экосистеме. Поэтому вопрос совместимости лучше выяснить ещё до покупки устройства.

Недавно захотел подобрать себе робот-пылесос, а на сайте Home Assistant как обычно указаны только обобщенные интеграции. При этом на сайте Алисы приведены тысячи конкретных совместимых устройств, правда без цены и даже без указания выпускается ли до сих пор эта модель или нет.

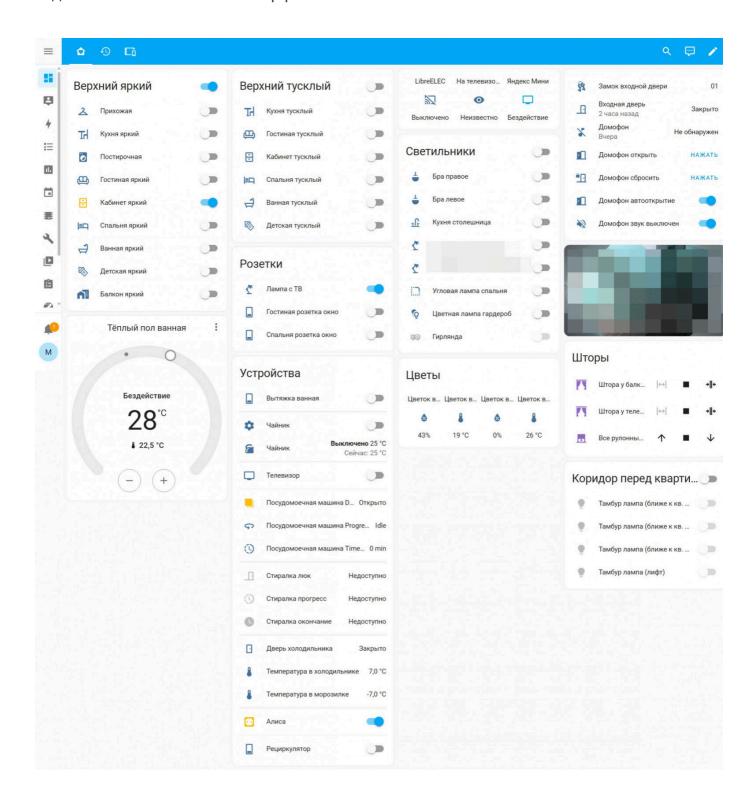
И тут мне пришла в голову идея: поместить эти совместимые устройства в Google Таблицу, а затем написать Google Apps Script, который через поиск найти цены на все эти устройства, а затем я уже выберу конкретное устройство исходя из моего бюджета.



# Проблема совместимости устройств

Разные платформы: от Home Assistant до Яндекс Алисы - используют собственные методы подключения. Одни устройства работают через Wi-Fi, другие через Zigbee или Bluetooth, а

некоторые требуют облачных сервисов. В результате поиск подходящей техники превращается в квест. Я сам использую локальный сервер Home Assistant, а Яндекс Алиса подключена как голосовой интерфейс.



#### Home Assistant: свобода выбора или сложность интеграции?

Home Assistant - мощная платформа с гибкими возможностями, но её открытая экосистема накладывает ограничения. Производители редко заявляют официальную поддержку НА, поэтому пользователям приходится самостоятельно искать способы интеграции. На сайте Home Assistant есть каталог интеграций, но он не содержит конкретных моделей техники — только названия брендов и технологий.

Дополнительно усложняет выбор большое количество разных протоколов:

- Wi-Fi наиболее распространённый вариант, но часто требует облака
- Bluetooth подходит для локальных устройств, но ограничен дальностью
- Zigbee и Z-Wave энергоэффективные протоколы для умного дома, но требуют шлюза
- Локальные АРІ идеальный вариант, но встречается редко

#### Как Яндекс Алиса помогает в поиске, но не решает проблему

В отличие от Home Assistant, у Яндекс Алисы есть удобный каталог совместимых устройств с конкретными моделями. Но даже если устройство есть в списке Алисы, это не гарантирует его работу в НА. Многие устройства взаимодействуют с Алисой через облако, а Home Assistant требует локального API или поддержки Zigbee.

#### Автоматизация поиска цены

Как уже говорил каталог Яндекс Алисы содержат сотни совместимых моделей, но без цен и актуальности их наличия. Проверять каждую вручную на маркетплейсах — долго и неудобно, особенно когда выбор включает десятки вариантов.

#### Google Apps Script для автоматического поиска цен на устройства

Чтобы не тратить время на ручной поиск, я написал скрипт на Google Apps Script. Он использует Google Custom Search Engine (CSE) для поиска цен в результатах выдачи. Это позволяет быстро находить магазины, где продаётся конкретная модель, и загружать ссылки с актуальными ценами прямо в таблицу.

```
Начать развертывание 🔻
🄐 Apps Script 🛮 Алиса: поиск цен умного дома 🔗
                               ÂZ + 5 € Выполнить БО Отладка aliceSmartHomePriceSearch ▼ Журнал выполнения
         Файлы
(i)
                                                         // Добавляем уточняющие слова и кавычки для точного совпадения модели var cleanProductName = productName .replace(/\s+/g, ' ')
<> Koд.gs
                                                          .replace(/\s+/g,
.trim();
Библиотеки
                                    + 46 47
Сервисы
                                                          // Формируем поисковый запрос с исключениями и уточне
                                                       // Формируем поиско
var searchTerms = [
                                                          secrets.initial_phrase
`${cleanProductName}`,
excludeString
₽,
(3)
                                                         ].filter(Boolean).join(' ');
                                                        var query = encodeURIComponent(searchTerms);
                                                          var url = 'https://www.googleapis.com/customsearch/v1?q=' + query + 
| '&cr=countryRU&cx=' + cx + '&key=' + apiKey;
                                                                                                 ${searchTerms}`);
                                                        Logger.log(`URL sampoca: ${url}`);
                                                         var response = UrlFetchApp.fetch(url);
var json = JSON.parse(response.getContentText());
                                                         if (json.items && json.items.length > 0) {
  for (var i = 0; i < json.items.length; i++) {</pre>
                                                               var titleLower = joon.items[i].title.toLowerCase();
var hasExcludedTerms = secrets.exclude_terms.some(term =>
    titleLower.includes(term.toLowerCase())
```

## Описание работы скрипта: использование Google Custom Search Engine

Скрипт получает список устройств из таблицы и для каждого отправляет запрос в Google CSE. Если найдено предложение с ценой, информация записывается в таблицу: ссылка и стоимость. Если подходящего предложения нет, выводится сообщение "Цена не найдена" и ссылка на поиск в Яндексе.

Ссылка на таблицу: Хабр Алиса: поиск цен умного дома

Скрипт:

```
const secrets = {
 apiKey_token: 'xxx', // Ключ № 1 на 100 запросов в день из xxx@gmail.com
 // apiKey_token: 'xxx', // Ключ № 2 на 100 запросов в день из xxx@gmail.com
 cx_token: 'xxx', // Замените идентификатором Google's Custom Search Engine (CSE) ID
  // https://cse.google.com/cse?cx=xxx&cr=countryRU
 initial_phrase: '',
 // Стоп-слова для исключения из поиска
 exclude_terms: []
};
// Подробнее про автоматический поиск цен на товары: https://habr.com/ru/articles/87957
function aliceSmartHomePriceSearch() {
 var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Πομςκ');
 var data = sheet.getDataRange().getValues();
  var results = [];
  for (var i = 1; i < data.length; i++) { // Пропускаем заголовок
   var productName = data[i][0]; // Колонка A
    if (productName) {
      console.log(`Обрабатываем строку ${i + 1}: ${productName}`);
      var result = searchProductPrice(productName);
     var hyperlink = '=HYPERLINK("' + result.link + '"; "' + result.title + '")';
      results.push([hyperlink, result.price]);
    } else {
      results.push(['', '']); // Пустые значения, если товара нет
   }
  }
  // Записываем результаты в столбцы В и С (2 и 3)
  var range = sheet.getRange(2, 2, results.length, 2);
```

```
range.setValues(results);
}
function searchProductPrice(productName) {
 var apiKey = secrets.apiKey_token;
 var cx = secrets.cx token;
  // Формируем исключающие термины
  var excludeString = secrets.exclude terms.map(term => '-' + term).join(' ');
  // Добавляем уточняющие слова и кавычки для точного совпадения модели
  var cleanProductName = productName
    .replace(/\s+/g, ' ')
    .trim();
 // Формируем поисковый запрос с исключениями и уточнениями
 var searchTerms = [
   secrets.initial_phrase,
   `${cleanProductName}`,
   excludeString
  ].filter(Boolean).join(' ');
  var query = encodeURIComponent(searchTerms);
  var url = 'https://www.googleapis.com/customsearch/v1?q=' + query +
           '&cr=countryRU&cx=' + cx + '&key=' + apiKey;
  Logger.log(`Поисковый запрос: ${searchTerms}`);
  Logger.log(`URL запроса: ${url}`);
 var response = UrlFetchApp.fetch(url);
  var json = JSON.parse(response.getContentText());
  if (json.items && json.items.length > 0) {
   for (var i = 0; i < json.items.length; i++) {</pre>
      // Проверяем, не содержит ли заголовок стоп-слова
     var titleLower = json.items[i].title.toLowerCase();
      var hasExcludedTerms = secrets.exclude terms.some(term =>
        titleLower.includes(term.toLowerCase())
      );
      if (hasExcludedTerms) {
        Logger.log(`Пропускаем результат с запрещенными словами: ${json.items[i].title]
       continue;
      }
      if (json.items[i].pagemap &&
```

```
json.items[i].pagemap.offer &&
          json.items[i].pagemap.offer[0].price) {
        var price = json.items[i].pagemap.offer[0].price;
        price = price.replace('.', ',');
        var link = json.items[i].link;
        var title = json.items[i].title;
        Logger.log(`Найдена подходящая цена: ${price} (${title})`);
        return {
          price: price,
          link: link,
          title: title
        };
      }
    }
  }
  Logger.log(`Подходящая цена не найдена для ${productName}`);
  return {
    price: '0',
   link: 'https://ya.ru/search/?text=' + query,
   title: 'Цена не найдена'
  };
}
// Тестовая функция
function testImprovedSearch() {
  var result = searchProductPrice("Робот пылесос Mijia Vaccum Cleaner 2 mopping 2 in 1
  Logger.log(result);
}
```

## Процесс работы:

- 1. Копируем список совместимых устройств в Google Таблицу, очищаем от штрих-кодов и лишних данных простой сортировкой.
- 2. Запускаем скрипт, который автоматически ищет цены.
- 3. Получаем в таблице ссылки на магазины и актуальные цены, что упрощает выбор и покупку.

Такой подход позволил мне оперативно находить нужные устройства без утомительного

мониторинга цен вручную. Теперь выбор техники для Home Assistant стал намного проще.

Как я использую этот инструмент для выбора техники

Google Apps Script значительно ускоряет поиск цен, но не всегда выдаёт идеальные

результаты. Иногда вместо нужной модели находятся запчасти или нулевые цены, потому что это устройство уже снято с производства. Например, при поиске Xiaomi Mi Robot Mop 2

PRO+ можно наткнуться на сменные щётки и фильтры вместо самого пылесоса.

Чтобы улучшить точность, я экспериментировал с формулировкой запросов. Хорошие

примеры:

• "Xiaomi Mi Robot Mop 2 PRO+ купить" – добавление слова «купить» повышает шанс

найти магазины.

• "Xiaomi Mi Robot Mop 2 PRO+ цена Пермь" – указывает город, если нужен локальный

поиск.

• Исключение лишнего: «-фильтр -щётка -запчасть» убирает ненужные товары.

Также можно расширить функционал, например, фильтровать по проверенным магазинам.

Для этого достаточно добавить названия крупных маркетплейсов в запрос.

Заключение

Автоматизированный поиск цен на совместимую с Home Assistant бытовую технику с

помощью Google Apps Script значительно экономит время и упрощает выбор. Теперь вы можете быстро получить актуальные цены на нужные модели, не тратя часы на рутинный

поиск.

Автор: Михаил Шардин

Моя онлайн-визитка

➡ Telegram «Умный Дом Инвестора»

18 марта 2025 г.

Теги: поиск цен

Хабы: Open source, Google App Engine, Google API, Умный дом

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

X

## Электропочта



183 90.7

Карма Рейтинг

# Михаил Шардин @empenoso

Автоматизация / Данные / Финансы / Умные дома





Сайт Сайт Github

