Государственное автономное образовательное учреждение

высшего образования Ленинградской области **ЛЕНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ А. С. ПУШКИНА**

Институт экономической безопасности

КУРСОВАЯ РАБОТА

на тему

«Работа внешнего API и создание асинхронного доступа

к API на примере Makeup API».

Модуль (код и наименование):

ПМ.02 Разработка, внедрение и адаптация программного обеспечения отраслевой направленности

Выполнил студент: Устинский Никита

Специальность: 09.02.05

«Прикладная информатика»

Номер зачётной книжки:

Курс 3 Группа С7120 Б

Преподаватель: Пискаев Д.Е.

Оценка:\_\_\_\_\_\_\_\_

Дата\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2023

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc135239717)

[ВВЕДЕНИЕ 3](#_Toc135239718)

[ПЕРВЫЙ РАЗДЕЛ. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБАТЫВАЕМОЙ ТЕМЫ 5](#_Toc135239719)

[1. API или программный интерфейс приложения 5](#_Toc135239720)

[1.1 Типы API 6](#_Toc135239721)

[1.2 Классификация API по сценариям использования 6](#_Toc135239722)

[1.3 Протоколы API 8](#_Toc135239723)

[1.4 Makeup API 10](#_Toc135239724)

[ВТОРОЙ РАЗДЕЛ. ПРАКТИЧЕСКИЕ РЕЗУЛЬТАТЫ РАБОТЫ 12](#_Toc135239725)

[1. Описание 12](#_Toc135239726)

[2. Скриншоты результатов в браузере и консоли 12](#_Toc135239727)

[3. Код скрипта с комментариями 15](#_Toc135239728)

[ЗАКЛЮЧЕНИЕ 16](#_Toc135239729)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ 17](#_Toc135239730)

[Приложение 1 18](#_Toc135239731)

# ВВЕДЕНИЕ

На сегодняшний день мы постоянно взаимодействуем с чем-либо как с людьми, так и с окружающими нас предметами. В век технологий, который с каждым годом набирает все большую популярность, наверное, самые длительные «отношения» у людей происходят с различного рода интернет-ресурсами. Конечно, не каждый понимает, что за тем, что мы видим, стоит чей-то колоссальный труд, что были люди, которые позаботились не только о себе и научили наши устройства общаться с нами. Вне наших глаз между программами, приложениями, веб-сервисами происходит непрерывное общение. И такое общение происходит на языке API.

На сегодняшний день API встречаются везде и уже стали привычной частью современного мира технологий, бизнеса в партнерском маркетинге и тем, без чего нам на данный момент не обойтись. Что такое API? Аббревиатура расшифровывается как Application Programming Interface, или интерфейс для программирования приложений.

Makeup API — это база косметических изделий, с помощью которой исследовать рынок косметических изделий по определенным параметрам. Makeup API использует протокол HTTP. Этот API соответствует принципам REST. Конечные точки API: конечная точка для текущей версии API: <http://makeup-api.herokuapp.com/api/v1/products.json>. В настоящее время существует один вызов API: поиск продуктов для макияжа. Тип: GET. Формат ответа: json.

В рамках курсовой работы будет создана функциональность обработки базы косметических изделий. Выбрав в инпуте бренд и тип продукта, нажать на кнопку и при выполнении кода программы в консоль будет выведен результат.

*Цель нашей работы* — создание функционального массива объектов базы косметических изделий.

Для достижения цели, нам необходимо выполнить следующие *задачи*:

1. На базе готового набора файлов kr9.html и style.css, создание скрипта script.js.

2. С помощью API Makeup API и целевой ссылки для метода fetch, создание внешней функции, которая перебирает элементы (объекты) массива и использует значения объектов полученного из удаленной базы массива для создания нового массива объектов, который выводится в консоль.

3. Создание в js функции krData (параметр), которая обрабатывается вложенной внешней функцией.

4. Тестирование скрипта script.js, обработка ошибок методом try и catch или catch и проверка результата.

В результате в консоль должен быть выведен новый массив объектов базы косметических изделий в соответствии с заданными условиями выборки, где

1. первый столбец – значение поля brand
2. второй столбец – значение поля name
3. третий столбец – значение поля описание
4. четвертый столбец – значение поля ссылка на image
5. пятый столбец – значение поля цена

Дополнительно применено условие, при котором вывод осуществляется только элементов, где цена выше 10$.

Подробная инструкция по API <https://makeup-api.herokuapp.com/> опубликована на английском языке. Для удобного использования данная инструкция была переведена на русский язык (приложение 1).

# ПЕРВЫЙ РАЗДЕЛ. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБАТЫВАЕМОЙ ТЕМЫ

## 1. API или программный интерфейс приложения

Для создания сайта, отвечающего всем перечисленным в постановке задачи требованиям, необходимо познакомиться с понятием API (англ. Application programming interface). Программный интерфейс приложения, интерфейс прикладного программирования, или же API - это набор функций, который позволяет одному программному обеспечению обмениваться данными с другим. Например, когда пользователь открывает приложение Facebook на телефоне, оно посылает запрос на сервер, который, в свою очередь, считывая запрос, отправляет необходимые данные назад в телефон. Чтобы лучше раскрыть суть данного термина, можно привести следующий пример: представим, что пользователь приложения - это посетитель ресторана, само приложение - меню, а сервер - это кухня. Посетитель выбрал самое вкусное блюдо в меню, но как же повар узнает, что именно нужно приготовить? Посредником, который отдаст новый заказ на кухню, а затем вернёт ответ сервера, то есть готовое блюдо, посетителю - это официант, и в этом примере он является тем самым связующим звеном - API. Стоит ещё раз отметить, что любой API состоит из вызова функций, которые требуют от программного обеспечения выполнения определённых действий. Описание функций, как правило, содержится в документации [1].

Прикладной программный интерфейс упрощает и ускоряет разработку программного обеспечения, поскольку позволяет добавлять функционал к старым приложениям или писать новые программы с использованием уже существующих технологий, а благодаря документации разработчику не нужно знать, как именно работает та или иная составляющая API, достаточно знать, какой функционал она предоставляет. Подводя итоги, можно отметить, что API является связующим звеном двух систем, скрывающим детали реализации и сложности некоторых деталей от разработчика.

## 1.1 Типы API

Прикладной программный интерфейс можно разделить на три категории: приватный, партнёрский и публичный [3]. Раскроем суть каждого термина подробнее.

Приватный API. Этот тип API используется внутри компаний и создан для того, чтобы разработчики могли интегрировать более новые системы и приложения в уже существующие или, наоборот, создавать новые продукты, используя уже готовые внутренние наработки. Даже если итоговый продукт становится публичным, приватный API доступен только разработчикам внутри компании - такой подход позволяет контролировать использование API и уберегает программы от злонамеренного вмешательства.

Партнёрский API, как и приватный, не даёт общего доступа к использованию ресурсов компании, но используется в случае сотрудничества компаний, когда одна сторона предоставляет своё программное обеспечение второй для достижения общих целей. Нельзя не отметить, что использование партнёрского API является своего рода рекламой для стороны, предоставляющей своё прикладное программное обеспечение партнёру.

Наконец, публичный API доступен для любых разработчиков, однако делится на два типа: открытый и коммерческий. Из названия понятно, что открытый API может быть использован без каких-либо ограничений, соглашений и дополнительных лицензий, а любая документация, относящаяся к открытому типу публичного API также находится в свободном доступе. Владельцы коммерческого открытого API взимают плату за использование их технологий, но, как правило, предлагают пробную бесплатную версию функционала.

## 1.2 Классификация API по сценариям использования

API можно классифицировать не только по параметру приватности, но и по системам, для которых они спроектированы. Выделим следующие типы: API для баз данных, для операционных систем, удалённые и web API. Рассмотрим подробнее каждый из представленных типов.

API для баз данных позволяет установить связь между приложением и системой управления базами данных. Разработчики общаются с базой данных посредством запросов для доступа к данным и таблицам, и такие сервисы, как The Drupal 7 Database API позволяют пользователям писать единые запросы для разных баз данных, таких как Oracle или MongoDB.

API для операционных систем определяет, как приложения используют ресурсы и различные сервисы ОС. Стоит отметить, что каждая из множества операционных систем, например, Windows, Linux или MacOS, содержит свой набор API и соответствующую документацию.

Удалённые API определяют правила взаимодействия для приложений, которые находятся на удалённых машинах. Другими словами, программное обеспечение запрашивает доступ к ресурсам, расположенным на другом устройстве. С появлением сетей, вроде Интернета, API стали основываться на web стандартах. Java Database Connectivity API - один из примеров удалённого прикладного программного интерфейса.

Наконец, рассмотрим понятие web API. Этот тип API предоставляет функциональную передачу между двумя web-based системами, которые представляют собой клиент-серверную архитектуру. Такой прикладной программный интерфейс главным образом доставляет запросы от web-приложения и ответы от сервера, используя HTTP-протокол. Разработчики могут использовать этот тип API, чтобы расширить функционал своего приложения или сайта. Например, социальная сеть Вконтакте позволяет магазинам, интернет-площадки которых находится не только на отдельном сайте, но также и в группе во Вконтакте, связать эти два ресурса между собой посредством добавления чата сообщества прямиком внутрь сайта магазина [3].

## 1.3 Протоколы API

Для того, чтобы использовать различные типы API, необходимо следовать специальным протоколам, которые содержат различные правила, указывающие на доступные типы данных и команды. Наиболее известными протоколами являются: REST, SOAP, RPC. Рассмотрим каждый из них по порядку.

Термин REST был введён в 2000 году, и не является протоколом, а скорее представляет собой архитектуру построения модели, состоящую из шести постулатов:

● клиент-серверная архитектура: интерфейс отделён от серверной части, - такой подход обеспечивает гибкость приложения и позволяет различным компонентам развиваться независимо друг от друга;

● отсутствие состояния означает, что клиентские данные не сохраняются на сервере между запросами;

● универсальный интерфейс предполагает наличие единого пути взаимодействия с сервером, независимо от устройства или типа приложения;

● кэшируемость позволяет сохранять ответы сервера, поэтому ответ по протоколу REST API должен содержать переменную, которая сообщает о том, могут эти данные быть закэшированы или нет;

● архитектура системы представляет собой слои, причём слои ничего не знают друг о друге. Между клиентом и сервером может быть много промежуточных уровней, что позволяет улучшить доступность системы за счёт распределения нагрузки и предоставления общих кэшей.

REST позволяет использовать данные как ресурсы, каждый из которых представляется в виде уникального URL, запросив который, можно получить доступ к необходимым данным. API, основанные на REST-архитектуре, используют HTTP запросы, такие как get, post, connect и другие, чтобы работать с ресурсами. Крупные бренды, вроде Twitter, Facebook или Netflex используют REST API, чтобы повысить узнаваемость своего бренда[1].

Протокол SOAP или протокол простого доступа к объектам - надёжный протокол, являющийся одним из типов web API, который используется с конца 1990-х годов. Он является расширяемым, независимым и нейтральным, то есть способным работать с различными протоколами связи (такими как HTTP, SMTP, TCP и другими). Спецификация этого протокола включает в себя:

● модель обработки, которая указывает, как именно стоит обрабатывать сообщения SOAP;

● правила привязки протокола, содержащие правила использования SOAP с базовыми протоколами, такими как HTTP;

● конструктор сообщений, выстраивающий структуру сообщений SOAP.

Стоит отметить, что этот протокол первым стандартизировал использование сетевых подключений приложениями. Однако SOAP содержит строгие правила и жёсткие стандарты, которые, порой, являются ресурсоёмкими, поэтому многие разработчики выбирают REST, нежели SOAP.

Наконец, поговорим о протоколе RPC или о протоколе удалённого вызова процедур. Эти протоколы являются самыми старыми и самыми простыми типами API. Главной целью RPC является организация распределённых вычислений. Наибольшая эффективность достигается в приложениях с малым временем ответа и малым объёмом передаваемых данных. RPC можно разделить на две группы: XML-RPC и JSON-RPC, - которые, как ясно из названия, используют форматы XML и JSON для передачи данных соответственно. Однако RPC-протоколы очень сложно поддерживать или обновлять, поскольку из-за плотной связи RPC API разработчику нужно хорошо понимать, как изменение может повлиять на другие составляющие этого протокола [1].

Таким образом, наиболее предпочитаемым стандартом для построения приложений, которые связаны по сети, является архитектура REST, поскольку она содержит в себе все стандарты, лежащие в основе всемирной паутины и является более простой, нежели web сервисы, основанные на SOAP. И в отличие от RCP, слабосвязанная многоуровневая архитектура позволяет разработчикам легко вносить изменения.

## 1.4 Makeup API

Makeup API - это специальный интерфейс, который позволяет получать информацию из баз данных сайта https://makeup-api.herokuapp.com/, используя http-запросы к серверу. Удобство использования API, как уже было сказано ранее, заключается в том, что разработчику не нужно знать устройство базы: из каких таблиц и полей она состоит, как эти таблицы связаны между собой, - достаточно только осуществлять уже готовые API-запросы. Синтаксис запросов и документацию можно посмотреть непосредственно на сайте https://makeup-api.herokuapp.com/ [2].

Все запросы имеют следующий формат:

http://makeup-api.herokuapp.com/api/v1/products.json?brand="Annabelle"  
&product\_type="blush",

Рассмотрим все его составляющие по порядку:

● http:// - протокол соединения;

● makeup-api.herokuapp.com/api/v1/products.json - адрес API-сервиса файл в формате JSON;

● ?brand="Annabelle"&product\_type="blush">&price\_greater\_than=10 –

параметры запроса. Полный список можно посмотреть в описании метода. Здесь разработчик сообщает серверу, что хотел бы получить список товаров бренда Annabelle категорий blush;

Сервер вернёт JSON-объект с данными, которые мы запросили. В случае ошибки, разработчик также получит соответствующее сообщение. Стоит пояснить, что формат JSON - это текстовый формат представления данных, особенностью которого является представление информации в виде объектов языка JavaScript. Первоначально, как следует из определения, этот формат создавался для языка JavaScript, однако позже, благодаря удобству представления данных в таком виде, во многие языки были добавлены библиотеки для работы с этим форматом данных[1].

В данном случае, ответ сервера будет выглядеть следующим образом:

{"response": [{"id":464,"brand":"annabelle","name":"Annabelle Blush ","price":"7.99","price\_sign":null,"currency":null,"image\_link":"https://d3t32hsnjxo7q6.cloudfront.net/i/0af800d01a052da93b9145fb59f849ba\_ra,w158,h184\_pa,w158,h184.jpg","product\_link":"https://well.ca/products/annabelle-blush\_108726.html","website\_link":"https://well.ca","description":"Annabelle Blush has a weightless and velvety texture that makes it easy to apply and\n perfectly blendable. Perfect to get a healthy and youthful look! ","rating":null,"category":"powder","product\_type":"blush","tag\_list":[],"created\_at":"2016-10-01T18:35:24.089Z","updated\_at":"2017-12-23T21:08:46.649Z","product\_api\_url":"http://makeup-api.herokuapp.com/api/v1/products/464.json","api\_featured\_image":"//s3.amazonaws.com/donovanbailey/products/api\_featured\_images/000/000/464/original/data?1514063326","product\_colors":[{"hex\_value":"#C97E79","colour\_name":"Hibicus "},{"hex\_value":"#D999AA","colour\_name":"Lotus "},{"hex\_value":"#D68D98","colour\_name":"Peony "},{"hex\_value":"#DB92A6","colour\_name":"Rosebud "},{"hex\_value":"#C9888C","colour\_name":"Spring "}]}]}

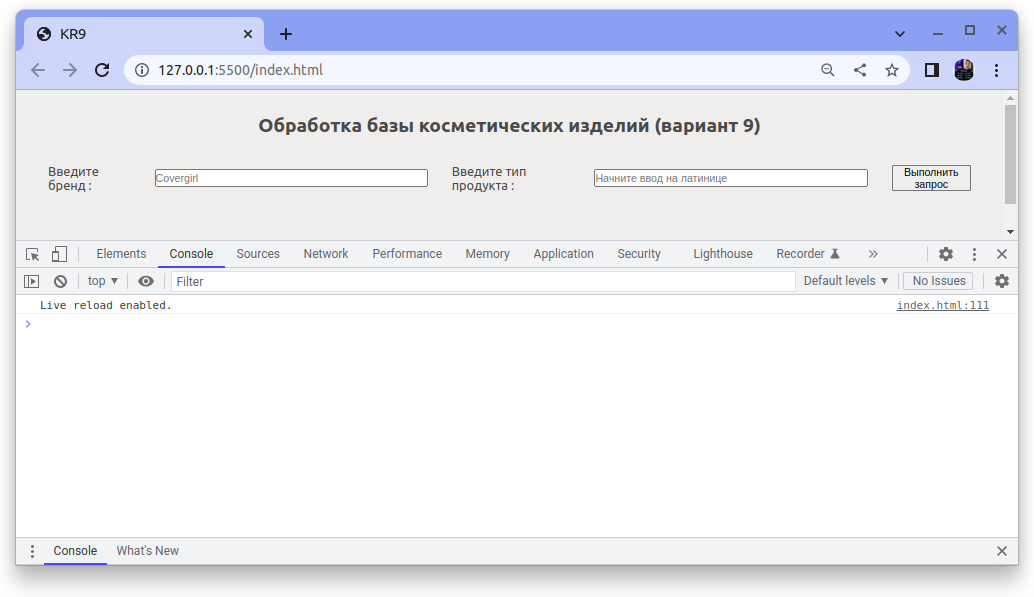
Удобство работы с API заключается ещё и в том, что в документации заранее обговорён формат ответа, что значительно упрощает процесс разработки.

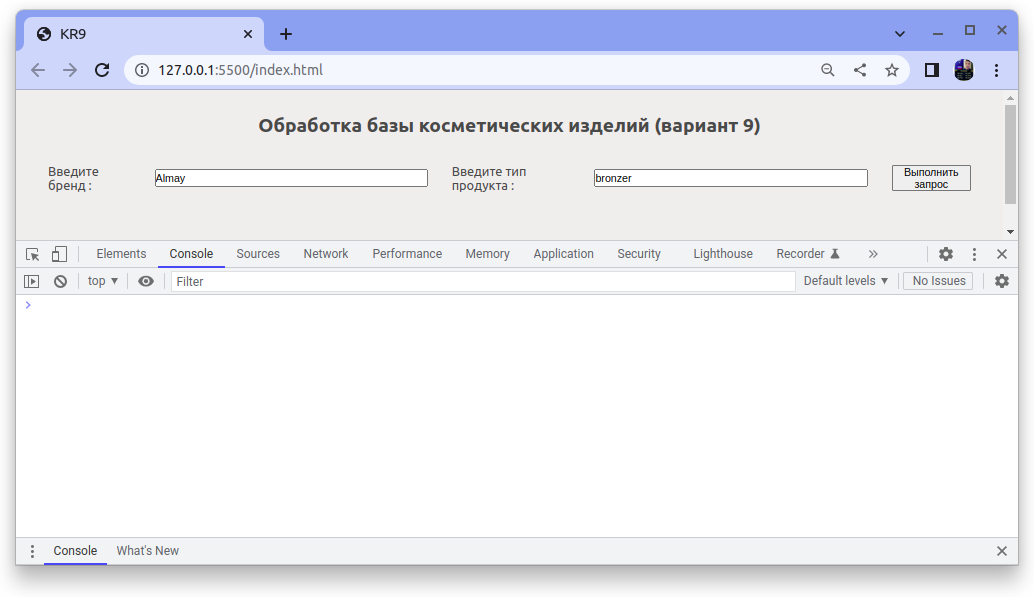
# ВТОРОЙ РАЗДЕЛ. ПРАКТИЧЕСКИЕ РЕЗУЛЬТАТЫ РАБОТЫ

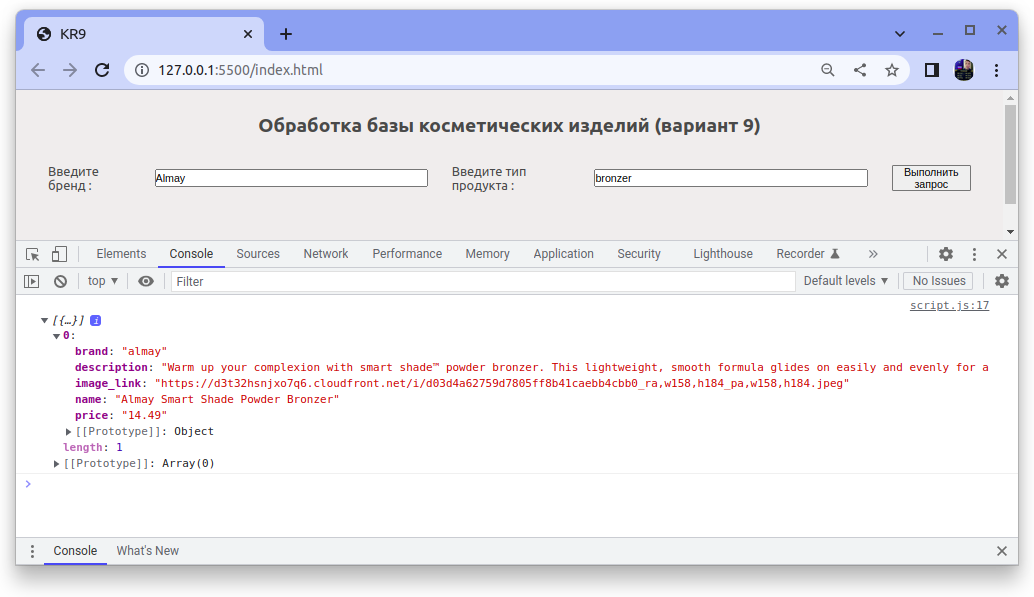
## 1. Описание

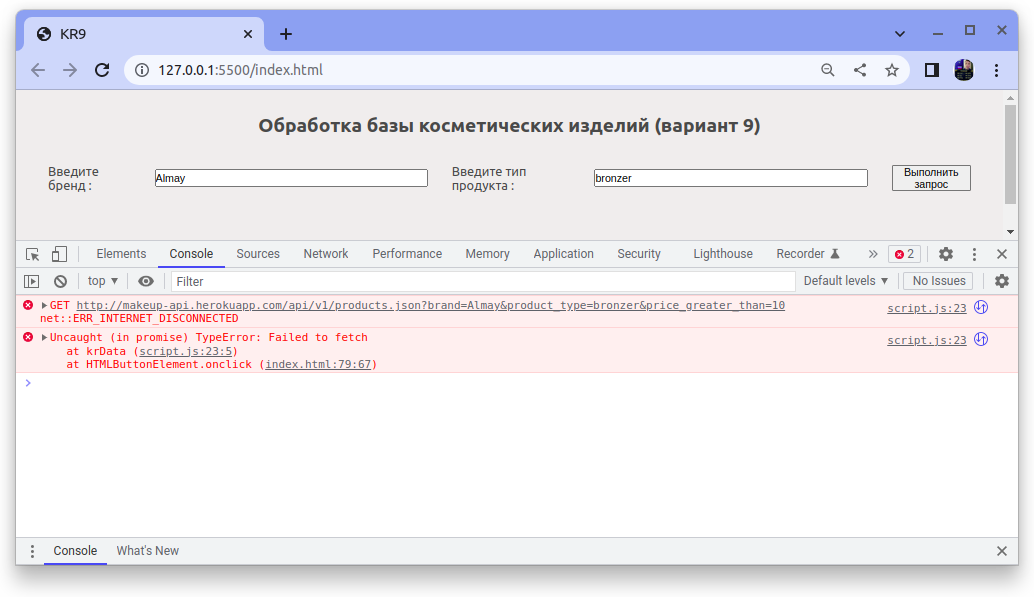
При открытии документа отображается пользовательский интерфейс с формой, в полях которой можно указать параметры поиска. При нажатии кнопки «Выполнить запрос» скрипт обращается к серверу и отправляет запрос с заданными параметрами. Сервер возвращает данные по запросу в формате JSON.Полученные данные обрабатываются и создается новый массив объектов базы косметических изделий в соответствии с заданными условиями выборки который выводится консоль [2].

## 2. Скриншоты результатов в браузере и консоли

  
*Рисунок 1. Вид после загрузки документа*

  
*Рисунок 2. Выбор параметров для поиска*

  
*Рисунок 3. Вывод результата*

  
*Рисунок 4. Вывод ошибки, если источник недоступен*

## 3. Код скрипта с комментариями

*//Объявляем переменные*

const brand\_item = document.getElementById("marka").value

const type\_item = document.getElementById("product").value

const makeup = []

*// перебираем полученный массив и из него создаем новый*

function getValue(array) {

for (let i = 0; i < array.length; i++) {

makeup[i] = {

brand: array[i].brand,

name: array[i].name,

description: array[i].description,

image\_link: array[i].image\_link,

price: array[i].price,

}

}

console.log(makeup) *//выводим новый массив в консоль*

}

*// Обновляем переменные данными из форм. Получаем данные из JSON*

const krData = function (brand\_item, type\_item) {

try {

var brand\_item = document.getElementById("marka").value

var type\_item = document.getElementById("product").value

fetch(`http://makeup-api.herokuapp.com/api/v1/products.json?bran  
d=${brand\_item}&product\_type=${type\_item}&price\_greater\_than=10 `,)

.then(response => response.json())

.then(data => getValue(data))

}

catch(err) {console.log("File not found")}s

}

# ЗАКЛЮЧЕНИЕ

При выполнении курсовой работы была изучена работа API на примере Makeup API. На базе готового набора файлов kr9.html и style.css, мы создали скрипт script.js. С помощью Makeup API и целевой ссылки для метода fetch, создали внешнюю функцию, которая перебирает элементы (объекты) массива и использует значения объектов полученного из удаленной базы массива для создания нового массива объектов, который выводится в консоль. Для обработки внешней функции, мы добавили в JS функцию krData (параметр). Протестировав скрипт script.js и обработав ошибки методом try и catch, мы создали функциональный массив объектов базы косметических изделий в соответствии с заданными условиями выборки.

Для удобства работы с документацией был выполнен ее перевод (Приложение 1). Созданный скрипт работает в соответствии с техническим заданием.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ

1. Хавербеке Марейн Выразительный JavaScript. Современное веб-программирование. 3-е изд. – СПБ.: Питер, 2021. – 480с.: ил. – (Серия «Для профессионалов»).
2. https://makeup-api.herokuapp.com/ (дата обращения 28.04.2023)
3. https://www.schoolsw3.com/js/index.php (дата обращения 28.04.2023)

# *Приложение 1*

**Перевод документации по API** [**https://makeup-api.herokuapp.com/**](https://makeup-api.herokuapp.com/)

Makeup API

ОСНОВНЫЕ ДАННЫЕ

**Протокол:** передача через HTTP. Это API соответствует принципам REST.

**Источник API:** ссылка на актуальную версию API:   
*http://makeup-api.herokuapp.com/api/v1/products.json*

**Назначение API**: Поиск косметических продуктов

**Тип запроса:** GET

**Формат ответа:** JSON

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

Для поиска бренда “*maybelline*”, используется ‘*brand=maybelline*’. Например: *http://makeup-api.herokuapp.com/api/v1/products.json?brand=maybelline*

Для поиска “*lipsticks*” в бренде “*covergirl*”, можно использовать ‘*brand=covergirl*’ и ‘*&product\_type=lipstic*k’. Например: *http://makeup-api.herokuapp.com/api/v1/products.json?brand=covergirl&product\_type=lipstick*

ПАРАМЕТРЫ ПОИСКА

Можно использовать фильтры: *brand, price, product category, tags* и так далее. Полный список приведен в таблице ниже. Этот API поддерживает все основные данные JSON.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип данных | Описание |
| product\_type | строка | Тип искомой косметики (например, помада, подводка для глаз). См. список типов товаров ниже. Возвращает список всех товаров этого типа |
| product\_category | строка | Подкатегория для каждого типа косметики. (т.е. блеск для губ относится к категории губной помады). Смотрите типы товаров ниже. Если категория существует, она будет находиться в разделе «По категориям». Вернет список всех товаров этой категории |
| product\_tags | строка, список разделенный запятыми | Параметры, которыми можно пометить каждый товар. (т.е. веган). Каждый товар может иметь несколько тегов. Если теги существуют, они будут в типах товаров в разделе «По тегам». Возвращает список товаров, отфильтрованных по всем указанным тегам |
| brand | строка | Марка товаров. Вернет все товары для каждого бренда |
| price\_greater\_than | число | Вернет список товаров с ценой выше указанного числа (только одно значение) |
| price\_less\_than | число | Вернет список товаров с ценой ниже указанного числа (только одно значение) |
| rating\_greater\_than | число | Вернет список товаров с рейтингом выше указанного числа (только одно значение) |
| rating\_less\_than | число | Вернет список товаров с рейтингом ниже указанного числа (только одно значение) |

|  |  |  |  |
| --- | --- | --- | --- |
| Список тегов | Список брендов | | Тип товара |
| Canadian  CertClean  Chemical Free  Dairy Free  EWG Verified  EcoCert  Fair Trade  Gluten Free  Hypoallergenic  Natural  No Talc  Non-GMO  Organic  Peanut Free Product  Sugar Free  USDA Organic  Vegan  alcohol free  cruelty free  oil free  purpicks  silicone free  water free | almay  alva  anna sui  annabelle  benefit  boosh  burt's bees  butter london  c'est moi  cargo cosmetics  china glaze  clinique  coastal classic creation  colourpop  covergirl  dalish  deciem  dior  dr. hauschka  e.l.f.  essie  fenty  glossier  green people  iman  l'oreal  lotus cosmetics usa  maia's mineral galaxy | marcelle  marienatie  maybelline  milani  mineral fusionmisa  mistura  moov  nudus  nyx  orly  pacifica  penny lane organics  physicians formula  piggy paint  pure anada  rejuva minerals  revlon  sally b's skin yummies  salon perfect  sante  sinful colours  smashbox  stila  suncoat  w3llpeople  wet n wild  zorah  zorah biocosmetiques | blush  bronzer  eyebrow  eyeliner  eyeshadow  foundation  lip\_liner  lipstick  mascara  nail\_polish |