1. https://www.detmir.ru/search/results/?qt=сок&20агуша /на сайте детского мира ищем сок марки агуша

<https://kapika.ru/poisk/?query=>сапоги+резиновые&s=Поиск /на сайте Капика ищем резиновые сапоги

1. GET запросы используются для запроса какой-либо информации с сервера, которая не является конфиденциальной, при этом никакая информация на сервер не добавляется и не меняется, такие запросы не могут быть очень больших размеров. Также такие запросы кешируются, если не указано иное. POST запросы используются с большими объемами данных, для конфиденциальной информации и не видны пользователю. Данные запросы не кешируются по дефолту, поэтому вся инормация из одного запроса не переносится в другой, даже при идентичном содержимом. В данныз запросах поддерживается отправка файлов.
2. метод get, тк запрашивается не какая-то секретная личная информация, и то, что видно в url может быть использовано пользователем (проверка правильности запроса, те ли товары ищет сайт), плюс такой поиск для удобства можно добавить в закладки
3. метод post, т.к. здесь будут отправляться персональные данные, которые не должны быть видны сторонним пользователям и не сохраняться браузером.
4. <https://api.giphy.com/v1/gifs/search?api_key=SET_KEY_HERE&q=cats&limit=25&offset=0&rating=g&lang=en>

api\_key=SET\_KEY\_HERE здесь отражается API Keyпользователя

q=cats -- query, поиск начинается по признаку cats

limit=25 выведется 25 объектов(это значение по дефолту), чтобы возвращал 5, нужно написать limit=5

offset=0 стартовая позиция в результатах (0 значение по дефолту)

rating=g означает, что выборка соответствует рейтингу g, самая общепринятый по соответствию нормам (нет запрещенного контента)

lang=en язык содержания (здесь – английский)

1. консоль разработчика, вкладка network, в колонке name выбрать нужный адрес, справа вкладка headers
2. Cross-Origin Resource Sharing. механизм, который позволяет подгружать на страницу данные из источников, отличных от источника самой страницы. Если хотя бы один из параметров источников отличается (хостинг, порт, протокол), то такой запрос считается Cross-Origin. Также он позволяет регулировать использование информации по таким запросам, ограничивать их или проверять возможность их совершения, что препятствует умышленным или случайным опасным действиям.
3. application/x-www-form-urlencoded , multipart/form-data или text/plain .
4. const dogsJson = {
5. breed: "Beagle",

size: "large",

color: "orange",

age: 6,

};

submit.onclick = function (e) {

e.preventDefault();

fetch("localhost/pets/add", {

method: "POST",

headers: {

"Content-Type": "application/json;charset=utf-8",

},

body: JSON.stringify(dogsJson),

})

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

.then((dogs) => {

console.log(dogs);

})

.catch((error) => console.log(error));

};

1. Promise – это объект, который содержит в себе функции, позволяющие отслеживать выполнение какого-либо процесса, и по итогу его выполнения привязывать какие-либо действия к результату. Например, если процесс выполнен, то через метод .then после завершения промиса запустится указанная программа(вывести фразу, показать сумму и тд), а если результат определен как ошибка, то вызовется соответствующее сообщение об ошибке. Можно обрабатывать только один из результатов resolve или reject. То есть данным объектом мы можем как бы обещать, что при выполнении одного действия выполнится какое-либо другое, заданное автором промиса, и этот результат будет обработан.