API\_NASA

<https://mars.nasa.gov/msl-raw-images/msss/03344/mcam/3344MR1012150011502683E01_DXXX.jpg>

**GET** - метод для чтения данных с сайта. Например, для доступа к указанной странице. Он говорит серверу, что клиент хочет прочитать указанный документ. На практике этот метод используется чаще всего, например, в интернет-магазинах на странице каталога. Фильтры, которые выбирает пользователь, передаются через метод GET.

**НТТP** - Широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

Аббревиатура HTTP расшифровывается как HyperText Transfer Protocol, «протокол передачи гипертекста».

**Метод HTTP запросов**

HTTP определяет множество **методов запроса**, которые указывают, какое желаемое действие выполнится для данного ресурса. Несмотря на то, что их названия могут быть существительными, эти методы запроса иногда называются *HTTP глаголами*. Каждый реализует свою семантику, но каждая группа команд разделяет общие свойства: так, методы могут быть

**Безопасный** Метод HTTP является **безопасным**, если он не меняет состояние сервера. Другими словами, безопасный метод проводит операции "только чтение" (read-only). Безопасные:GET, HEAD или OPTINS.

Даже если безопасные методы являются по существу "только для чтения", сервер всё равно может сменить своё состояние: например, он может сохранять статистику. Браузеры могут вызывать безопасные методы это позволяет им выполнять некоторые действия, например, предварительная загрузка без риска. Поисковые роботы также полагаются на вызовы безопасных методов.

Правильная реализация безопасного метода - это ответственность **серверного приложения**, потому что сам веб-сервер, будь то Apache, nginx, IIS это соблюсти не сможет.

**Идемпотентный** Если повторный идентичный запрос, сделанный один или несколько раз подряд, имеет один и тот же эффект, не изменяющий состояние сервера. Другими словами, идемпотентный метод не должен иметь никаких побочных эффектов (side-effects), кроме сбора статистики или подобных операций. Корректно реализованные методы GET, HEAD, PUL, DELETE

**Кешируемый** это HTTP-ответы, которые могут быть закешированы, то есть сохранены для дальнейшего восстановления и использования позже, тем самым снижая число запросов к серверу. Не все HTTP-ответы могут быть закешированы. Вот несколько ограничений:

Метод, используемый в запросе, кешируемый, если это GET или HEAD. Ответ для POST или PATCH запросов может также быть закеширован, если указан признак "свежести" данных и установлен заголовок Content-Location (tn-us)( **Content-Location указывает альтернативное расположение возвращаемых данных. Основное использование заключается в указании URL-адреса ресурса, переданного в результате согласования содержимого.**), но это редко реализуется. Другие методы, такие как PUT и DELETE не кешируемые, и результат их выполнения не кешируется.

Коды ответа, известные системе кеширования, которые рассматриваются как кешируемые: 200, 203, 204, 206, 300, 301, 404, 405, 410, 414, 501

Отсутствуют специальные заголовки в ответе, которые предотвращают кеширование: например, Cache-control (используется для задания инструкций кеширования как для запросов, так и для ответов. Инструкции кеширования однонаправленные: заданная инструкция в запросе не подразумевает, что такая же инструкция будет указана в ответе Cache-control)

**Методы**

* **GET** запрашивает представление ресурса. Запросы с использованием этого метода могут только извлекать данные.
* **HEAD** запрашивает ресурс так же, как и метод GET, но без тела ответа.
* **POST** используется для отправки сущностей к определённому ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты на сервере.
* **PUT** заменяет все текущие представления ресурса данными запроса.
* **DELETE** удаляет указанный ресурс.
* **CONNECT** устанавливает "туннель" к серверу, определённому по ресурсу.
* **OPTIONS** используется для описания параметров соединения с ресурсом.
* **RACE** выполняет вызов возвращаемого тестового сообщения с ресурса.
* **PATCH** используется для частичного изменения ресурса.

**URL** обозначает Uniform Resource Locator. URL это лишь адрес, который выдан уникальному ресурсу в интернете. В теории, каждый корректный URL ведёт на уникальный ресурс. Такими ресурсами могут быть HTML-страница, CSS-файл, изображение и т.д. На практике, существуют некоторые исключения, когда, например, URL ведёт на ресурс, который больше не существует или который был перемещён. Поскольку ресурс, доступный по URL, а также сам URL обрабатываются веб-сервером, его владелец должен внимательно следить за размещаемыми ресурсами и связанными с ними URL.

**?** – разделитель между параметрами в адресе

параметр даты по Земному календарю 1 января 2022 года + api\_key=........ личный ключ полученный при регистрации на сайте НАСА.