7 Мощных библиотек HTTP-клиентов и запросов Node.js, которые необходимо знать разработчику

Описание

Ознакомьтесь с несколькими модными библиотеками HTTP-запросов Node.js или клиентскими инструментами, которые позволят вашим веб-приложениям, веб-сайтам или приложениям обмениваться данными с другими серверами. Если вы собираетесь разработать веб-приложение, которое будет взаимодействовать со сторонними или собственными серверами для получения данных и выполнения таких задач, как обмен сообщениями, заказ товаров и т.д., вашему приложению необходимо взаимодействовать с веб-интерфейсами API. Для этого можно использовать HTTP-запросы.

Вместо того чтобы программировать модуль HTTP-запросов с нуля, разработчики полагаются на различные библиотеки HTTP-запросов Node.js. Большинство из этих инструментов доступны по лицензии с открытым исходным кодом. Таким образом, не требуется никаких дополнительных затрат. Использование HTTP-клиентов и библиотек запросов Node.js является современным способом разработки приложений, и большинство разработчиков осваивают его в начале своей карьеры. Поэтому, если вы начинающий разработчик или просто хотите узнать о популярных библиотеках HTTP-запросов для Node.js, вы попали по адресу. Наслаждайтесь чтением до конца и откройте для себя некоторые интересные инструменты для работы с HTTP-запросами.

Что такое HTTP-запросы?

Клиент, использующий веб-приложение или мобильное приложение, выполняет HTTP-запросы к именованному хосту. Хост обычно располагается на собственном или стороннем сервере. Клиент использует различные компоненты URL, такие как хост, схема, строка запроса и путь, чтобы выполнить HTTP-вызов к именованному хосту на сервере. В веб-приложении или мобильном приложении может быть как несколько, так и множество HTTP-запросов. Независимо от их назначения, для правильного составления HTTP-запроса необходимо наличие следующих элементов:

- Строка HTTP-запроса.
- Поля HTTP-заголовка или заголовки в последовательности.

- При необходимости может присутствовать и тело сообщения.

Строка запроса

В сообщении HTTP-запроса строка запроса является первым элементом. Однако она включает в себя еще три элемента: метод запроса, путь запроса и версию HTTP. Метод запроса обычно представляет собой однословную команду для сервера. Например, обычными методами запроса являются GET, HEAD, POST, PUT, CONNECT и т.д. Между методом и версией HTTP должен быть указан путь или местоположение ресурса на целевом сервере.

```
GET /applications/htp/cics/index.html HTTP/1.1
```

HTTP-заголовки

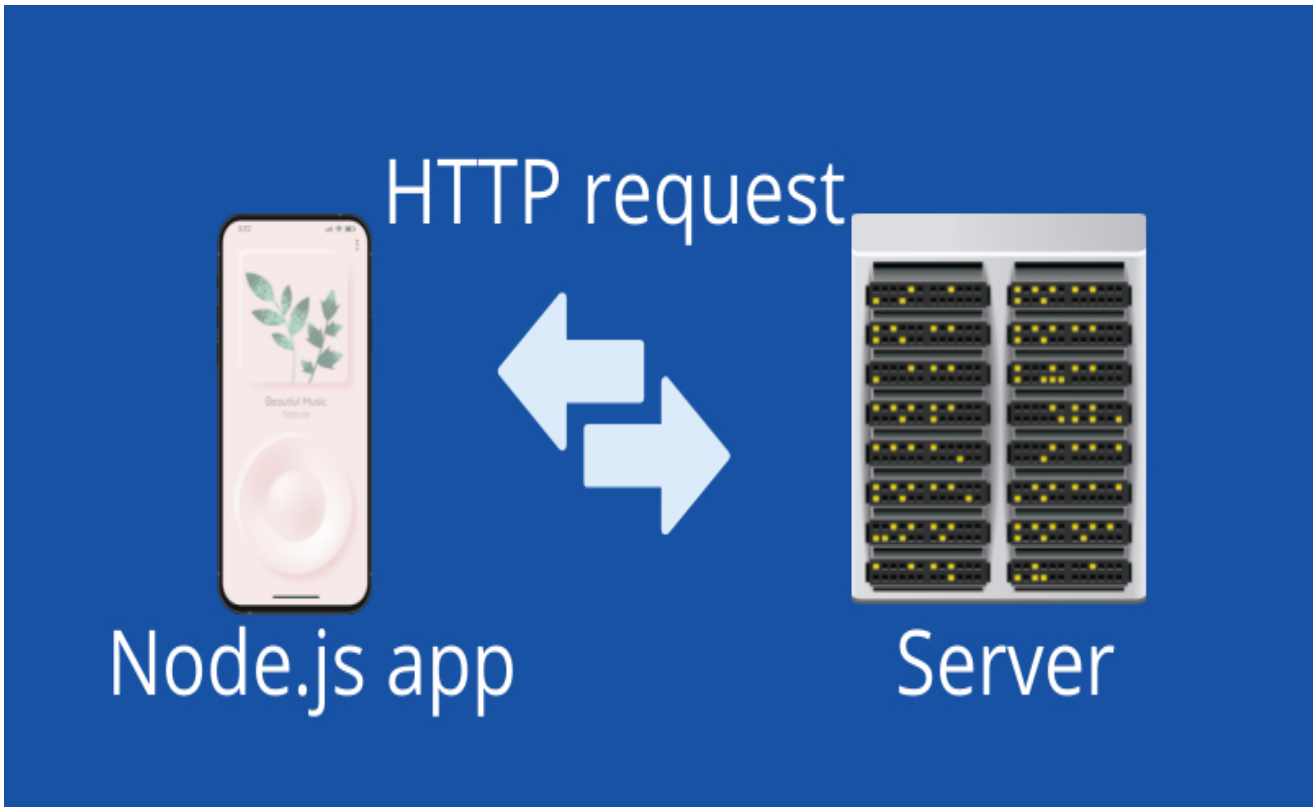
Чтобы помочь получателю получить важную информацию о запросе, необходимо поместить заголовок сообщения. Оно сообщает серверу-получателю, что это за сообщение, информацию об отправителе и способ связи. Например, вы хотите, чтобы ваше приложение получало документы, которые были обновлены 11 января 2022 года в 10:00:00 GMT на английском и французском языках. Тогда вы должны использовать следующий заголовок:

```
Accept-Language: en, fr<br>If-Modified-Since: Tue, 11 Jan 2022 10:00:00 GMT
```

Тело сообщения

Текстовое содержимое HTTP-сообщения – это entity-body или тело сообщения. Оно не является обязательным для всех методов запроса. Например, не нужно включать тело сообщения, если вы вызываете HTTP-запрос методом GET. Напротив, если вы используете метод POST для отправки данных на сервер с помощью HTTP-запроса, вы должны включить сообщение об инструкциях по отправке данных.

Что такое HTTP-запросы в Node?



В приложениях реального времени и веб-приложениях, где программа должна взаимодействовать с внешними или внутренними серверами и получать данные для клиентского приложения, необходимо выполнять HTTP-запросы в Node.js. Здесь клиентское приложение и сервер функционируют как два компьютера. HTTP-запрос от серверной среды исполнения Node.js – это первый набор данных, передаваемых между этими двумя компьютерами.

Целевой сервер обрабатывает этот HTTP-запрос и отправляет в ответ другой набор пакетов данных. Мы называем это ответом. Разработчики активно используют Node.js для таких задач, как чаты, сбор данных с форм, браузерные игры, API на объектной базе данных, потоковая передача данных, ввод данных в очередь и т.д. Во всех этих сценариях использования HTTP-запросы в Node играют важную роль.

Необходимость библиотек HTTP-клиента и запросов Node.js

Библиотека HTTP-запросов для Node.js предоставляет упрощенный и чистый пользовательский интерфейс для создания HTTP-вызовов. Такие библиотеки или модули делают задачи, основанные на HTTP-запросах, действительно легкими.

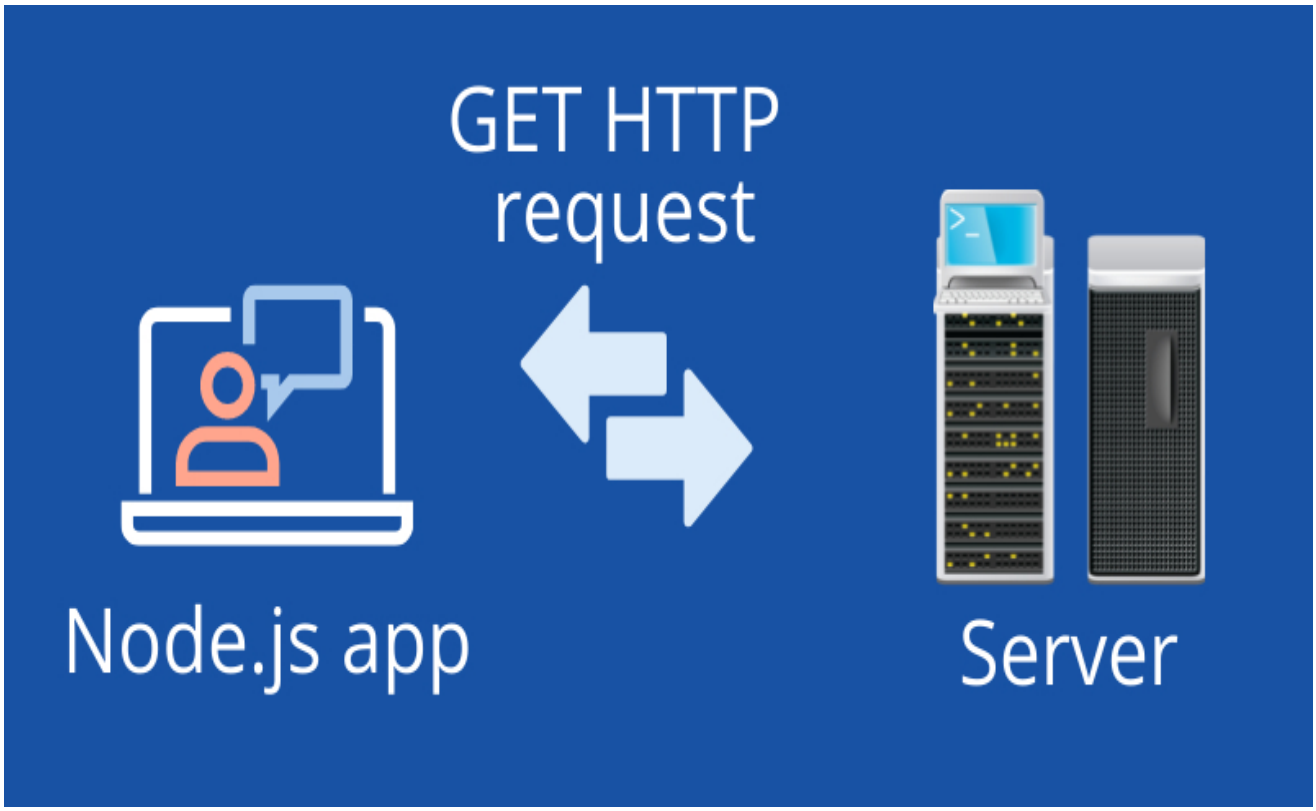
Использование таких инструментов позволяет избавиться от большого количества строк кода и сделать свои приложения более легкими. Кроме того, вы экономите время, не создавая с нуля полноценный модуль HTTP-запросов для Node.js. Кроме того, вы не поручаете эту задачу стороннему разработчику и экономите на стоимости разработки.

Вы можете выбрать подходящую библиотеку HTTP-запросов для Node.js для своего проекта разработки приложения и позволить ей обрабатывать все HTTP-вызовы, которые будет выполнять ваше приложение. Все эти платные и открытые библиотеки имеют определенный фреймворк для создания HTTP-запросов. Этот фреймворк основан на встроенных в Java классах HTTP. Кроме того, эти библиотеки используют Simple Logging Facade for Java, SLF4J, в качестве единственной внешней зависимости. Таким образом, результирующая кодовая база библиотеки становится исключительно легковесной.

Как работает HTTP-запрос в Node.js?

С помощью веб-браузера вашего компьютера можно сопоставить HTTP-запросы, позволяющие соединиться с удаленным сервером, обслуживающим GRE-приложения. Когда вы нажимаете на ссылку регистрации, браузер запрашивает у сервера некоторые данные. Затем, когда вы заполняете экзаменационную форму GRE, браузер отправляет данные на сервер и просит его сохранить их. Для того чтобы события HTTP-запроса работали, программисты используют различные методы запроса, приведенные ниже:

GET



Метод GET максимально используется для вызовов API и других HTTPS-запросов, где не требуется изменение данных. В основном он устанавливает первичное соединение с сервером и позволяет клиентскому приложению запрашивать данные с сервера.

DELETE

При удалении ресурсов с удаленного сервера среда выполнения Node.js выполняет HTTP-вызов на основе метода DELETE.

POST

HTTP-запросы на основе метода POST выполняются, когда веб-приложению или мобильному приложению требуется удаленный сервер для сохранения данных клиентского приложения.

PUT

Представьте, что вы зашли в веб-приложение социальной сети и изменили информацию о своей учетной записи. Это означает, что вы обновляете данные на сервере. В таких случаях базовая среда выполнения выполняет HTTP-запросы с

использованием метода PUT.

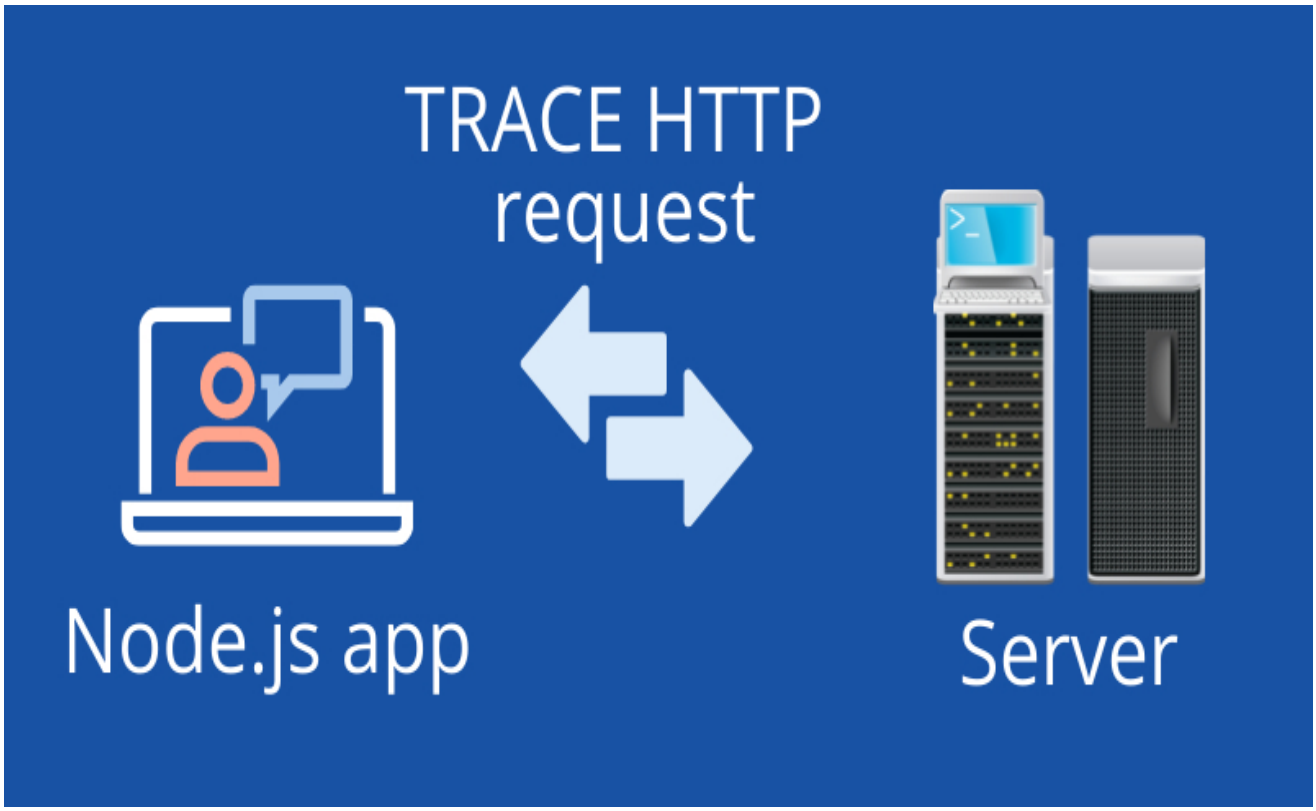
OPTIONS

Удаленный сервер может не поддерживать все методы передачи данных, которые может использовать ваше веб-приложение. Поэтому для понимания методов обмена данными с сервером приложение должно выполнить HTTP-запрос с использованием метода OPTIONS.

HEAD

Библиотека HTTP-вызовов будет выполнять запрос методом HEAD, если передаваемые данные велики и хочется избежать тела сообщения в ответных данных.

TRACE



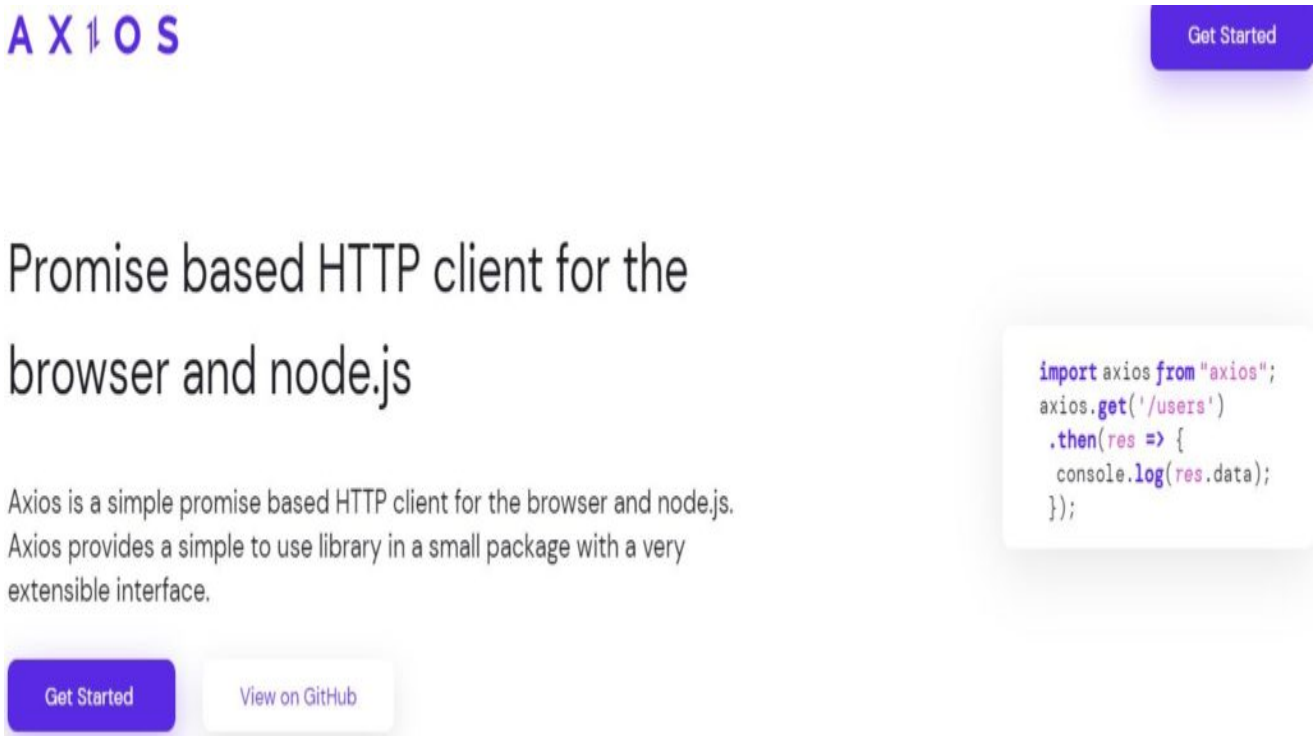
Метод TRACE для HTTP-запросов помогает клиентскому приложению регистрировать маршрут взаимодействия с сервером. Итак, мы рассмотрели основы, потребности и работу библиотек HTTP-запросов Node.js. На очереди – изучение различных HTTP-клиентов и библиотек запросов в Node.js, которые сделают вашу жизнь по разработке программ, приложений и сайтов более

доступной. Вот инструменты, которые необходимо изучить, если вы увлекаетесь HTTP-запросами:

7 Мощных библиотек HTTP-клиентов и запросов Node.js, которые необходимо знать разработчику

Axios

Axios – это популярная и легкая библиотека HTTP-запросов JavaScript Node.js, основанная на обещаниях. HTTP-клиент поддерживает современные веб-браузеры и веб-приложения Node.js. Помимо Google Chrome и Mozilla, он хорошо совместим с Internet Explorer 8 и более поздними версиями браузеров на базе Microsoft.



Вы должны использовать Axios вместо собственных стандартных библиотек Node.js HTTP или HTTPS для вызовов API благодаря следующим особенностям:

- Axios является наиболее подходящей библиотекой для остановки задач запроса.
- Она совместима со старыми и современными веб-браузерами.
- Вы получаете встроенную защиту от CSRF.
- Можно установить таймаут ответа.

- Предусмотрено автоматическое преобразование данных JSON.
- Можно создать отображение прогресса загрузки.
- Axios совместима с Promise API.

Библиотеку можно получить бесплатно на GitHub. Она поставляется с лицензией MIT. Таким образом, существует ограничение на повторное использование кода.

GOT

GOT – это мощная и удобная библиотека HTTP-запросов для Node.js. На момент написания статьи более 8 тыс. пакетов и более 4 млн. отчетов используют GOT для HTTP-запросов для Node.js. Кроме того, за проектом стоят более десяти экспертов-разработчиков, которые активно поддерживают библиотеку, чтобы ваши приложения работали стабильно и обслуживали клиентов без каких-либо проблем.

Несмотря на то, что библиотека поставляется с открытой лицензией MIT, ряд популярных компаний и брендов также доверяют ей выполнение HTTP-запросов. Например, Segment, Antora, GetVoIP, Exoframe, Natural Cycles, Microlink и другие используют библиотеку GOT.

SuperAgent

SuperAgent – еще одна предпочтительная библиотека HTTP-запросов среди разработчиков, поскольку она представляет собой прогрессивный ajax API. Она также является легковесной и не делает кодовую базу вашего конечного веб-приложения сложной и тяжелой. Она была разработана для того, чтобы обеспечить удобство чтения и гибкость при выполнении HTTP-запросов.

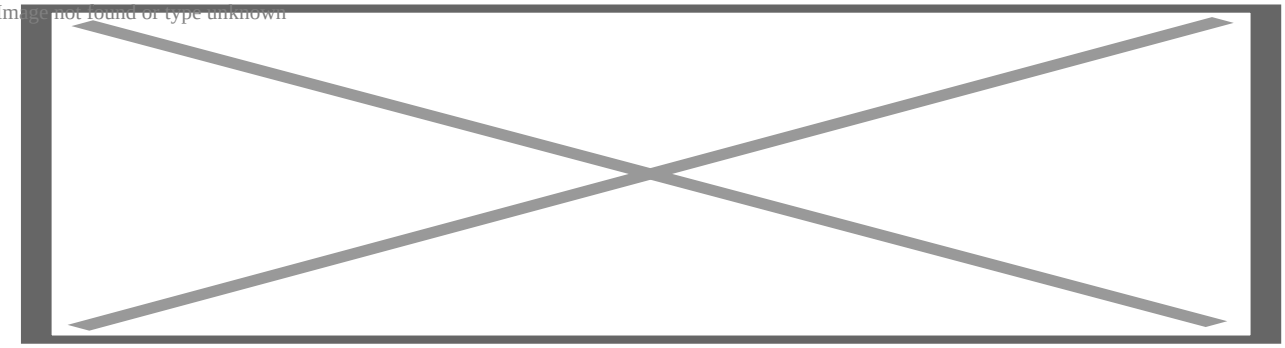
Основные возможности библиотеки:

- Опция тайм-аута для прерывания запросов, когда целевой сервер не отвечает и задерживается на неопределенное время.
- Для дополнительной безопасности можно включить аутентификацию с помощью метода .auth().
- Этот клиент HTTP-запросов Node.js предлагает передачу данных из запроса и в запрос.

Если вы перепробовали множество клиентов HTTP-запросов для Node.js и считаете их чрезмерно сложными, то можете попробовать Superagent. По сравнению со своими конкурентами он не требует больших усилий для обучения.

Node-Fetch

Еще одной модной и активно загружаемой библиотекой HTTP-запросов для Node.js на npm является Node Fetch. На момент написания статьи эта библиотека уже получила более 36 млн. загрузок в неделю.



Примечательными особенностями этого клиента HTTP-запросов Node.js являются:

- Последовательное выполнение при работе с API window.fetch.
- Библиотека сознательно идет на компромиссы, следуя спецификациям потоков и реализациям стандарта WhatWG fetch.
- Она использует родную библиотеку обещаний, но вы можете заменить ее на другую, которая вам нравится.
- Поставляется с собственными потоками Node для тела ответа и запроса.
- Может декодировать кодировки содержимого, такие как deflate или gzip.

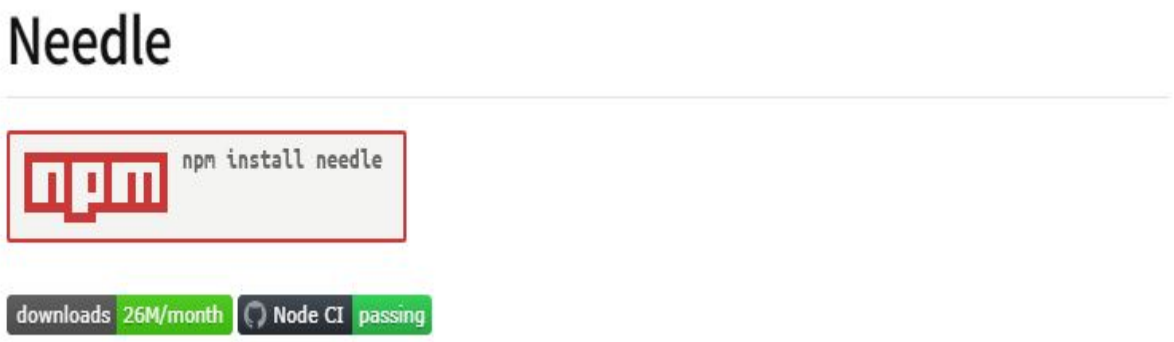
Как и его конкуренты, он поставляется с лицензией MIT для свободного использования в коммерческих и веб-приложениях.

KY Delightful HTTP Requests

KY использует браузерный fetch API для работы в качестве клиента HTTP-запросов. Он элегантен, миниатюрен и полностью разработан на языке JavaScript. Он подходит для среды исполнения Deno и различных современных браузеров. Одним из его достоинств является отсутствие зависимостей и, следовательно, легкость. Если вы до сих пор использовали fetch и хотите чего-то современного, попробуйте KY. Он предлагает множество преимуществ, таких как более простой API, повторение неудачных запросов, поддержка тайм-аута, возможность работы с JSON, возможность использования префикса URL и многое другое.

Needle

Needle – это потоковый и проворный клиент HTTP-запросов для Node.js. Его кодовая база включает всего две зависимости и считается легковесной библиотекой. Среди особенностей можно выделить следующие:



- HTTPS/HTTP-запросы обычными методами.
- Она поддерживает все собственные TLS-опции Node, такие как rejectUnauthorized, ciphers, secureProtocol и т.д.

Это библиотека с открытым исходным кодом и поставляется с известной лицензией MIT.

Undici

Undici – итальянское слово, означающее “одиннадцать”. Это клиент HTTP/1.1 для HTTP-вызовов для веб-приложений на базе среды исполнения Node.js. Модуль доступен под лицензией MIT для инструментов с открытым исходным кодом.

На момент написания статьи число еженедельных загрузок npm превысило 2 млн. Он пользуется популярностью среди разработчиков, поскольку является полнофункциональным клиентом HTTP-запросов и не использует никаких сторонних зависимостей.

Заключительные слова

Наконец, вы познакомились с некоторыми мощными и популярными библиотеками HTTP-клиентов и запросов Node.js. Все вышеперечисленные библиотеки и клиентские средства для работы с HTTP-запросами Node.js обладают некоторыми стандартными и уникальными возможностями. Уникальные возможности определяют, какой инструмент следует использовать. Перед выбором клиента и библиотеки HTTP-запросов необходимо выполнить определенную домашнюю работу. В противном случае вы можете столкнуться с ошибками, проблемами и неудобствами для пользователей в середине проекта.

Например, технологический стек проекта разработки должен подсказать, какой API следует выбрать. Еще один важный момент – количество HTTP-запросов, которые будет выполнять ваше приложение или веб-сайт. Кроме того, частота HTTP-запросов также является важным фактором. И последнее, но не менее важное: если вы хотите, чтобы конечный проект был легким или тяжелым, вам необходимо выбрать соответствующий API для HTTP-запросов в самом начале проекта.

Дата Создания
17.07.2023