**Питання по Node.js**

1. У чому різниця між Node.js і JavaScript?

Node.js є інтерпретатором, а також середовищем для JavaScript, яке в основному використовується для доступу або реалізації будь-яких неблокуючих процедур для будь-якого типу ОС. Тут працює двигун Google Chrome.

Тоді як JavaScript - це мова програмування, яка використовується для будь-яких завдань на стороні клієнта для інтернет-додатків. Саме тут працює двигун Firefox, Safari, Google Chrome тощо.

1. Що таке node.js?

node.js — це легка, але ефективна структура, створена на & nbsp; Рушій JavaScript в Chrome. Він об'єднує JavaScript безпосередньо в код місцевого виробника. Він використовується в структурі серверних інтернет-додатків, а також розширює JAVAScript API для забезпечення типових можливостей на стороні сервера. Люди використовують цю структуру для створення великих додатків, таких як односторінкові програми, веб-сайти, які транслюють відеокліпи, і різні інші онлайн-програми

1. Надати деякі переваги використання Node.js

* Це швидко, тому що він створюється на движку Chrome JavaScript, який прискорює свою колекцію при реалізації коду.
* Це асинхронно, тому що він ніколи не очікує, що API поверне інформацію.
* Він масштабується завдяки своїй системі подій, яка дозволяє веб-серверу реагувати неблокуючим методом.
* Це відкритий вихідний код, що забезпечує надзвичайне середовище з відкритим вихідним кодом, що призвело до деяких чудових компонентів, які додали ще більше функцій до додатків Node.js.
* Вузол.js ніколи не блокує інформацію, так як результат виходить частинами.

1. Які відмінності між Angular  і Node.js?

Angular — це відкритий вихідний код, який використовується для інтернет-додатків, а & Node.js, як правило, є кросплатформним середовищем виконання для додатків.

Кутова написана на TypeScript & Node.js складається такими мовами, як C, C++ і JavaScript.

Angular в основному використовується для розробки односторінкових клієнтських інтернет-додатків, а Node.js в основному використовується для розробки масштабованих, а також швидких серверних мережевих додатків.

Angular — це структура інтернет - програми, & Node.js має багато & nbsp; різні конструкції & nbsp; наприклад, Часткова.js, Вітрила.js, експрес.js і т.д.

Angular найкраще використовувати для розробки дуже енергійних, а також інтерактивних інтернет-додатків, а & Node.js найкраще використовувати для створення & nbsp; менші завдання

Незграбна допомагає розділити програму на частини MVC, а & Node.js допомагає у & nbsp; розробляти запити проти джерел даних & nbsp;

Angular добре використовується для розробки додатків в режимі реального часу. & Node.js добре використовується у випадках, коли він працює швидше, а також є більш масштабованим необхідним

Читати далі: 30 найкращих запитань для інтерв'ю Git, які вам потрібно знати у 2021 році

1. Чому Node.js однопотоковий?

Він служить однопотоковою версією, а не звичайним виконанням на основі потоків, щоб допомогти асинхронній обробці. Використовуючи асинхронну обробку, додаток може бути реалізовано набагато краще, а також набагато більш масштабоване через тонни інтернету.

1. Як саме працює Node.js?

Node.js - це онлайн-розробник, який працює в атмосфері v8, використовуючи JavaScript як мову сценаріїв. Він обслуговує однопотокову версію, а також неблокуючий вводу-виводу, який забезпечує високу ціну, оскільки він може керувати широким спектром синхронізованих вимог. Оскільки він використовує компонент HTTP, він може працювати на будь-якому автономному інтернет-сервері.

1. Де я можу використовувати Node.js?

Вузол.js може бути використаний для розробки:

Мережеві програми

Інтернет-додатки в режимі реального часу

Загальні об'єктивні програми

Дисперсне обладнання

1. У чому різниця між асинхронною і неблокуючою функцією?

Асинхронні (не паралельні) функції - це ті, за допомогою яких ми можемо створювати асинхронні HTTP-запити, які не чекають відповіді веб-сервера. Ці функції залишаються, щоб повернутися до запиту, на який він фактично отримав відповідь від веб-сервера.

Неблокуючі функції використовуються для процедур вводу-виводу. Вони швидко реагують на будь-яку запропоновану інформацію, а також підтримують роботу відповідно до вимог. Якщо рішення не повертається, API негайно повертається з помилкою.

1. Поясніть шоу на основі подій

Це програмна стратегія, яка в основному використовує функції для активації різних функцій. У деяких випадках це може бути натискання клавіші, клацання миші комп'ютера і т.д. При спрацьовуванні події виконується функція зворотного дзвінка по телефону, попередньо зареєстрована в компоненті. Цей метод в першу чергу використовує шаблон публікації та підписки. Саме завдяки цій техніці Node.js працює швидше, ніж інші нововведення.

1. У контексті Node.js обговорити REPL

REPL у вузлі.js означає & nbsp; Читайте, оцінюйте, публікуйте та & nbsp; Лазівку. Він ілюструє створення комп'ютерної системи, такої як оболонка Unix/Linux або домашня віконна консоль, або де будь-який вид команди може бути підписаний, і після цього система може відповісти результатом. Node.js завантажується з атмосферою REPL за замовчуванням. REPL може реалізувати наступні завдання:

1. Чому Google використовує двигун V8 для Node.js?

LTS & nbsp; способи & nbsp; Довгий & nbsp; Термін & nbsp; Опція підтримки Node.js, яка виправляє всі важливі проблеми з помилками з оновленнями безпеки, а також підвищує ефективність. Всі ці варіанти розраховані щонайменше на 18 місяців і в основному зосереджені на питаннях безпеки, а також безпеки. Зміни, внесені до варіацій LTS, обмежуються лише контролем комах, npm, покращеннями безпеки, документацією та підвищенням ефективності.

1. Що ви дізнаєтеся з зворотного дзвінка?

Зворотний дзвінок Хек також називають пірамідою руїни. Це закономірність, яка виникає з серйозних вбудованих зворотних викликів, які не читаються. Зазвичай він включає в себе кілька вбудованих функцій зворотного виклику, які ускладнюють перегляд або налагодження коду. В основному це пов'язано з неправильним виконанням асинхронних міркувань.

1. Що таке libuv?

Це багатоплатформний вузол.js допоміжна колекція, яка в основному використовується для асинхронного вводу-виводу. Спочатку він був створений для Node.js, сьогодні він додатково використовується в нашій системі, таких як pyuv, Luvit, Julia та ін. Libuv, як правило, є абстракцією від libev / IOCP на основі системи, яка надає окремим користувачам API на основі лібев. Деякі з важливих особливостей лібуву:

* Інциденти системи документів
* Повнофункціональна лазівка подій
* Асинхронні виходи TCP і UDP
* Асинхронні документи, а також процедури системної документації
* Дитячі процедури

1. Потоки в node.js

Потоки дозволяють переглядати та вдосконалювати величезні набори даних у Node.js. Ці елементи є інформаційними зборами, які виглядають як рядки, так і діапазони. Вони дозволяють постійно переглядати інформацію з ресурсу або контактувати з місцем. Існує 4 основних типи ниток в Node.js: розбірливі, записувані, дуплексні та перетворені.

1. Як саме ви переглядаєте та вбудовуєте документи в Node.js?

У Node.js існує 2 способи перегляду і реалізації документів - readFile і createReadStream. Основна відмінність між ними полягає в тому, що readFile повністю буферизований, а createReadStream частково буферизований. readFile() повертає зворотний зв'язок, коли всі документи натискаються безпосередньо в бар'єр, скорочуючи час обробки.

З іншого боку, createReadStream розглядає процедуру як випадкову колекцію, розділяючи документи безпосередньо на частини, а також повертаючись дії індивідуально. Таким чином, він дає надійний метод роботи з величезними інформаційними документами.

1. Як саме керують дочірніми рядками в Node.js?

Ви, ймовірно, припустимо, що, враховуючи, що Node.js однопотоковий, ви не можете використовувати дочірні рядки. Хоча процедура зазвичай не ідентифікує дочірні рядки, ви все одно можете реалізувати певні асинхронні завдання вводу-виводу в історії за допомогою методу generate(). Таким чином, не виключена лазівка в центрі програми. За бажанням можна складатися з компонента ChildProcess, щоб чітко використовувати цю ідею багатопотокового мовлення.

1. Чим NODE\_ENV користуюся?

Якщо ваше завдання стосується ступеня виробництва, Node.js рекламує угоду про використання змінної NODE\_ENV для її позначення. Це допомагає зробити найкращий вибір під час виконання завдання. Більше того, коли ви вкладаєте NODE\_ENV у виробництво, ваш додаток працює приблизно в 3 рази швидше.

1. Поясніть використання бар'єрного курсу в Node.js

Бар'єрний курс у Node.js зберігає необроблену інформацію, таку як цілі числа. Він являє собою завдання пам'яті за межами світу V8, вказуючи на те, що це міжнародний курс. Ви можете отримати доступ до курсу, не імпортуючи бар'єрний компонент у Node.js. Необхідність бар'єрної ставки виникає через те, що JavaScript не підходить для бінарної інформації.

1. Розділити між generate() і fork()

Ви використовуєте generate() в Node.js представити абсолютно нову процедуру. З наданим набором інструкцій ця процедура передбачає лише одну копію компонента вузла на процесорі. Враховуючи, що не створюються абсолютно нові обставини V8, метод generate() виявляється особливо цінним, коли процедура вашої дитини повертає значну кількість інформації.

З іншого боку, метод вилки () виконує абсолютно нові функції движка V8. Це вказує на те, що кілька співробітників працюють з одним і тим же кодом на основі вузла для декількох позицій.

1. Front-end, а також бек-енд просування

Удосконалення зовнішнього інтерфейсу базується на асинхронних вимогах, а також AJAX, тоді як внутрішнє вдосконалення інтерфейсу базується на дизайні веб-сервера. Раніше розмітка використовувалася, а також інтернет-мови, такі як HTML, JavaScript, CSS і т.д., для цілей пошукової оптимізації. При цьому останній використовує установку мов Python, Ruby, а також інших для резервного копіювання. Front-end просування набагато доступніше, ніж back-end просування.

1. Надайте кроки для використання функції циркуляції керування.

Код, виконаний між телефонними дзвінками з асинхронною функцією в Node.js є функцією циркуляції управління. Функцію можна пояснити наступними кроками:

* + Спочатку керуйте порядком впровадження & nbsp;
  + Після цього накопичуйте необхідну інформацію & nbsp;
  + Потім обмежте паралелізм
  + Врешті-решт, викликати наступну дію програми

1. Виклик Node.js функції синхронізації.

Node.js надає компонент Таймери, який включає в себе багато функцій для запуску коду через певний проміжок часу. Численні функції, пропоновані цим компонентом:

* setTimeout / clearTimeout & nbsp; - Використовується для програмування реалізації коду в декількох наносекундах.
* setInterval / clearInterval & nbsp; - Використовується для виконання блоку коду багато разів.
* setImmediate / clearImmediate & nbsp; - Використовується для запуску коду в кінці поточного циклу лазівки.
* process.nextTick & nbsp; - Використовується для встановлення функції зворотного виклику для активації в наступній моделі лазівки події.

1. Концепція Punycode в Node.js?

Punycode - це структура запису фрази. Він використовується для прирівнювання рядка ідентифікації Unicode (UTF-8) безпосередньо до основного рядка ідентифікації ASCII. Це необхідно, тому що імена хостів просто підтверджують посвідчення ASCII. Він використовується за замовчуванням для всіх версій після 0.6.2. Однак, якщо ви хочете використовувати його для старих варіантів, ви можете зробити це за допомогою структури фрази:

punycode = need ('punycode');

1. У чому різниця між Node.js і Ajax?

Node.js є серверним JavaScript, на відміну від Ajax, який є клієнтською інновацією. Ajax в основному призначений для оновлення або зміни веб-сторінки без необхідності оновлювати веб-сторінку. У той час як Node.js необхідний для установки програмного забезпечення веб-сервера, яке зазвичай управляється веб-серверами, а не інтернет-браузерами.

1. Чи використовує Node.js будь-який налагоджувач?

Node.js використовує основну процедуру на основі TCP на додаток до вбудованого клієнта налагодження. Якщо ви бажаєте налагодити свої документи JavaScript, ви можете скористатися вузлом налагодження [script.js | -e "script" | : ], за яким слідує & nbsp; js & nbsp; Назва документів, які ви збираєтеся налагоджувати.

1. Вкажіть Node.js коди подання.

Коди подачі - це набір розрізнених кодів, які використовуються для виконання конкретної процедури. Вона також може складатися з міжнародних речей. Нижче наведено кілька прикладів кодів подання, що використовуються в Node.js:

* Крім того
* Нерозкрите смертельне звільнення
* Внутрішня помилка виконання винятків тренера
* Смертельна помилка
* Помилка внутрішньої перевірки JavaScript

1. Чому використання блокуючих /синхронних операцій є поганою практикою в Node.js?

Node.js є однопотоковим, тому одна синхронна операція може заблокувати цикл подій, а всі інші будуть зупинені. Уявіть, що ви обслуговуєте http-запити і отримуєте запит на прочитання файлу і починаєте читати цей файл - поки не завершите його, ви більше не зможете обробити запит. Ця практика може працювати в розробці, але не на виробництві. На щастя, вузол надає синхронні та асинхронні API, такі як readFile та readFileSync.

1. Що таке стиль продовження-передачі (CPS)? Чому він такий популярний в Node.js?

CPS - це стиль, коли функція не повертає результат виконання безпосередньо, але повертає результат функції зворотного виклику, переданої як параметр. Як ви вже знаєте, асинхронне програмування є ключовим принципом в JS, тому, природно, воно широко використовується в Node. Наприклад, для операцій вводу-виводу, таких як читання/запис файлів. Але у CPS є одна серйозна проблема, дуже легко зіткнутися з проблемою зворотного пекла.

1. За що відповідає модуль кластера? Як це допомагає масштабувати програми Node.js?

Кластерний модуль - це вбудований модуль, який надає утиліти для розгалуження вашого однопотокового додатка і дозволяє повністю використовувати потужність багатоядерної системи. За допомогою цього модуля ви можете створити необмежену кількість дитячих процесів, крім того, доступна комунікація через МПК.

1. Що робитиме Node, коли черги стека викликів і циклу подій порожні?

Вузол просто закриється.

Під Вільний час запуску застосунку, Вузол автоматично запускає цикл подій, і коли цикл подій простоює, коли він не має нічого спільного, процес вимикається.

Щоб процес Node не припинився, потрібно поставити щось у черзі подій. Наприклад, коли ви запускаєте таймер або HTTP-сервер, ви говорите циклу подій, що він повинен продовжувати працювати і стежити за цими подіями.