Лабораторная работа №1

Node.js - это среда выполнения JavaScript на стороне сервера, которая позволяет разработчикам использовать JavaScript для написания серверного кода. Она состоит из следующих ключевых компонентов:

V8 JavaScript engine: это компонент, который является ядром Node.js и отвечает за выполнение JavaScript кода. Он представляет собой движок JavaScript, разработанный компанией Google, который позволяет быстро и эффективно компилировать и выполнять JavaScript код.

Libuv: это мультиплатформенная библиотека, которая обеспечивает асинхронную I/O и другие системные функции для Node.js. Она позволяет Node.js выполнять многозадачность и обрабатывать большое количество запросов одновременно.

Core modules: это набор встроенных модулей, которые доступны в Node.js без установки сторонних библиотек. Эти модули предоставляют базовую функциональность, такую как работа с файлами, сетевое взаимодействие, криптография и другие.

NPM: это менеджер пакетов для Node.js, который позволяет устанавливать и управлять сторонними библиотеками и зависимостями в проекте. Он предоставляет доступ к большому количеству библиотек, которые помогают упростить и ускорить процесс разработки.

СommonJS: это спецификация модульной системы, которая используется в Node.js для организации кода. Она позволяет разделить код на отдельные модули, которые могут иметь свою собственную область видимости и экспортировать свои функции и переменные.

В целом, Node.js состоит из нескольких ключевых компонентов, которые позволяют разработчикам писать эффективный и масштабируемый серверный код на JavaScript.

Причины популярности

Высокая скорость. JavaScript-код, который выполняется в среде Node.js, может быть в несколько раз быстрее, чем написанный на языках вроде Ruby или Python. В Node.js используется модель асинхронного программирования. Модель позволяет продолжить обработку других задач, не дожидаясь завершения передачи данных. Когда требуется выполнить операцию ввода-вывода вроде доступа к файловой системе или базе данных, Node.js не блокирует главный поток ожиданием результатов. Платформа инициирует ее выполнение и продолжает выполнять другие задачи, пока результаты предыдущей операции не будут получены.

Универсальность и гибкость. В Node.js выполняется код, который написан на JavaScript. Это означает, что frontend-разработчики, которые уже используют JavaScript в браузере, могут писать и клиентский, и серверный код на привычном языке программирования, не изучая инструмент с нуля. В Node.js можно быстро переходить на новые стандарты ECMAScript по мере их реализации. Новые возможности языка становятся доступны сразу после установки поддерживающей их версии Node.js.

Большое количество модулей и библиотек. Экосистема Node.js стремительно развивается благодаря менеджеру пакетов NPM. Он содержит более 500 000 модулей и библиотек open-source, которые находятся в свободном доступе. Также постоянно появляются новые.

Работа на движке Chrome V8. Node.js работает на JavaScript-движке V8 от Google. V8 — движок JavaScript с открытым исходным кодом, распространяемый по лицензии BSD. Он применяется в браузерах на основе Chromium. Это означает, что в Node.js использованы наработки тысяч инженеров. Движок написан на C++, имеет открытый исходный код и продвинутые библиотеки.

NVM (Node Version Manager) - это инструмент, который позволяет управлять установленными версиями Node.js на вашем компьютере. Он позволяет быстро и легко устанавливать, переключаться между различными версиями Node.js и использовать несколько версий Node.js на одном компьютере.

Вот некоторые из преимуществ использования NVM для управления версиями Node.js:

Легкость установки и обновления версий Node.js. С помощью NVM вы можете быстро и легко установить и обновить любую доступную версию Node.js.

Поддержка нескольких версий Node.js. NVM позволяет установить несколько версий Node.js на одном компьютере и быстро переключаться между ними, что полезно при разработке приложений, которые работают на разных версиях Node.js.

Удобное управление глобальными модулями. С помощью NVM вы можете установить глобальные модули для каждой версии Node.js, и эти модули будут доступны только при использовании соответствующей версии Node.js.

Быстрое и простое восстановление. NVM позволяет быстро переустановить любую версию Node.js, если что-то пошло не так или если вы хотите откатиться к предыдущей версии.

В целом, использование NVM для управления версиями Node.js позволяет легко управлять установленными версиями Node.js, использовать несколько версий Node.js на одном компьютере и установить глобальные модули для каждой версии Node.js. Это существенно облегчает разработку Node.js приложений и помогает избежать проблем совместимости между версиями.

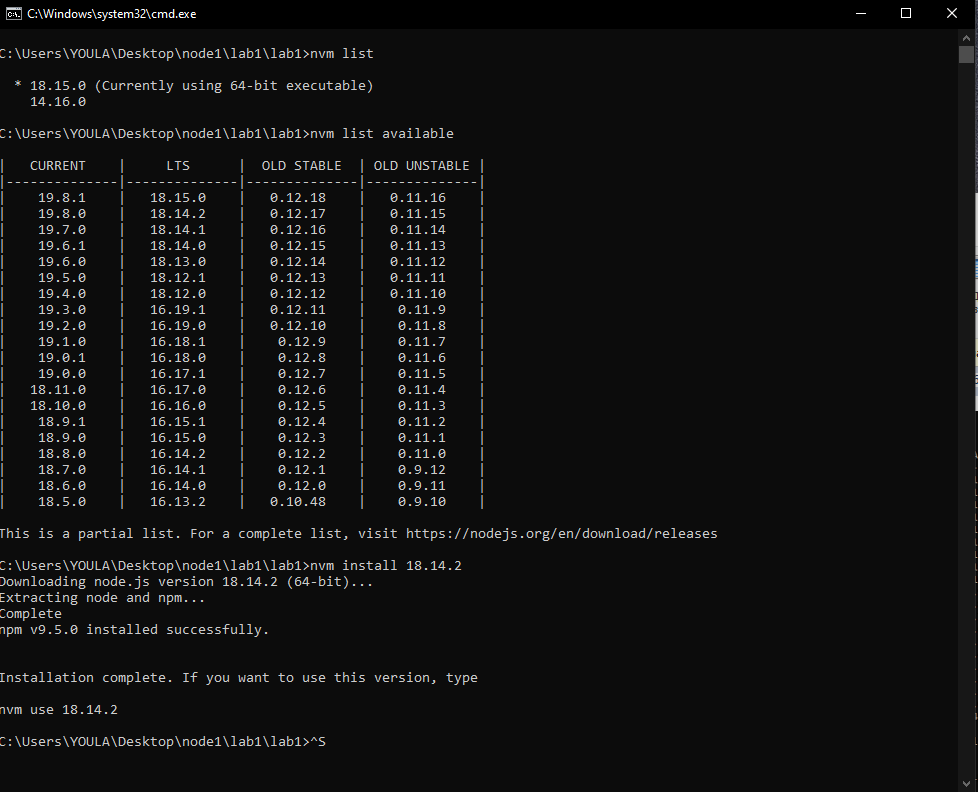


Рис.1 установка node.js с помощью nvm

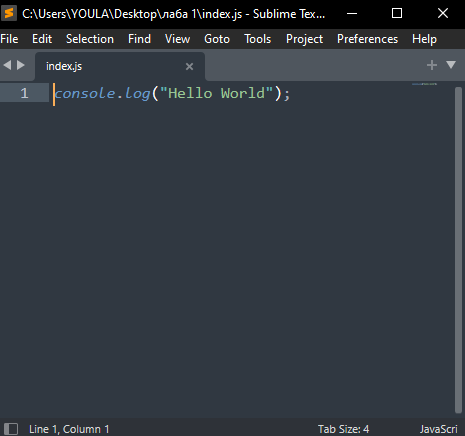


Рис.2 привет мир

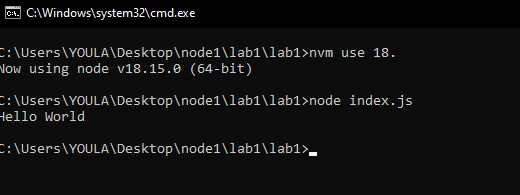


Рис.3 Запуск файла

Вывод: Node.js - это платформа для выполнения JavaScript-кода на сервере, которая позволяет разработчикам создавать высокопроизводительные и масштабируемые веб-приложения. Использование Node.js имеет несколько преимуществ:

1. Быстрое время разработки: Node.js позволяет использовать JavaScript на сервере и на клиенте, что упрощает разработку веб-приложений.
2. Высокая производительность: Node.js основан на событийно-ориентированной архитектуре, что позволяет ему обрабатывать большое количество запросов одновременно и эффективно работать с большими объемами данных.

Выполнил студент 3 курса

Группы РПИа-о20

Шангареев Н.В.