Node.js и NPM.

Node.js – это кроссплатформенная программа, представляющая собой среду исполнения JS-кода и использующая для этого JS-движок V8. Она позволяет писать приложения на языке Javascript, предоставляя API для доступа к файловой системе, к базам данных, потокам ввода/вывода, к получению и отправке сетевых запросов и т. д. Как правило, Node.js используется для создания веб-серверов. В нашей работе мы используем Node.js для возможности устанавливать npm-пакеты.

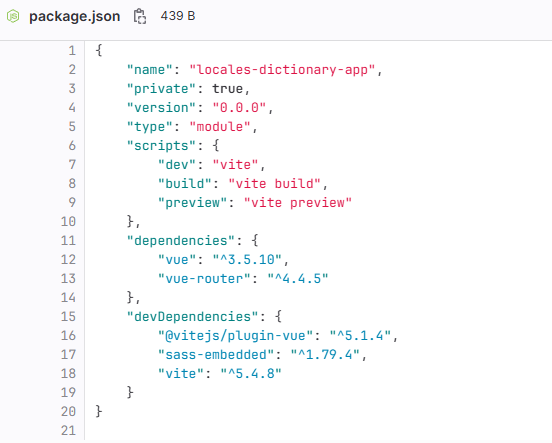
NPM - это менеджер пакетов программного обеспечения (как с открытым исходным кодом (open source), так и приватных), который поставляется в комплекте с Node.js и предназначен для управления зависимостями и модулями в проектах на базе Node.js.

На вебсайте NPM [4] представлен доступ к репозиторию NPM-пакетов – некое облачное хранилище, где вы можете осуществить поиск общедоступных пакетов, приватных пакетов (для авторизованных пользователей при определенных условиях), а также авторизоваться, ознакомиться с документацией и т. д.

Под пакетом подразумевается директория, содержащая один или несколько JS-файлов и в целом реализующая некоторый функционал. Пакет может быть использован в том числе и как библиотека.

В директории NPM-пакета обязательно должен быть файл package.json. Файл package.json содержит в себе информацию о пакете: название, версия, зависимости и т. д. Подробнее о том, что такое пакеты и модули вы можете посмотреть в документациях [1] и [2], о структуре package.json в [3] и [11].

Вот пример файла package.json одного из наших проектов:

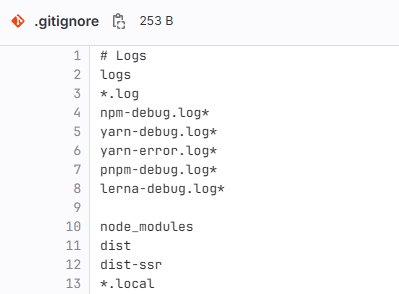


Интерфейс командной строки (CLI) NPM позволяет устанавливать, обновлять или удалять пакеты вашего проекта.

Для установки нужного пакета, можно либо указать его название и необходимую версию в поле dependencies, optionalDependencies или devDependencies в файле package.json (см. скриншот выше) и установить командой npm install, либо напрямую установить пакет командой npm install <package-name> @<version>, где версию можно не указывать (тогда поставится последняя доступная версия пакета). В последнем случае, запись об установленном пакете автоматически появится в package.json. Подробнее в документации [8].

Все пакеты, находящиеся в поле dependencies попадут в итоговую сборку вашего проекта (сборка для продакшена). То есть, после сборки ваш проект сам станет NPM-пакетом, который могут использовать другие разработчики если вы его загрузите в NPM-репозиторий (облачное хранилище). В свою очередь, пакеты из devDependencies не попадут в финальную сборку – они будут использоваться только в процессе разработки. Подробнее об этом будет разобрано в следующих темах. Что касается optionalDependencies, то туда устанавливают пакеты, которые в случае неуспешной установки через npm install, могут быть пропущены. Но для этого ваш проект должен учесть отсутствие фич, предоставляемых таким пакетом, то есть это ответственность разработчика. Подробнее в документации [9] и [10].

После установки пакетов (командой npm install), директории всех зависимостей попадают в директорию node\_modules, которая находится в корне вашего проекта и инициализируется при запуске команды npm install. Как правило, эта директория очень загружена, поэтому при работе с Git-репозиториями, она добавляется в .gitignore, чтобы не допустить ее загрузки в удаленный репозиторий.



Другие разработчики после клонирования вашего проекта в свой локальный репозиторий смогут восстановить эту директорию путем запуска команды npm install в корне проекта.

Вопросы и задания

1. Для дальнейшей работы Вам потребуется установить Node.js с помощью утилиты nvm [5]. Инструкция по установке на Linux есть на официальной странице в GitHub [6], для Windows можно скачать установщик nvm.exe по ссылке [7].

Внимание: если имя пользователя компьютера содержит пробелы, то установите nvm в корень диска (например, C:// для Windows) в папку с именем nvm (которую вы можете создать из окна установщика). Иначе это приведет к ошибке при запуске программы из командной строки, так как nvm не распознает пробелы в пути до места установки.

После установки перезапустите терминал и введите nvm -v. Если установка прошла успешно, то будет выведена версия установленного nvm. Установите последнюю версию Node.js с помощью команды nvm install node. Вы можете также установить любые другие доступные версии. Переключитесь на последнюю версию командой nvm use <version>, где <version> – желаемая версия Node.js для использования в текущий момент. Этой командой вы можете переключаться в дальнейшем на любую необходимую версию. Проверьте успешность установки командами node -v и npm -v.

Ссылки

1. <https://nodejs.org/en/learn/getting-started/an-introduction-to-the-npm-package-manager>
2. <https://docs.npmjs.com/about-packages-and-modules>
3. <https://nodejs.org/docs/latest/api/packages.html#nodejs-packagejson-field-definitions>
4. <https://www.npmjs.com/>
5. <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
6. <https://github.com/nvm-sh/nvm?tab=readme-ov-file#installing-and-updating>
7. <https://github.com/coreybutler/nvm-windows/releases>
8. <https://docs.npmjs.com/downloading-and-installing-packages-locally>
9. <https://nodejs.org/en/learn/getting-started/an-introduction-to-the-npm-package-manager>
10. <https://docs.npmjs.com/specifying-dependencies-and-devdependencies-in-a-package-json-file>
11. <https://docs.npmjs.com/cli/v11/configuring-npm/package-json>