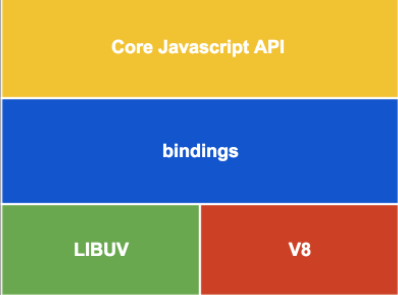
1. Перечислите основные свойства сервера **NODE.JS**
2. **NODEJS:** основные свойства:

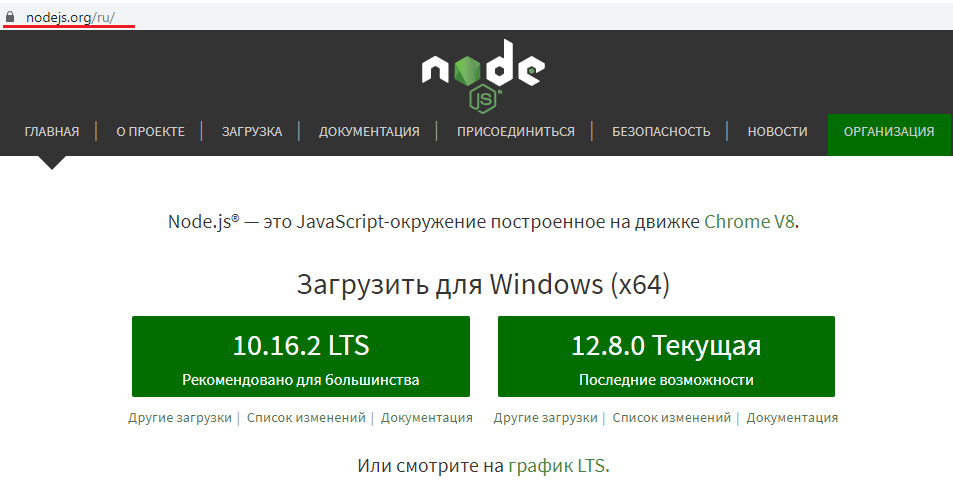
* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается свой поток, в Node.js все соединения обрабатываются в одном JS-потоке;
* **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4х одновременно);
* в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования С++ приложений;
* использует библиотеки: **V8** – библиотека V8 Engine, **libuv** – библиотека для абстрагирования неблокирующих операций ввода/вывода; **http-parser** – легковесный парсер http-сообщений (написан на C и не выполняет никаких системных вызовов); **c-ares** -библиотека для работы с DNS; **OpenSSL** – библиотека для криптографии; **zlib** – сжатие и распаковка.



* разработчик: **Райан Дал**:

****

* первая версия: **2009 г**.;
* стабильные версии: с **2015 г., Node.js 4.0.0;**
* официальный сайт:

****

* основная сфера применения: **разработка web-серверов**;
* четные версии Node.js –версии длительной поддержки (LST), нечетные версии – нестабильные версии, включающие последние разработки.

1. Что такое **npm**?

**npm** – пакетный менеджер

1. Поясните назначение HTTP-заголовка **Content-Type**.

Он сообщает получателю, **какой тип данных** содержится в теле запроса или ответа. Это позволяет браузеру, серверу или API-клиенту правильно интерпретировать и обработать содержимое. Он указывает **формат передаваемых данных.**

1. Поясните назначение функции **require**.

Функция require в Node.js — это ключевой механизм для **импорта модулей**. Она позволяет подключать внешние библиотеки, встроенные модули Node.js или собственные файлы, чтобы использовать их функциональность в текущем скрипте.

 Загружает модуль (файл или библиотеку)

 Выполняет его код (один раз)

 Возвращает объект, функцию или значение, экспортированное из этого модуля

1. Поясните понятие «**Модуль Node.js**».

**Модуль** — это **файл с JavaScript-кодом**, который инкапсулирует определённую функциональность и может быть **импортирован** в другие части приложения с помощью require.

Модули позволяют:

* Разделять код на логические части
* Повторно использовать функции, классы, переменные
* Избегать конфликтов имён и глобальных переменных
* Упрощать поддержку и масштабирование проекта

1. Поясните понятие «**Node.js built-in modules**» («**Node.js Core modules**»).

Это **предустановленные библиотеки**, написанные на JavaScript и C++, которые Node.js предоставляет "из коробки". Они оптимизированы, надёжны и поддерживаются официально.

1. Какой модуль **NODE.JS** обеспечивает работу с протоколом HTTP?

В Node.js работу с протоколом HTTP обеспечивает встроенный **модуль** http. Он входит в состав **core modules** и позволяет создавать как **HTTP-серверы**, так и **HTTP-клиенты** — без необходимости устанавливать дополнительные библиотеки.

1. Какой модуль **NODE.JS** обеспечивает работу с файловой системой?

В Node.js работу с файловой системой обеспечивает встроенный модуль fs — сокращение от *File System*. Это один из самых часто используемых **core modules**, и он позволяет читать, записывать, удалять, переименовывать и управлять файлами и папками на диске.