

ПРОГРАММИРОВАНИЕ В INTERNET

ВВЕДЕНИЕ В NODE.JS

Node.js

=

программная платформа для
разработки преимущественно
серверных приложений на
языке JS.



Основные свойства

- основан на **Chrome V8**;
- **среда** (контейнер) **исполнения** приложений на JavaScript;
- **однопоточный** (код приложения выполняется только в одном потоке, один стек вызовов);
- ориентирован на **события**;
- поддерживает механизм **асинхронности**;
- **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4-ёх одновременно).
- в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования C++ приложений;
- основная сфера применения: **разработка web-серверов**;
- **версионирование**: две ветки 18.x.x – версии длительной поддержки (LST, Long Term Support), 20.x.x – нестабильные версии, включающие последние разработки (Current);
- **документация**: <https://nodejs.org/api/>
- **Stability Index** - Deprecated (0), Experimental (1), Stable (2), Locked (3)

18.17.1 LTS

Рекомендовано для большинства

20.6.1 Текущая

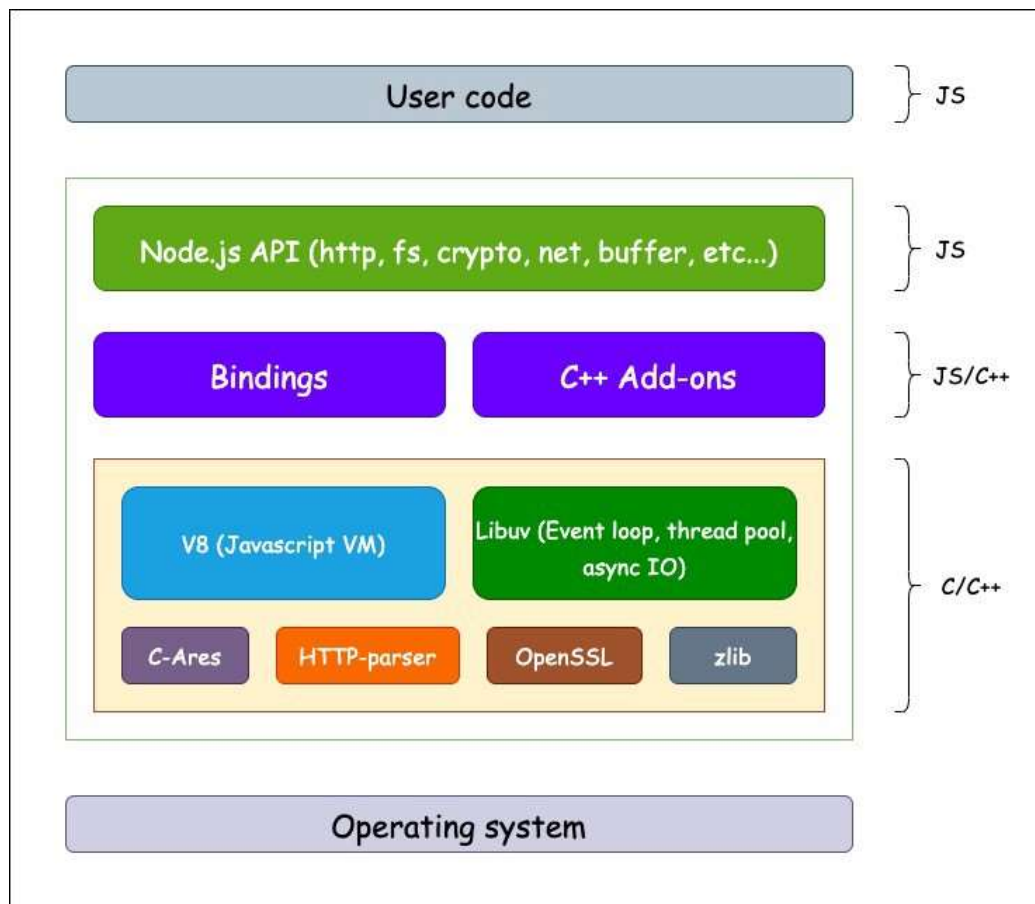
Новые возможности

Stability: 2 - Stable

Stability: 1 - Experimental

Stability: 0 - Deprecated: Use `buf.buffer` instead.

Архитектура Node.js



Использует библиотеки:

V8 – библиотека V8 Engine;

libuv – библиотека для абстрагирования неблокирующих операций ввода/вывода (представляет собой обертку над *epoll*, *kqueue*, *IOCP*);

lhttp – легковесный парсер http-сообщений (написан на С и не выполняет никаких системных вызовов);

c-ares - библиотека для работы с DNS;

OpenSSL – библиотека для криптографии;

zlib – сжатие и распаковка.

История Node.js

2009: Первая версия Node.js, создание первого варианта npm

2010: Express, Socket.io, поддержка на Heroku

2011: NPM 1.0, большие компании (LinkedIn и Uber), начали пользоваться Node.js

2012: Стабильная версия (0.8.x), быстрый рост популярности

2013: MEAN-стэк, Кoa

2014: Netflix, io.js (форк Node.js, цель создания – внедрение поддержки ES6 и ускорение развития платформы)

2015: Слияние IO.js и Node.js в 4.x.x

2016: Yarn, Node.js 6

2017: Node.js 8, HTTP/2

2018: Node.js 10, ES-модули, worker_threads

2019: OpenJS Foundation, Node.js 12,13

2020: Node.js 14,15

2021: Node.js 16,17

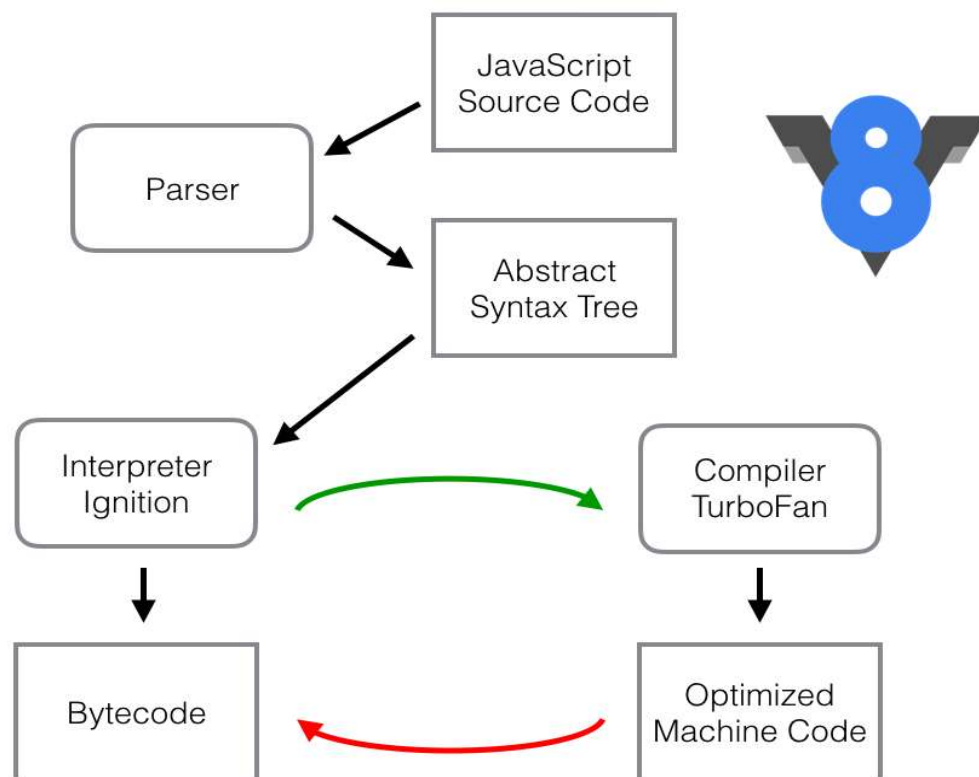
2022: Node.js 18,19

2023: Node.js 20



Райан Дал
Разработчик Node.js

Принцип работы V8

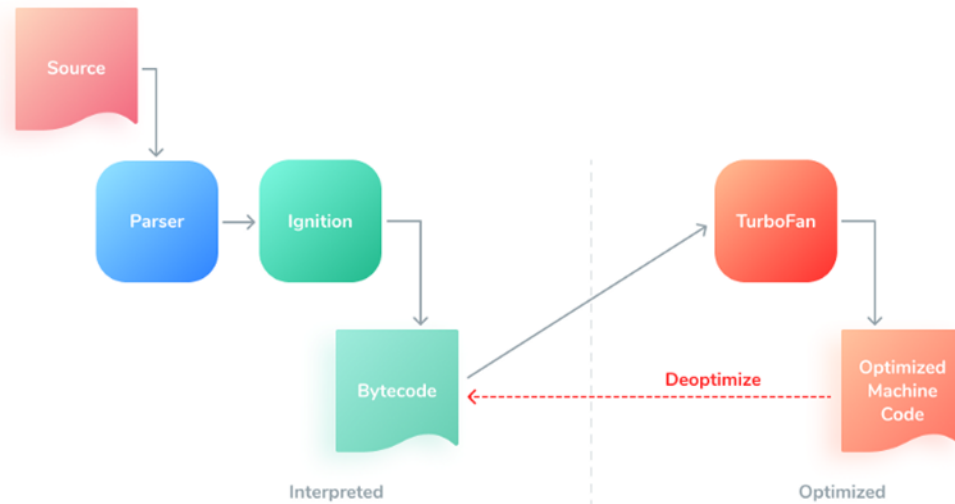


Ignition создает байт-код. В этот момент движок запускает код и собирает отзывы о типах. Чтобы он работал быстрее, байт-код может быть отправлен компилятору **TurboFan** вместе с данными обратной связи. Оптимизирующий компилятор делает на его основе определенные предположения, а затем **создает высокооптимизированный машинный код**. Если в какой-то момент одно из предположений оказывается неверным, **оптимизирующий компилятор деоптимизируется** и возвращается к интерпретатору.

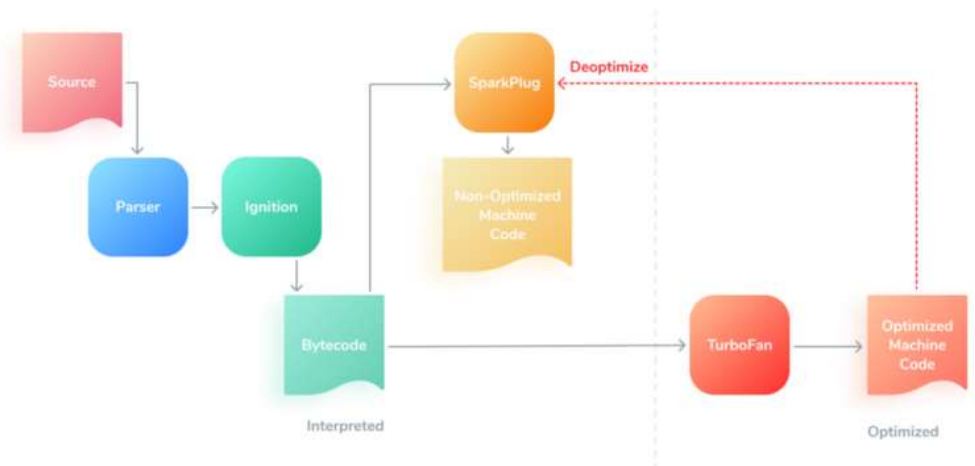
2021: **новый неоптимизирующий компилятор Sparkplug** (работает быстрее за счет компиляции не исходного кода, а байт кода + не генерирует IR (intermediate representation)).



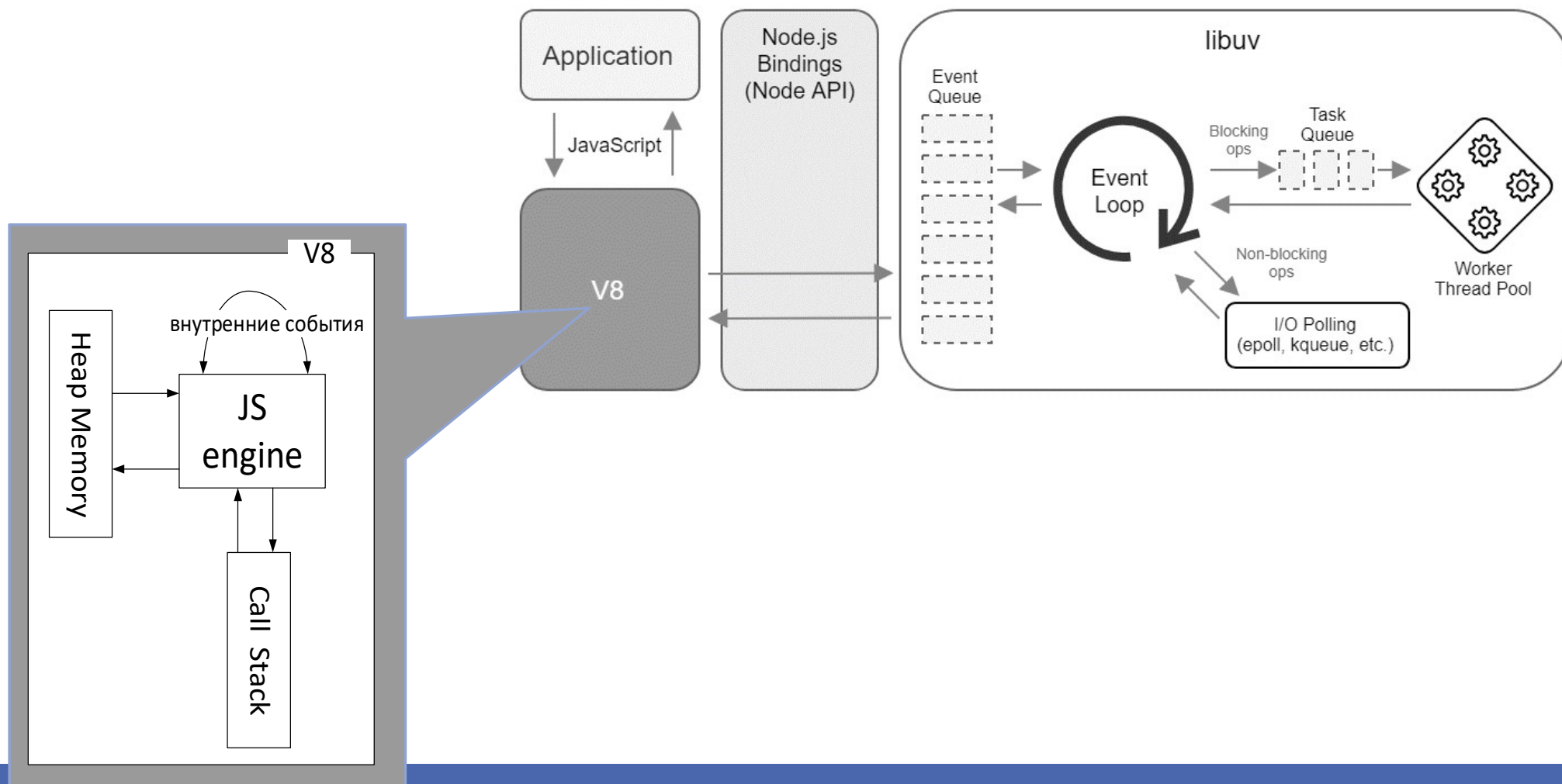
Было



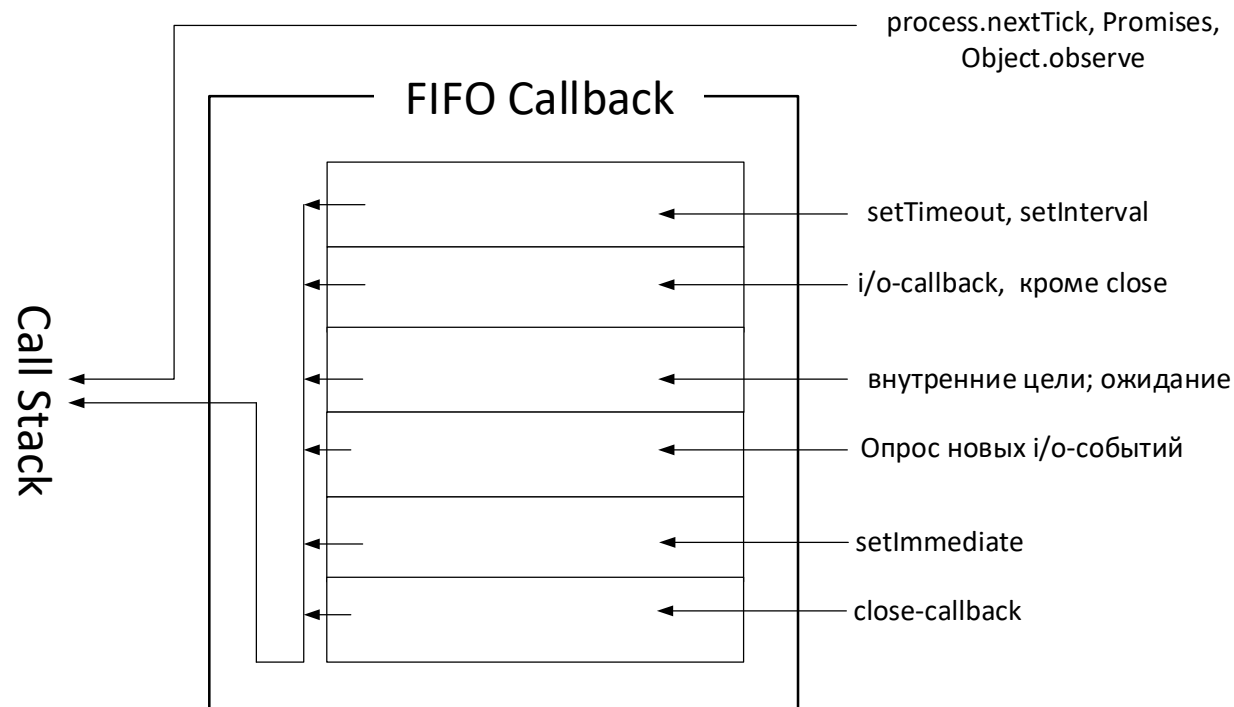
Стало



Принцип работы Node.js

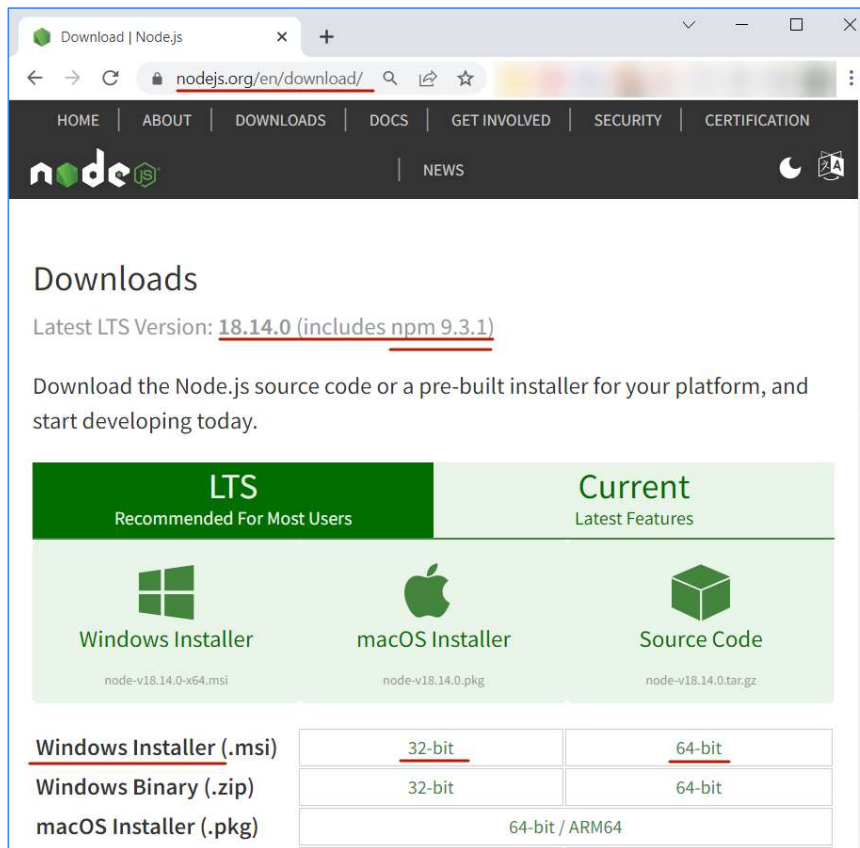


Обзор фаз event loop'a

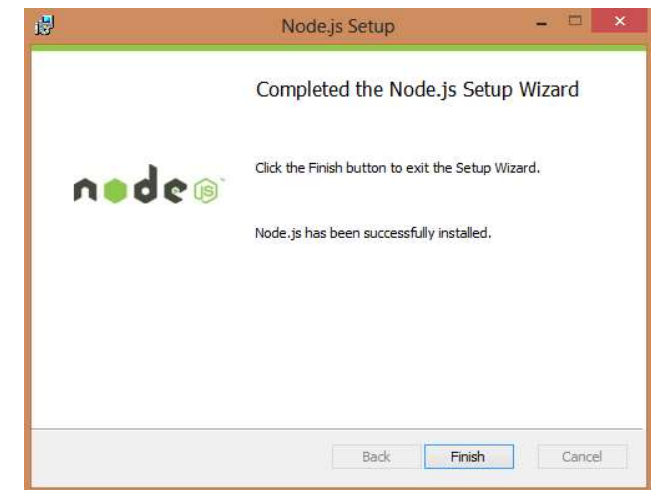
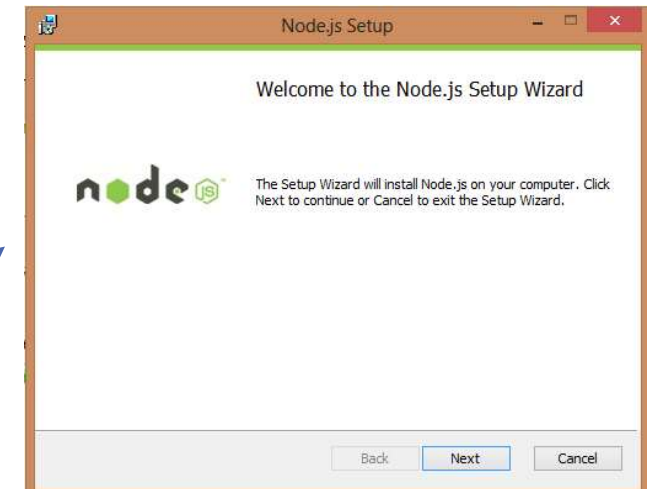


- **макронадачи** – выполняются по одной за один проход цикла (*setTimeout, setInterval, setImmediate, requestAnimationFrame, I/O, UI rendering*);
- **микронадачи** – на каждом проходе цикл выполняет все накопившееся (*process.nextTick, Object.observe, Promises*).

Установка Node.js



<https://nodejs.org/en/download/>



Проверка

Этот компьютер > Локальный диск (C:) > Program Files > nodejs >				
Имя	Дата изменения	Тип	Размер	
node_modules	31.12.2018 13:50	Папка с файлами		
node.exe	26.12.2018 6:29	Приложение	24 298 КБ	
node_etw_provider.man	02.08.2018 16:56	Файл "MAN"	9 КБ	
node_perfctr_provider.man	26.12.2018 5:54	Файл "MAN"	5 КБ	
nodevars.bat	02.08.2018 16:56	Пакетный файл ...	1 КБ	
npm	02.08.2018 16:56	Файл	1 КБ	
npm.cmd	02.08.2018 16:56	Сценарий Windo...	1 КБ	
npx	02.08.2018 16:56	Файл	1 КБ	
npx.cmd	02.08.2018 16:56	Сценарий Windo...	1 КБ	

Администратор: Командная строка

```
C:\Users\Win10_ISiT_Server>npm --version  
6.4.1
```

```
C:\Users\Win10_ISiT_Server>node --version  
v10.15.0
```

```
C:\Users\Win10_ISiT_Server>
```

Инструменты

- Visual Studio Code
- WebStorm
- Visual Studio
- Atom
- Sublime Text
- Brackets
- Браузер
- Postman
- Insomnia

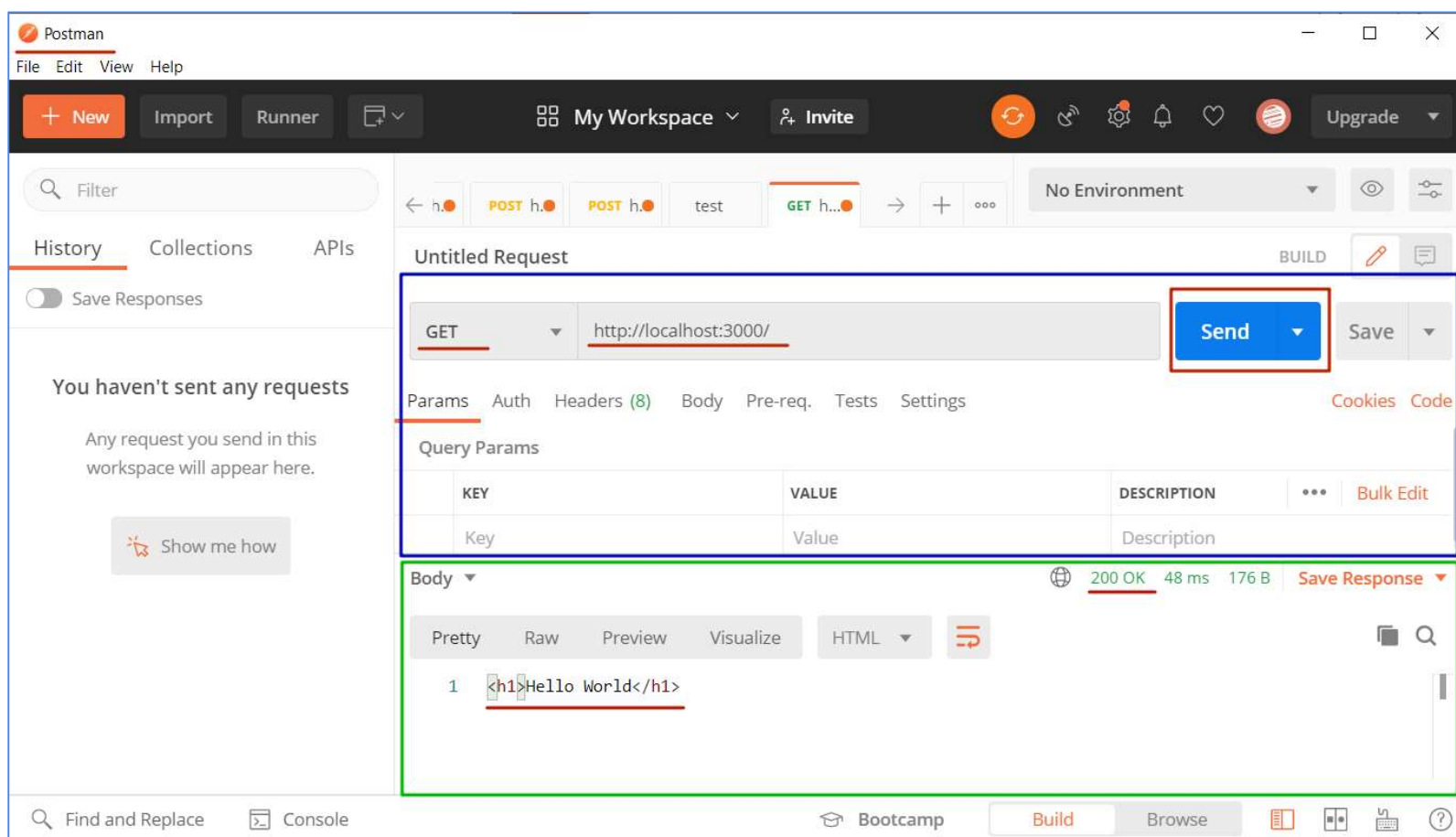
Разработка простейшего HTTP-сервер и его запуск



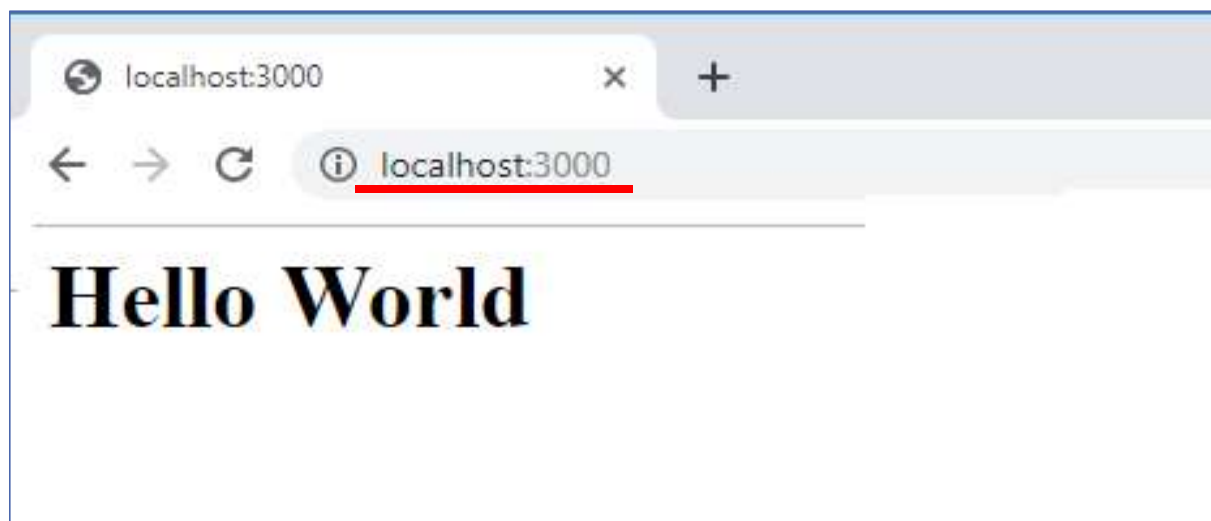
```
const http = require('http');  
  
http.createServer(function (request, response) {  
  response.writeHead(200, { 'Content-Type': 'text/html' });  
  response.end('<h1>Hello World</h1>');  
}).listen(3000, () => { console.log('Server running at http://localhost:3000/') });
```

```
PS D:\NodeJS\samples\cwp_01> node .\02-01.js  
Server runing at http://localhost:3000/  
█
```

Демонстрация работы (Postman)



Демонстрация работы (браузер)



Исследование запроса

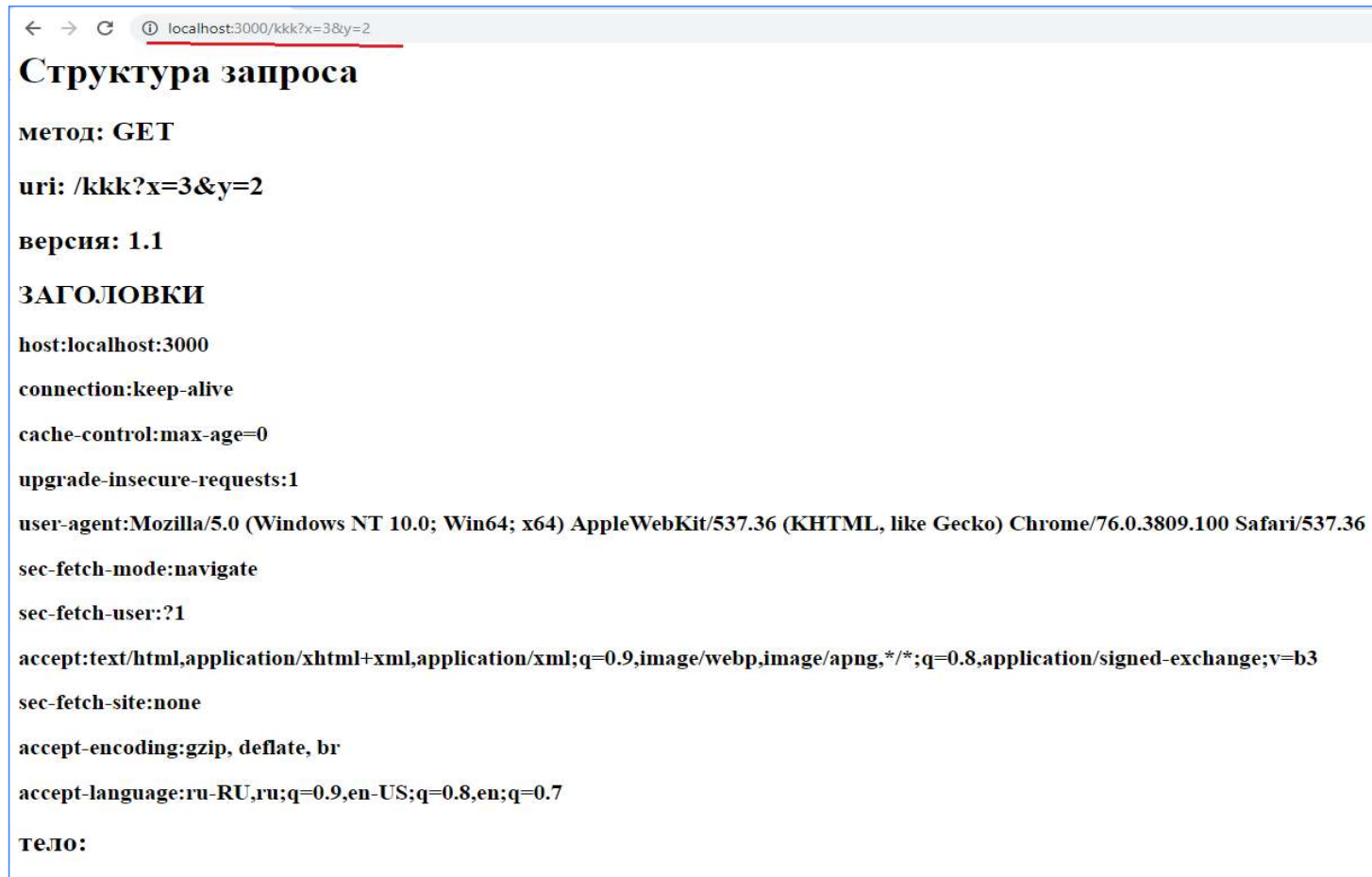
```
const http = require('http');

let h = (r) => {
  let rc = '';
  for (key in r.headers) rc += '<h3>' + key + ':' + r.headers[key] + '</h3>';
  return rc;
}

http.createServer(function (request, response) {
  let b = '';
  request.on('data', str => { b += str; console.log('data', b); });
  response.writeHead(200, { 'Content-Type': 'text/html;charset=utf-8' });
  request.on('end', () => response.end(
    '<!DOCTYPE html><html><head></head>' +
    '<body>' +
    '<h1>Структура запроса</h1>' +
    '<h2>' + 'метод: ' + request.method + '</h2>' +
    '<h2>' + 'uri: ' + request.url + '</h2>' +
    '<h2>' + 'версия: ' + request.httpVersion + '</h2>' +
    '<h2>' + 'ЗАГОЛОВКИ: ' + h(request) +
    '<h2>' + 'тело: ' + b + '</h2>' +
    '<body>' +
    '<html>'
  ));
}).listen(3000, () => console.log('Server running at http://localhost:3000/'));
```

```
PS D:\NodeJS\samples\cwp_01> node .\02-02.js
Server running at http://localhost:3000/
```


GET-запрос (браузер)



POST-запрос (Postman)

