



Node.js

JavaScript już nie tylko we front-endzie



Co to jest Node.js



Kilka pytań na które musimy sobie odpowiedzieć, m.in.

- Czym jest Node.js?
- Po co go używać?
- Dla kogo jest?
- Jak go używać?
- Jak Node.js działa?



<https://nodejs.org>

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS

FOUNDATION

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.



Download for Windows (x64)

10.15.3 LTS

Recommended For Most Users

11.14.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

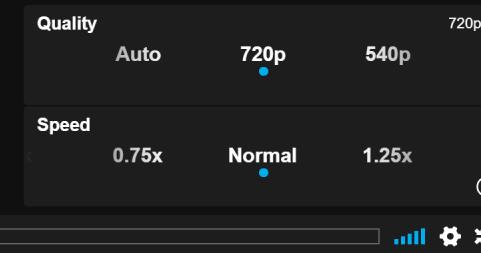
Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

Kiedyś definicja była bardziej rozbudowana

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Its package ecosystem, npm, is the largest ecosystem of open source libraries in the world.



Node.js to środowisko uruchomieniowe JavaScript zbudowane na silniku V8 (tym samym co w przeglądarce Chrome). Node.js korzysta z architektury zdarzeń (model oparty o zdarzenia) oraz nieblokujących operacji Wejścia-Wyjścia (Input/Output) co czyni go lekkim i wydajnym. Node.js korzysta z npm (node package manager) czyli menedżera rozszerzeń (pakietów), który dostarcza największą bibliotekę open-source na świecie.



Środowisko uruchomieniowe i V8 (w skrócie)

Środowisko uruchomieniowe określa sposób w jaki uruchamiany jest napisany przez nas kod. W praktyce możemy powiedzieć, że jest to wszystko co się dzieje z naszym kodem w chwili jego wykonywania. Środowiskiem uruchomieniowym dla JavaScript może być przeglądarka lub Node.js.

Głównym elementem środowiska uruchomieniowego Node.js (o którym wspomina definicja) jest silnik V8, stworzony przez Google i zaimplementowany do Node.js, który odpowiada za komplikacje JavaScript do kodu maszynowego i wykonanie tego kodu.

Node API
(wszystkie wbudowane rozwiązania)
oraz
dodatkowe moduły/pakiety

Silnik V8



Programowanie oparte na zdarzeniach (w skrócie)

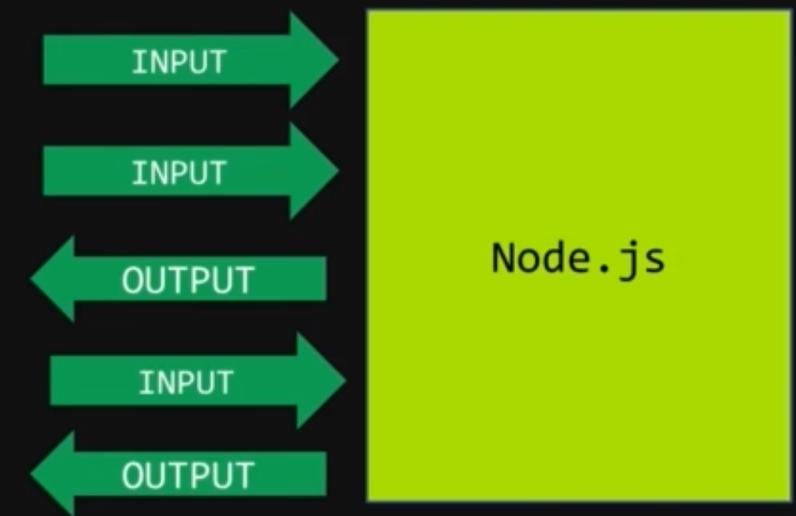
Działanie Node.js jest oparte w dużej części na zdarzeniach np. [zapytanie HTTP do serwera od klienta](#). Node.js pracuje, czyli wykonuje kod odpowiedzialny za daną czynność, w odpowiedzi na zdarzenie. Kiedy nie ma zdarzeń, nie obciąża procesora.



wróćmy do tego

nieblokujące I/O (wejście/wyjście)

Node.js, mimo że jest jednowątkowy, nie jest (tylko) synchroniczny. Korzysta bowiem z nieblokujących operacji I/O. Przykładem takich operacji może być zapytanie i odpowiedź HTTP czy odczytanie pliku.



Request od klienta stanowi nieblokujący input, tzn. że otrzymanie zapytania nie powoduje, że aplikacja zostanie zablokowana do czasu zwrócenia odpowiedzi (response, który jest operacją wyjścia).

Wieloplatformowe środowisko uruchomieniowe



- Działa po stronie serwera. Nie jest jednak serwerem (czy web serwerem)
- nie jest też frameworkiem webowym - tym jest w Node.js Express (podobnie jak Spring w Java, Ruby on Rails w Ruby, Django w Pythonie, Laravel w PHP).
- Node.js możemy uruchomić na serwerze jak i na naszej lokalnej maszynie.
- Node.js to wieloplatformowe środowisko uruchomieniowe, możesz je uruchomić na różnych systemach: Windows, Mac, Linux.

Node.js napisany w C++ (w skrócie)

Node.js to oprogramowanie stworzone w C++ z zagnieżdżonym silnikiem V8 (też napisanym w C++). Nie musisz jednak znać C++ by korzystać z Node.js.

Node.js rozszerza możliwości JavaScript. Wiele elementów node pod spodem jest C++ (piszesz JS, a wykonuje się C++).

Node używa JavaScript



Za sprawą Node.js unifikujemy (częściowo) technologie programistyczny web developmentu czyniąc ich rdzeniem JavaScript. To nie jest jedyna droga i nie koniecznie musi być najpopularniejsza w przyszłości, ale to na pewno jeden z najważniejszych i trwałych trendów. **JavaScript rules.**

JS w przeglądarce i w Node.js



Pamiętaj jednak o różnicach między wersją przeglądarkową a Node. Przeglądarki i Node to nie tylko JavaScript zgodny ze standardem ale i mnóstwo dodatkowych rozwiązań i API.

Przeglądarka (Web APIs): obiekt window, DOM

Node (Node APIs): praca z plikami, tworzenie serwera, tworzenie REST API, praca z bazą danych.

Node.js jest szybki, wydajny, skalowalny



- Tanio, szybko, dobrze, elastycznie
- od prostych i bardziej zaawansowanych stron/serwisów/sklepów interneowych (jako alternatywa dla wykorzystania PHP czy frameworków innych języków), strumieniowania danych, tworzenia serwerów API (REST API), komunikatorów, SPA (świetne jako backend dla Reacta czy Angulara), a nawet gier (z wyłączeniem tych obciążających procesor). Wszędzie tam gdzie model jest oparty na zdarzeniach i tych zdarzeń jest mnóstwo (model oparty na nieblokujących I/O jest bardzo wydajny). Wszędzie poza projektami obciążającymi procesor czy karte graficzną.
- Wsparcie największych (Amazon Web Services, Microsoft Azoure, Google Cloud Platform)