

ePortfolio: Node.js



Agenda

- Grundlagen
- Vorteile und Nachteile
- Beispiele und Anwendungsfälle
- Selbstanwendung

Grundlagen

- Javascript Framework
- Open Source
- Node.js basiert auf der V8 Engine
- npm dient der Modilverwaltung

Grundlagen

- Node.js ist single-threaded
- Es gibt eine Event-Loop und Callbacks
- Der Input und Output ist non-blocking

Grundlagen

- Hilfreiche Commands in der README.md
- Installationsanleitung in Github

Warum node.js?

Vorteile von node.js:

- Asynchrone Codeausführung
- Echtzeitfähige Standards möglich
- Hohe Performance (single thread)
- Client übergreifende Variablen (global)
- Multiplattform kompatibel

Warum node.js?

Nachteile von node.js:

- Serverneustart bei Codeupdate
- Fehler stoppt Server (Lösung: forever)
- Debugging ist komplizierter

Beispiel

Konsolenausgabe (demo1)

- Sehr einfach realisierbar
- Simpler Aufruf von `console.log(„text“)`

Beispiel

Webserver (demo3)

- Sehr einfach dank Modulen
- Sehr Ressourcenschonend
- Extrem vielfältig einsetzbar

Beispiel

Webchat (demo4)

- Echtzeit „einfach“ möglich
- Sockets als Modul
- Immer erweiterbar

Beispiel

Web Multiplayer Game

- Skalierbar
- Schnell
- Echtzeit
- Siehe browserquest

Beispiel

Offline Anwendungen

- Konvertierung von Datenformaten
- Berechnung von parallelen Abläufen
- Simple/komplexe Projekte

An die Arbeit

- Repository clonen (URL)
- Konsolenfenster öffnen (win+r, cmd)
- cd in Ordner demo(x)
- Kommando „node <datei>.js“ ausführen

Useful urls

- Nodeforum.de
- Stackoverflow
- Nodejs.org
- <https://www.tutorialspoint.com/nodejs/>