**Server Side Scripting со JavaScript**

Најпрво, да се потсетиме на некои детали за JavaScript.

Рековме дека JavaScript е програмски јазик за креирање на динамички и интерактивни веб апликации. Она што е најкарактеристично за JavaScript е тоа што JavaScript се извршува во Веб браузер. Исто така, знаеме дека Веб браузерите се опремени со таканаречени JavaScript машини кои го извршуваат JavaScript изворниот код. JavaScript машината на Google Chrome се вика V8, на Mozilla Firefox е SpiderMonkey, на Safari е JavaScriptCore, на Opera беше Caracan, а на Internet Explorer (и првата верзија на Microsoft Edge) се вика Chakra.

Практичноста на користењето на JavaScript и можностите кои ги нуди, како и идејата да се применува еден јазик во различни околини доведоа до развој на решение со кое се обезбедува извршување на JavaScript код надвор од веб браузерите. Така, во 2010 година американецот Ryan Dahl го креираше Node.js. Node.js се заснова на JavaScript машината на Google Chrome, односно машината V8, која Гугл во 2008 година ја објави како софтвер со отворен код.

Node.js му овозможува на програмерите да пишуваат скрипти во JavaScript кои се извршуваат на сервер (наместо во браузер), а со тоа динамички да генерираат веб содржини со JavaScript, слично како што тоа се прави со PHP или со други јазици за сродна намена. Развојот на Node.js ја поддржува идејата за Javascript насекаде, чија основна цел е да обезбеди унифицирање на развојот на веб апликации со еден ист програмски јазик.

Се смета дека Node.js е одлична платформа за креирање на брзи и високо скалабилни веб апликации, па од тие причини го користат и голем број познати компании како PayPal, Walmart, Uber, NASA, Netflix, Yahoo, eBay и други.

Главната карактеристика на Node.js е што овозможува asynchronous programming!

Еден најобичен пример за функционирањето на веб сервер е: клиентот побарува датотека од веб серверот, а серверот ја враќа побараната содржина назад кон клиентот.

Ако се работи со Server Side Scripting како PHP или ASP…, одговорот се праќа на следниот начин:

1. Серверот го добива барањето и се обраќа до file системот на машината за да ја добие побараната датотека.

2. Чека се додека file системот не ја прочита бараната датотека.

3. Ја испраќа содржината на датотеката до клиентот.

4. Подготвен е да прими ново барање од клиент.

Во случајот со Node.js текот на работата е на следниот начин:

1. Серверот го добива барањето и се обраќа до file системот на машината за да ја добие побараната датотека.

2. Подготвен е да прими ново барање од клиент.

3. Кога file системот ќе ја прочита бараната датотека веб серверот ја испраќа содржината на датотеката до клиентот.

Ова значи дека Node.js го елиминира чекањето за ново барање и едноставно продолжува да прифаќа нови барања од клиенти, иако претходните клиенти се уште чекаат на одговор.

Со други зборови, Node.js работи со single-threaded, non-blocking, asynchronous programming, што во суштина е исклучително ефикасно во поглед на искористување на меморијата.

Што може да се прави со Node.js?

1. Со Node.js може да се генерираат динамички веб содржини;

2. Со Node.js може да се креираат, читаат, менуваат и бришат датотеки на серверот;

3. Со Node.js може да се преземаат податоци од HTML форми;

4. Со Node.js може да се додаваат, бришат и менуваат податоци во база на податоци;

Во 2018 година Ryan Dahl издава нов runtime за JavaScript со име Deno (Дино), кој претставува runtime за JavaScript и TypeScript, исто така базиран на JavaScript машината V8 на Google, а како програмски јазик е користен Rust (наместо C и C++ кај Node.js). Deno беше објавен од страна на Dahl во 2018 година на неговиот говор со наслов "10 Things I Regret About Node.js". Ова не значи дека Node.js ќе биде веднаш заменет со Deno, но на овие часови по предметот Веб технологии ќе се работи со поновата технологија за извршување на JavaScript на сервер, односно со Deno (Deno е анаграм од Node).

Deno е едноставен, модерен и безбеден runtime за JavaScript и TypeScript.

Поточно, Deno обезбедува матична работа со TypeScript без никакви дополнителни инсталации или подесувања.

Нуди безбедност по дифолт, односно пристапот до датотеките, мрежата или околината е оневозможен се додека експлицитно не се одобри.

За инсталација има само една executable датотека без dependencies.

TypeScript е програмски јазик кој е замислен како надградба на JavaScript, а е развиен од Microsoft.

Се работи за еден таканаречен суперсет на JavaScript кој му додава одредени можности.

TypeScript е дизајниран за развој на големи апликации во JavaScript, односно секоја TypeScript апликација се претвора во JavaScript апликација за да се процесира. Ова значи дека со TypeScript во голема мера се олеснува пишувањето на апликации со JavaScript, слично како што CSS предпроцесорите (SASS и LESS) го овозможуваат тоа кога се работи со CSS.

За инсталација има различни начини, еден од нив е со користење на PowerShell (Windows), со следната наредба:

C:\> iwr https://deno.land/x/install/install.ps1 -useb | iex

Примери:

C:\deno> deno run hello.js

C:\deno> deno run hello.ts

C:\deno> deno run --allow-net webserver.ts

За користење на Deno сервер на датотеки (http file server), најпрво треба да се инсталира стандардната библиотека file\_server.ts, на следниот начин

C:\> deno install --allow-net --allow-read https://deno.land/std@0.93.0/http/file\_server.ts

File серверот кој ќе опслужува датотеки од папката C:\deno може да се активира со наредбата:

C:\deno> file\_server

во овој случај дифолт бројот на порт е 4507

Ако треба да се промени бројот на порт се пишува:

C:\deno> file\_server --port 55

Ако треба да се опслужуваат барањата од друга папка, на пример од папката public која се наоѓа во папката од која се активираме file серверот, наредбата ќе биде:

C:\deno> file\_server public --port 55