|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
|  | Summanus |  | Olimijada  Datum: 10.3.2018. |
| Uputstvo za NodeJS | | | |

## Zašto NodeJS?

NodeJS je programska platforma koja omogućava izvršavanje JavaScript-a na serveru. Potpuno je besplatan i može se izvršavati na više različitih platformi(Windows, Linux, Mac OS X). Glavna karakteristika NodeJS-a je ta što radi asinhrono, što znači da se funkcije pozivaju bez čekanja na rezultat. Ovakav način rada ne proizvodi nepotrebne zastoje u radu. NodeJS ima samo jednu nit izvršavanja ali zbog asinhronog načina rada ne dolazi do zastoja.

## Instalacija i kreiranje projekta

Instalaciju za NodeJS možete pruzeti na sledećem [linku](https://nodejs.org/en/). Uz NodeJS instalira se i NPM (Node Package Menager) koji služi za instalaciju dodatnih modula koji obezbedjuju različite funkcionalnosti. Otvorite Command Prompt i izaberite putanju na kojoj želite da se nalazi vaš NodeJS projekat. Neophodno je da na željenoj putanji u terminalu izvršite komnadu

**npm init**

Prilikom inicijalizacije projekta možete uneti podatke kao što su ime projekta, ime autora i ostalo. Neophodno je uneti “Entry point“ koji predstavlja naziv fajla koji će predstavljati NodeJS server.

## Korišćenje modula

Ažuriranje NPM-a možete uraditi sledećom komandom

**npm install npm -g**

Instalacija modula vrši se na sledeći način

**npm install <naziv modula>**

Sledeći primer predstavlja instalaciju express modula

**npm install express**

Na sličan način možete izbrisati neki od modula koji vam više nije potreban

**npm uninstall <naziv modula>**

Module koristimo tako što ih uključimo u fajl u kojem želimo da ih koristimo i to radimo na sledeći način

**var app = require(’express’);**

Dalje korišćenje zavisi od modula ali najčešće pozivamo metode karakteristične za taj modul. U tom slučaju promenljiva koja pokazuje na modul se ponaša kao objekat koji sadrži sve metode karakteristične za određeni modul. Metode pozivamo na sledeći način

**imePromenljive.NazivMetode();**

Takođe možete i samo kreirati svoje module i pritom eksportovati cele module ili čak sam određene klase ili funkcije. Fajl koji će sadržati vaš modul treba da počinje na sledeći način

**var exports = module.exports = {};**

Ukoliko želite da eksportujete određeu funkciju, to radite na sledeći način

**exports.nazivFunkcije = function( promenljive…)**

**{**

**return;  
}**

Ukoliko želite da napravite klasu koju ćete koristiti iz nekog fajla to radite na sledeći način

**Exports.NazivKlase = Class NazivKlase**

**{**

**constructor**

**{  
 }**

**function nazivMetode()**

**{  
 }**

**}**

Kao što možete videte, prilikom eksportovanja klase nije potrebno eksportovati svaku funkciju zasebno.

Uključivaje modula koje ste kreirali se razlikuje od uključivanja modula koje ste preuzeli putem npm-a

**var noviModul + require( \_\_dirname + ‘nazivFajla.js’);**

Promenljiva \_\_dirname predstavlja putanju do foldera u kome se nalazi fajl koji koristimo.

## Server

Da bi smo pokrenuli naš server u ovom slučaju koristićemo modul “express”, njegovu instance ćemo nazvati *app* I zatim pozvati funkciju *listen* koja kao argumente prima port na kojem se osluškuje i  *callback* funkciju

**var express = require(‘express’);**

**var app = express();**

**app.listen(8081, function(){**

**console.log(“Server osluškije na portu 8081!”);**

**});**

Server nije gotov u potpunosti jer svaki zahtev mora biti obrađen. Obrada zahteva izvršavaće se u *callback*-u GET funkcije. Ova funkcija prima dva argumenta, naziv stranice kojoj klijent želi da pristupi I *callback* funkciju koja će obraditi zahtev

**app.get(“/index.html”, function(req,res){**

**res.send(“Hello world!”);**

**});**

Pokretanje servera vršimo komandom **node nazivFajla**

Nakon toga u pretraživaču možete pristupiti stranici localhost:8081/index.html i prikazaće se poruka *Hello world!*

## Statički fajlovi

U NodeJS-u je neophodno napisati kod koji obrađuje svaki zahtev koji je poslat server što uključuje i zahteve za *css* I *js* fajlove kao I slike. Međutim, postoji način da se ovo olakša tako što možemo izabrati jedan folder i proglasiti ga za *static* folder. U tom slučaju za fajlove koji se nalaze u tom folderu nije neophodno pisati kod koji će obrađivati te zahteve. Ukoliko u *html* kodu postoji element koji kao **src** ima *slike/proba.png* onda je neophodno da se taj fajl nalazi na putanji *imeStatičkogFoldera/slike/proba.png*.

Da bismo omogućili ove statičke fajlove neophodno je napraviti *static* folder (koji se najčešće naziva *public*) u korenu projekta I dodati sledeći kod

**app.use(express.static(‘public’));**

Ovo uputstvo obuhvata samo najosnovnije stvari potrebne za korišćenje NodeJS-a i Express-a, više informacija možete potražiti na sledećem [linku](https://expressjs.com/).