

Node.js - סטנדרט

207

required - יצא לי module

terminal

הרבה דברים - למה שיש לי module

var foo = module.export
required('fs')

event

יש הרבה שקורה במערכת ויש לי אירועים

on - מקבל 2 פרמטרים, event ומה קורה
Event emitter - מתחבט עם כל האירועים

emit - מקבל (3 פרמטרים) event קורה אחר

on - add listener

listener count - כמה יש לי אירועים

remove listener - למסור אירוע

removeAllListeners - למסור את כל האירועים

fs - סטנדרט / fs - סטנדרט
Node.js

fs

callback - מקבל 2 פרמטרים, error או data

data - הנתונים

fs - סטנדרט / fs - סטנדרט

open - מקבל 3 פרמטרים, אירוע, אירוע, callback

writefile - מקבל 3 פרמטרים, אירוע, אירוע, callback

appendfile - מקבל 3 פרמטרים, אירוע, אירוע, callback

unlink - מקבל 2 פרמטרים, אירוע, אירוע

rename - מקבל 2 פרמטרים, אירוע, אירוע

JSON.parse - מקבל 1 פרמטר, אירוע

JSON.stringify - מקבל 1 פרמטר, אירוע

```

var myEventsModule = require('events'); //import events module, events
module is a built in module in node.js
var myEventsEmitter = new myEventsModule.EventEmitter(); //create
instance of EventEmitter

// myEventsEmitter.on("myEvent", function () { //create listener to
myEvent
//     console.log("myEvent has occurred!");
// });
// myEventsEmitter.emit("myEvent"); //print - "myEvent has occurred!"

// myEventsEmitter.on("myEvent2", function (parameter1, parameter2) {
//callback gets parameters
//     console.log("myEvent2 has occurred!");
//     console.log("here are the parameter1: " + parameter1);
//     console.log("here are the parameter2: " + parameter2);
// });
// myEventsEmitter.emit("myEvent2", 'aaa', 'bbb'); //emit myEvent2 with
sending parameters to callback

// myEventsEmitter.once("myEvent3", function () {
//     console.log("myEvent3 has occurred!");
// });
// myEventsEmitter.emit("myEvent3"); //print - myEvent3 has occurred!
// myEventsEmitter.emit("myEvent3"); //not print. because that the
listener listens to myEvent3 only one time.

myEventsEmitter.on("myEvent4", function () { //create listener to
myEvent4
    console.log("myEvent4 has occurred!");
});
var myListenerFunction = function () {
    console.log("second listener on myEvent4");
};
myEventsEmitter.on("myEvent4", myListenerFunction); //create second
listener to myEvent4
myEventsEmitter.addListener("myEvent4", function () { //create third
listener to myEvent4. "on" is the same to "addListener"
    console.log("third listener on myEvent4");
});
// console.log(myEventsEmitter.listenerCount('myEvent4')); //print - 3
- the number of the listeners of myEvent4
// myEventsEmitter.emit("myEvent4"); //run the three listeners
// myEventsEmitter.removeListener("myEvent4", myListenerFunction);
// myEventsEmitter.emit("myEvent4"); //run the first and third listener
without the second listener that removed before.
myEventsEmitter.removeAllListeners("myEvent4");
myEventsEmitter.emit("myEvent4"); //do nothing

```


Http

- יבאי את ה module המתאים לשימוש בserver.
- מה שם ה module הזה? http
- צרי שרת בשם "myFirstServer". באיזו פונקציה השתמשת? createServer
- אלו שני פרמטרים קיבל ה callback? request - response
- הגדירי שהשרת יקשיב ל port 5555.6000
- באיזו פונקציה השתמשת? listen
- הגדירי שה response שיחזור למשתמש יהיה "this is my first response".
- הריצי את הסרוור על הדפדפן. מה כתובת ה url שכתבת? localhost:5555

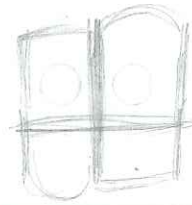
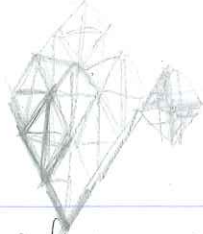
תרגיל מתכונים:

הגדירי שעבור כל מאכל שנכתב בשורת ה url (לדוגמא: "localhost:5000/cake")

יגיעו הפרטים הבאים:

- באופן דיפולטיבי יופיע שם המאכל בפורמט של טקסט פשוט.
- בשרשור של פרמטר details=receipt יופיע המתכון בפורמט של .html
- בשרשור של פרמטר details=image תופיע תמונת המאכל.
- בניתוב של "/aa" יופיע סטטוס של 404.

ionig



302

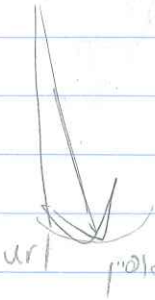
require("http") -

: ברור רק

createServer() - פונקציה שיוצרת שרת

req, res - פונקציות שמקבלות את הrequest והresponse

listener (port) - פונקציה שמקבלת את הפורט



routing - כל מה שקשור לטיפול בrequests

url - כתובת של השרת



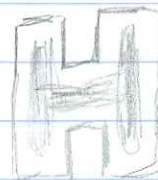
urlAddress - כתובת השרת

urlData - נתונים

urlData

pathname

query - key value



Mime type - סוג של הנתונים
text - plain

the libal Corp models to show express

207

then to show that the express is not

to content-type

middleware

Given a request, the middleware is a function that

returns a response or the next middleware

use (the middleware)

ent

Modules

- הגדירי module שבו שתי פונקציות שמקבלות פרמטרים של אורך ורוחב של צורה. פונקציה אחת מחזירה את ההיקף של הצורה והפונקציה הנוספת את השטח שלה.
- יבאי את module הזה והדפיסי את השטח וההיקף של מלבן שהרוחב שלו הוא 3 והאורך של 5.
- איך גרמת לכך שתהיה גישה לפונקציות מבחוץ?
module.export - module.p, אל תסמן את require
- איזה פונקציה מיבאת עבורך את module? require
- הגדירי module נוסף ובו מערך של חמש מספרים בדילוג של שניים: 2,4,6 וכולי.
- יבאי את ה module הזה והדפיסי את במקום הרביעי במערך.

Events

- יבאי את ה module המתאים.
מהי השורה שהשתמשת ליבוא module? require("events")
- מה ההבדל בין יבוא module מובנה בשפה לבין module שיצרתם לבד?
ה module מובנה הוא module של node, וה module שאתם יצרתם הוא module שלכם.
- הגדירי שבעת ארוע שנקרא "try-event" תודפס ההודעה: "try-event was called" - באמצעות הפונקציה "on".
- הוסיפי לארוע הזה listener נוסף, באמצעות הפונקציה "addListener". הפונקציה שנקראת תדפיס: "try-event was called. This is the second listener".
- הגדירי משתנה x שהערך שלו הוא פונקציה שמדפיסה "x was called".
הרימי את הארוע הזה ובדקי אלו ערכים מודפסים ל console. כתבי אותם כאן:
try-event was called
try-event was called. This is the second listener
- באיזה פונקציה השתמשת כדי להרים ארוע? emit
- הדפיסי את מספר המקשיבים לארוע "try-event".
באיזה פונקציה השתמשת? listener count
- הסירי את הפונקציה x מרשימת הפונקציות שמקשיבות לארוע "try-event".
באיזה פונקציה השתמשת? removeListener
- הרימי את event הזה שוב ובדקי שהפונקציה x לא הופעלה.

1/3 - ✓

2/1
מסמך
מסמך

useEvent

```
var myEvents=require("events");  
var myEventEmitter=new myEvents.EventEmitter();  
myEventEmitter.on("try-event",function(){  
  console.log("try-event was called");  
});  
myEventEmitter.emit("try-event");  
myEventEmitter.addListener("try-event",function(){  
  console.log("try-event was called. This is the second listener");  
});  
myEventEmitter.emit("try-event");  
var x=function(){  
  console.log("x was called");  
};  
myEventEmitter.addListener("try-event",x);  
myEventEmitter.emit("try-event");  
console.log(myEventEmitter.listenerCount("try-event"));  
myEventEmitter.removeListener("try-event",x);  
console.log(myEventEmitter.listenerCount("try-event"));  
myEventEmitter.emit("try-event");
```

✓ - עובד כמו שצריך

שלי
2/1

Files

- יבאי את ה module המתאים לשימוש בfiles.
- מה שם ה module הזה? fs
- צרי קובץ טקסט חדש בשם "myDetails.txt" וכתבי בו את שמך ואת שם משפחתך בצורה סינכרונית. באיזה פונקציה השתמשת? writeFileSync
- הוסיפי לקובץ את הכתובת שלך בצורה אסינכרונית על ידי פונקציה אחרת ממה שהשתמשת כדי ליצור את הקובץ. באיזה פונקציה השתמשת? appendFile
- אלו פרמטרים שלחת לה? callback, file, data
- מה מכיל הפרמטר הראשון של callback? error
- הדפיסי ל console את תוכן הקובץ בצורה אסינכרונית.
- עדכני את שם הקובץ ל"myNewDetails.txt".
- צרי module חדש ששמו יהיה "myDetailsModule".
- צרי קובץ חדש ששמו יהיה "myJsonDetails" שבו יהיו כתובים פרטיך בפורמט של Json.
- ב module תשלפי את הפרטים לתוך משתנה מתוך ה Json בצורה סינכרונית.
- המירי את ה json לאובייקט. באיזה פונקציה השתמשת? Json.parse
- יחצני את האובייקט הזה לשימוש מחוץ ל module. איך יחצנת? export
- יבאי את ה module - "myDetailsModule".
- הדפיסי את הכתובת שלך.

✓ - ע.ר. מ. צ. י. א. !

2/1 173/18 12

```
var myFilesModule = require('fs');

myFilesModule.writeFileSync("./myDetails.txt", "Sari Goldman" );
console.log("File opened and write successfully!");

myFilesModule.appendFile("./myDetails.txt", "\nDessler 10 " ,
function(err) {
    if (err) throw err;
    console.log("Saved!");
});

myFilesModule.readFile('./myDetails.txt', function (err, data) {
    if (err) {
        return console.error(err);
    }
    console.log("read file: " + data);
});

myFilesModule.rename('./myDetails.txt', 'myNewDetails.txt', function
(err) {
    if (err) throw err;
    console.log('File Renamed!');
});

var myDetailsModule = require('fs');
var objJson=myDetailsModule.readFileSync('./myJsonDetails.json');
var myObj = JSON.parse(objJson);
console.log(myObj);
myObj.export();
```

✓ - ע"י דף המסמך - המסמך
בזמן כתיבת המסמך
היה callback להשלמה.
אחרונה אחרונה בשורה כדי שלא יזכר
בזמן קריאת המסמך
אם יש צורך במסמך.

ש"י 14/3

EXPRESS

- מה שם module שעליך ליבא? express
הריצי מופע מסוג module הזה.
הגדירי פורט שעליו ירוץ השרת. מה מספר הפורט שהגדרת? 3000
- יבאי module המשמש לעבודה עם קבצים.
מה שם module הזה? fs
- פתחי קובץ (לא על ידי קוד) ששמו user.txt וכתבי בו שם פרטי ומשפחה.
- עבור בקשת GET:
קראי את השם מהקובץ. באיזה פונקציה השתמשת? readFile
האם הפונקציה הזאת היא סינכרונית או א-סינכרונית? א-סינכרונית
מה זה פונקציה א-סינכרונית? לא חוצה
- שלחי את השם מהקובץ בחזרה כresponse.
באיזה פונקציה השתמשת? send
- עבור בקשת GET url של "/img" תרד למשתמש תמונת נוף.
באיזה פונקציה השתמשת? download
- עבור url נוסף לבחירתך החזירי תגובה המשתמשת באחת מן הפונקציות שנלמדו.
באיזה פונקציה השתמשת? type
עבור איזה url? html
איזה תגובה חזרה למשתמש? <html> - P

לוי גלדמן ואלקי גרין

```
var myExpress = require("express");

var myApp = myExpress();


myApp.listen(3000, function() {
  console.log("app listening on port 3000");
});

var myFile=require("fs");

myApp.get("/", function(req, res) {
  myFile.readFile("./user.txt",function(err,data){
    if(err){
      return console.error(err);
    }
    res.send(data.toString());
  })
});

myApp.get("/image", function(req, res) {
  res.download('ISRA0022.JPG');
});

myApp.get("/html", function(req, res){
  res.type("html");
  res.send("<center><h1>Our name is Sari Goldman & Malki  
Greenwald</h1></center>");
});
```



היקשרה אלכן דסייג

! ציון מזור הרטל

ש. אולרין
2/1

EXPRESS – NODE.JS

סמני נכון או לא נכון עבור המשפטים הבאים: 12/6

- ✓ Node.js היא שפה א-סינכרונית. ✓
- ✗ modulen שמשמש עבור קבצים נקרא files. ✓
- ✗ Node.js תמיד רץ על פורט 9080. ✓
- Node.js היא שפת סרזור.
- ההרצה של node.js היא כמו הרצה של קובץ javascript רגיל.
- modulen של url נועד כדי לאפשר בקשות POST. ✓
- ✗ הפונקציה readFile היא פונקציה סינכרונית. ✓
- ✓ הפונקציה createServer מקבלת פונקציה כפרמטר.
- יבוא של module נעשה באמצעות הפונקציה require.
- אין אפשרות להמיר מ JSON לאובייקט של javascript.
- ✗ הפקודה של console.log מדפיסה לדפדפן. ✓
- exports נועד על מנת לאפשר שימוש ב module בחוץ.

עני על השאלות הבאות: 5/2

- מהו callback?
- תני דוגמא לפונקציה שמשתמשת בcallback. ✓
- מה התועלת בexpress? יש לו יכולת יאגר מאשר (זקנים?) - Node.js סכור.
- מה מכילה התיקיה node_modules?
- מהו middleware? שנגג גונני שניצלט קין הקקסה למקדג- שנגה קיני.
- למה משמשת ההגדרה של content-type?
- תני דוגמא לtype.

בזמן מצבין!

זהצחה דהמער!

2/1 פרק 3

oop - exercise

```
function Task(workerId, type, durationTime, priority) {
  this.workerId = workerId;
  var type = type;
  var _durationTime = durationTime;
  var _priority = priority;
  var getTaskDetails = function(){
    return workerId + " " + type + " " + durationTime + " " + priority;
  }
  Object.defineProperty(this, "durationTime", {
    get: function() {
      return _durationTime;
    }
  });
  Object.defineProperty(this, "priority", {
    get: function() {
      return _priority;
    },
    set: function(value) {
      if (value < 1 || value > 5) {
        console.log("error in priority number");
      }
      else{
        _priority = value;
      }
    }
  });
}
var cssTask = new Task(123, "css", 2, 4);
```

what will be the output?

- console.log(cssTask.workerId); - 123 ✓
- console.log(cssTask.type); - cssTask.type is not defined ✓
- cssTask.workerId = 456; ✓
- console.log(cssTask.workerId); 456 ✓
- cssTask.durationTime = 5; ✓
- console.log(cssTask.durationTime); 2 - לא משתנה ✓
- cssTask.priority = 3; ✓
- console.log(cssTask.priority); 3 ✓
- cssTask.priority = 7; error in priority number ✓
- console.log(cssTask.getTaskDetails()); 123 css 2 ✓
- console.log(cssTask.hasOwnProperty("name")); false ✓

9- דיוקן מצב
המשימה מציגה

1- משוב

הdurationTime לא השתנה כי זהו שדה שלא קיבל פונקציית set

עברי בלולאה והדפיסי את המאפיינים של האובייקט cssTask

דריסי את הפונקציה getDate של האובייקט Date שתחזיר את המחרוזת של השם הפרטי שלך.

```
var d = new Date();
Date.prototype.getDate = function() {
  return "sari";
}
console.log(d.getDate());
```

דריסם לדיוקן מצב
פונקציה
פרטי

```
for (var item in cssTask) {
  if (typeof cssTask[item] != 'function') {
    console.log(item + " : " + cssTask[item]);
  }
}
```