Welcome To Advanced NodeJS

Monday, October 12, 2020 10:38 AM

Advanced

Node Js

Assignment01

Monday, October 12, 2020 10:41 AM

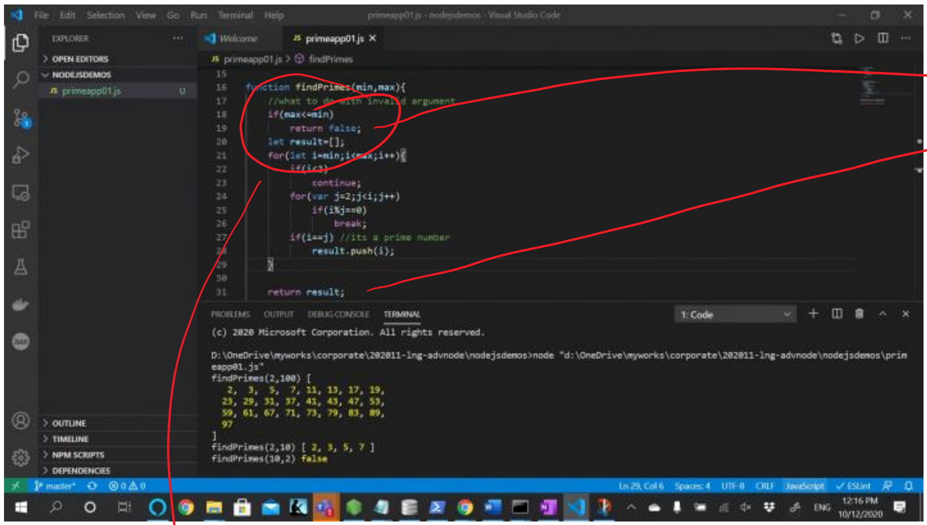
- Create a function to find and return all primes in a given min and max range
    - Example find primes between 2 and 200

- Psudo code of isPrime

```
bool isPrime(int x){

    If(x<2)
        return false;

    for(int i=2;i<x;i++)
        If(x%i==0)
            return false;

    return true;

}
```

The common problems

Monday, October 12, 2020 12:15 PM



```
function findPrimes(min,max){
    //what to do with invalid argument
    if(max<=min)
        return false;
    let result=[];
    for(let i=min;i<max;i++){
        if(i<2)
            continue;
        for(var j=2;j<i;j++)
            if(i%j==0)
                break;
        if(i==j) //its a prime number
            result.push(i);
    }

    return result;
```

```
(c) 2020 Microsoft Corporation. All rights reserved.

D:\OneDrive\myworks\corporate\202011-lng-advnode\nodejsdemos>node "d:\OneDrive\myworks\corporate\202011-lng-advnode\nodejsdemos\prim
eapp01.js"
findPrimes(2,100) [
   2,  3,  5,  7, 11, 13, 17, 19,
  23, 29, 31, 37, 41, 43, 47, 53,
  59, 61, 67, 71, 73, 79, 83, 89,
  97
]
findPrimes(2,10) [ 2, 3, 5, 7 ]
findPrimes(10,2) false
```

Returning completely different type of values

- Client is forced to check the types

Recommendation!

- **If you function returns an array, always return an array, may be an empty array when you have not value to return instead of returning false or null.**

Don't return a value to indicate an error. If possible **throw exception or any standard Mechanism to indicate error.**

Loose types?

- Javascript as loose (dynamic) types.
- But to create a consistent API we must adhere to some common denomniators

- Example a method may return

```
{
        status: 'success',
        data:[1,2,3,4]
}

Or

{
        Status:'failed',
        reason:'invalid range'
}
```

Different data

Common denom

Nodejs is Single threaded Asynchronous Programming model

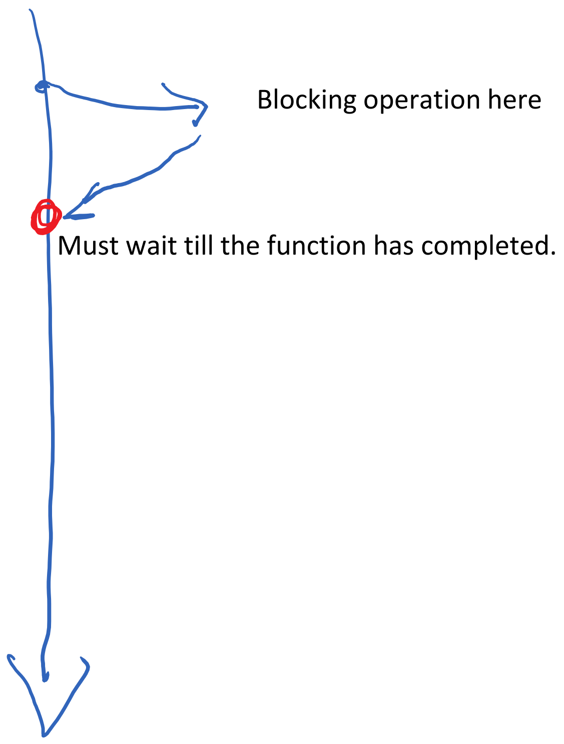
Monday, October 12, 2020 12:30 PM

NodeJS expects your functions to be async by default

- If you function is synchronous for whatever reason, it must be suffixed with the word sync

Note

- **Languages like java and C# using async suffix to mark an asynchronous function.**
- **By default functions are synchronous**
- **NodeJs expects functions to by async by default.**



Assignment 02

Monday, October 12, 2020 12:52 PM

1. Continue with Assignment01 and make the API asynchronous
2. Use Modular approach by separating business and presentation tier

Asynchrnous Programming

Monday, October 12, 2020 1:03 PM

- A general paradigm of programming, where we don't need to wait for a function to finish
  - Function returns immediately
  - Continues to work in backgournd
  - Updates the client once it finishes with the help of some kind of call back

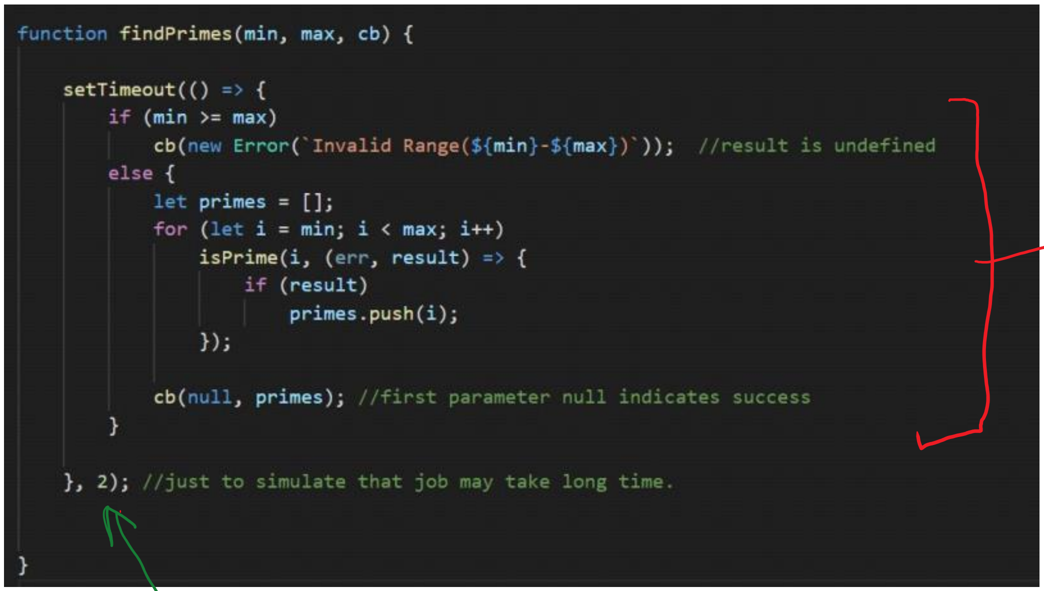
1. NodeJS callback architecture

- Nodejs expects your functions not to return using return keyword
- You pass a callback as the last parameter to your function
- Once function finishes it calls the call back
- The callback should take two parameter in order
  - Err
    - Should specificy in case of error
    - Second parameter should be null/undefined
  - Result
    - Err should be null
    - Result should contain the result

```
function findPrimesSync(min,max){

        let result=[];

        return result;
}
```

Should change to

```
function findPrimes(min,max, cb){

        let result=[];
        if(success)
                cb(null, result); //success
        else
                cb('invalid input'); //error
}
```



```
function findPrimes(min, max, cb) {

    setTimeout(() => {
        if (min >= max)
            cb(new Error(`Invalid Range(${min}-${max})`));  //result is undefined
        else {
            let primes = [];
            for (let i = min; i < max; i++)
                isPrime(i, (err, result) => {
                    if (result)
                        primes.push(i);
                });

            cb(null, primes); //first parameter null indicates success
        }

    }, 2); //just to simulate that job may take long time.

}
```

- Is running synchronously as one big chunk of code.
- Once you start, you end only after searching everything
- Not giving any other job time to work
- This is called **selfish** programming

Simulates a long running process

Cooperative Worker Pattern

- A code should allow other codes to work by taking a break
- This should allow vital UI updates and other short worker to complete

How to implement co-operative worker in our code

- Say we are finding all primes between 2 and 500000
- We may take a short break of say 10ms after every 1000 iteration.

Assignment 03

Monday, October 12, 2020 12:52 PM

1. Implement co-operative worker pattern in the findPrimes function shared with you.
    - Take short break of say 2ms after every 1000 number iteration.

2. The client shouldn't change

Expected output

- Task 2 and 3 should finish before task1