Tópicos Especiais I - Desenvolvimento aplicações móveis

NodeJs

Prof. Dr. Fábio Rodrigues de la Rocha

NodeJS

- Criado em 2009 por Ryan Dahl https://www.youtube.com/watch?v=SAc0vQCC6UQ

NodeJS

- Criado em 2009 por Ryan Dahl[https://www.youtube.com/watch?v=SAc0vQCC6UQ]
- Basicamente É Javascript rodando no servidor

NodeJS

- Criado em 2009 por Ryan Dahl https://www.youtube.com/watch?v=SAc0vQCC6UQ
- Basicamente É Javascript rodando no servidor
- Utiliza o motor do google chrome para interpretar js (motor opensource v8)

NodeJS

- Criado em 2009 por Ryan Dahl https://www.youtube.com/watch?v=SAc0vQCC6UQ
- Basicamente É Javascript rodando no servidor
- Utiliza o motor do google chrome para interpretar js (motor opensource v8)
- Tem várias bibliotecas escritas em C++/Js para realizar tarefas

NodeJS

- Criado em 2009 por Ryan Dahl https://www.youtube.com/watch?v=SAc0vQCC6UQ
- Basicamente É Javascript rodando no servidor
- Utiliza o motor do google chrome para interpretar js (motor opensource v8)
- Tem várias bibliotecas escritas em C++/Js para realizar tarefas
- I/O, envio de pacotes pela rede, implementação de protocolos

NodeJS

- Em mais detalhes é um framework para aplicações altamente concorrentes e escaláveis

NodeJS

- Em mais detalhes é um framework para aplicações altamente concorrentes e escaláveis
- Arquitetura orientada a eventos

NodeJS

- Em mais detalhes é um framework para aplicações altamente concorrentes e escaláveis
- Arquitetura orientada a eventos
- Modelo não bloqueante de I/O - Event Loop(callbacks)

NodeJS

- Em mais detalhes é um framework para aplicações altamente concorrentes e escaláveis
- Arquitetura orientada a eventos
- Modelo não bloqueante de I/O - Event Loop(callbacks)
- É single thread (em termos)

NodeJS

- Em mais detalhes é um framework para aplicações altamente concorrentes e escaláveis
- Arquitetura orientada a eventos
- Modelo não bloqueante de I/O - Event Loop(callbacks)
- É single thread (em termos)
- É javascript mas sem o DOM

NodeJS

Para que Node é utilizado ?

- Criar um servidor para receber requisições HTTP (em 10 linhas de código)
- Criar um servidor de TCP, DNS, servidor WEB
- Criar um servidor como o Gtalk para lidar com vários clientes simultâneos
- Criar todo o tipo de sistema online com vários usuários simultâneos (jogos online, aplicações de bate-papo, rastreamento, etc)

]

Servidor HTTP

```
1  const http = require ('http')
2  const server = http.createServer(function (requisicao, resposta) {
3    console.log('Recebendo uma request!')
4    resposta.end('<h1>Aqui fica o que vamos enviar para o navegador como resposta!</h1>')
5  })
6
7  server.listen(3000, function () {
8    console.log(`Servidor rodando em http://localhost:3000`)
9    console.log('Para derrubar o servidor: ctrl + c');
10 });
```

Node

- módulos, require, npm
- É JS no servidor, no cliente e hoje também nos smartfones
- É interpretada, facilita o ciclo de desenvolvimento. (in a way..)
- Diversos tutoriais, livros, podcasts, grupos de estudo, etc.

Node

Onde usar Node ?

Aplicações concorrentes com muito I/O

Node

Onde NÃO usar Node ?

Aplicações que usam muita CPU

Exemplo:sincrono

```
1 console.log('1');
2 t();
3 console.log('3');
4 function t() {
5   console.log('2');
6 }
```

assincrono

```
1 console.log('1');
2 t();
3 console.log('3');
4 function t() {
5   setTimeout(function() {
6     console.log('2');
7     },100);
8 }
```

Javascript básico

- criada em 1995, parecida com C, subestimada, a maioria das pessoas não entende...
- tipos de dados (boolean, number, string, null, undefined)
- funções são entidades de primeira classe
- arrow functions

assincrono

```
1 var vetor = [10,20,30,40,50];
2
3 vetor.map(s=>console.log(s+1)
```

Objetos

```
1  function droid() {
2   this.nome='tste';
3   this.end='sss';
4  }
5   droid.prototype.func = function ()
6   {
7       console.log('estou aqui');
8   }
9  var x = new droid()
10
11 console.log(x.nome)
12 x.func()
```

Objetos

```
1 function Droid() {}
2 var teste = new Droid();
3 Droid.prototype.retorna = function() {return this.lingua;};
4 Droid.prototype.configura = function(x) {this.lingua = x;};
5
6 teste.configura('portugues');
7 console.log('Resultado:'+ teste.retorna());
```

Objetos

```
1  function Droid() {}
2  var teste = new Droid();
3  Droid.prototype.retorna = function() {return this.lingua;};
4  Droid.prototype.configura = function(x) {this.lingua = x;};
5
6  teste.configura('portugues');
7  console.log('Resultado teste :'+ teste.retorna());
8
9  var teste2 = new Droid();
10 teste2.configura('ingles');
11 console.log('Resultado teste2:'+ teste2.retorna());
12 console.log('Resultado teste :'+ teste.retorna());
```

Controle de fluxo - if/else

```
1 var x=10;
2
3 if (x> 10) {
4   console.log('x eh maior do que 10');
5 }
6 else console.log('x em menor ou igual a 10');
```

Controle de fluxo - switch/case

```
1  var x=10;
2
3  switch (x)
4  {
5    case 0:
6    console.log(' 0');
7    break;
8    case 10:
9    console.log(' 10');
10   break;
11   default:
12   console.log('nem 0, nem 10');
13 }
```

Repetição - for

```
1 var x = [1,2,3,4,5]
2 for (var i=0; i< x.length;x++)
3   console.log(x[i])
```

Repetição - while

```
1 var x = [1,2,3,4,5]
2 var i=0;
3 while (i<x.length) {
4   console.log(x[i])
5   i++;
6 }
```

Repetição - for/in

```
1 var x = [1,2,3,4,5]
2 for (y in x)
3 {
4   console.log(y)
5 }
```

map

```
1 var x = [1,2,3,4,5]
2
3 x.map (function(x){
4   console.log(x)
5 });
```

array

```
1  var x =[];
2  var y = [1,3,4,6,7];
3
4  x.push(123);
5  x.push(332);
6  console.log('tamanho='+x.length)
7
8  var k = [];
9  k.push({nome:'Fabio Rocha', ocupacao:'professor'});
10 k.push({nome:'Maria Luisa', ocupacao:'estudante'});
11
12 for (var z=0;z<k.length;z++)
13 {
14   console.log('nome='+k[z].nome+' ocupacao='+k[z].ocupacao)
15 }
```

Operações

```
1+3+'2'
'42'


'42'== 42
true

'42'=== 42
false
```

Strings

```
1 var A = 'oi';
2 var B = 'mundo';
3 var C = A+B;
4
5 var D = 'vamos testar £{A} £{B}';
6 var E = `vamos testar ${A} ${B}`;
7
8 console.log(D);
9 console.log(E);
```

linha de comando

```
node hello2.js status port 42

// dentro do hello2.js
process.argv.forEach(function (arg) {
  console.log(arg);
});
```

linha de comando

```
1  #!/usr/bin/env node
2  /*
3  Lembre-se de tornar executvel o arquivo com 'chmod +x hello2.js'
4
5  */
6
7
8  process.argv.forEach(function (arg) {
9    console.log(arg);
10 });
```

npm

```
npm init
//install, dependencias, configurao de scripts
```