Aviso legal

Serviços para Aplicações Web e Móveis

Node

Fábio Marques (fabio@ua.pt)
Mestrado em Informática Aplicada

Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

- Ambiente de execução JavaScript de código aberto e multiplataforma
- Permite a execução de código JavaScript no lado do servidor
- Utiliza o motor de runtime JavaScript V8 (utilizado internamente pelo *Chrome*)
- Permite a utilização do JavaScript para o desenvolvimento de aplicações Web tanto do lado do cliente como do servidor

- I/O assíncrono: foi concebido para lidar eficazmente com operações de I/O assíncronas, permitindo ter várias ligações simultâneas sem bloquear a execução de outras tarefas.
- Escalabilidade: é adequado para a criação de aplicações de tempo-real e para o tratamento de várias ligações simultâneas e com um consumo de recursos relativamente baixo.
- Thread única: todas as operações são executadas numa única thread, embora o Node.js utilize um modelo de I/O assíncrono que permite a execução de várias operações em simultâneo.

- Arquitectura orientada a eventos: o Node.js utiliza um ciclo de eventos para sinalizar a ocorrência de determinado acontecimento importante.
- Multiplataforma: suporta Windows, Linux, Mac OS X, entre outros.
- NPM (Node Package Manager): gestor de pacotes que permite a instalação de módulos e bibliotecas.
- Arquitetura de microsserviços: é adequado para a criação de aplicações que são construídas como uma coleção de serviços pequenos e independentes.
- Aplicações em tempo-real: é adequado para a criação de aplicações em tempo-real, como por exemplo, aplicações de chat, jogos online, ferramentas colaborativas, entre outros.

Para aceder ao REPL basta executar o comando node na linha de comandos

Exemplo: Introduza o código que se segue no REPL:

```
function dizOla(nome) {
  console.log(nome)
}

dizOla
dizOla("Antonio");
```

Tem um conjunto de metacomandos:

- Para os listar execute o comando .help.

Exercícios:

1. Crie um conjunto de funções matemáticas (soma, subtração, multiplicação, divisão, ...) e execute-as.
2. Explore o REPL.

Apenas é necessário executar o comando node <nomefich>

Exemplo: Crie um ficheiro javascript com o código abaixo e execute-o no Node

```
"use strict";
function dizOla(nome) {
  console.log(nome)
}
dizOla("Antonio");
```

Exercícios:

1. Crie um conjunto de funções matemáticas no ficheiro operacoesMatematicas.js (soma, subtração, multiplicação, divisão, ...) e execute-as. Corra o ficheiro utilizando o Node.js.

- Criar um servidor
  - Carregar a package `http`
  - Invocar o método `createServer` para criar um objeto HTTP Server
  - Invocar o método `listen` para criar um listener no porto indicado

Exemplo:

```
"use strict";
var http = require("http");
var servidor = http.createServer( (req, res) =>{
res.writeHead(200);
  res.write("Olá Mundo!");
  res.end();
});
servidor.listen(9001);
```

- Devemos separar o nosso código em módulos para ser mais simples de gerir (ler, editar, manter, corrigir, ...)

Exemplo: (Ficheiro `pedidos.js`)

```
"use strict";
function trataPedido( req, res) {
  res.writeHead(200);
  res.write("Olá mundo!");
  res.end();
}

module.exports.trataPedido = trataPedido;
```

Ficheiro `servidor.js`

```
"use strict";
var http = require("http");
var tratador = require("./pedidos.js")
var servidor = http.createServer(tratador.trataPedido);
servidor.listen(9001);
```