Mi Página | Cursos | Universidades e instituciones | Conócenos | Soporte



Desarrollo de servicios en la nube con HTML5, Javascript y node.js

¥ Follow @miriadax 8+1

Inicio Syllabus Foro Blog

Módulos

- Modulo 0. Introducción al curso, al programa y al Sistema Operativo UNIX
- Modulo 1. Introducción a JavaScript de servidor y a node.js. Sentencias, Variables, Booleanos, Números, Strings y Funciones
- Modulo 2. Introducción a JavaScript de servidor y a node.js. Bucles, Clases predefinidas, Objetos, Propiedades y Métodos; Prototipos y Clases; Arrays; JSON; Funciones como Objetos y Cierres (Closures)
- Modulo 3. Modulos node.js; Expresiones Regulares; Eventos, Entorno de Ejecución y Concurrencia en node.js; Ficheros y Flujos
- Modulo 4. Introducción a HTTP y a los Servidores Web; Introducción a express y al Middleware Static; Introducción a REST; Aplicaciones express.js y Composición de Middlewares; Formularios GET y POST; **Parámetros Ocultos**
 - Tema 0: Transparencias del módulo
 - Tema 1. introducción a HTTP
 - Tema 1. Cuestionario obligatorio

 - Tema 2. Servidor Web

Tema 2. Cuestionario obligatorio

- Tema 3. Introducción a express.js y al middelware static
- Tema 3. Cuestionario obligatorio Tema 4. Introducción a REST
- Tema 4. Cuestionario obligatorio
- Tema 5. Aplicaciones REST con express.js
- Tema 5. Cuestionario obligatorio Tema 6. Acceso a campos de la ruta
- Tema 6. Cuestionario obligatorio
- Tema 7. Composición y ejecución de
- middlewares Tema 8. Formulario GET
- ✓ Tema 9. URL encode
- Tema 10. Formulario POST

override

 Modulo 5. Gestión de versiones de proyectos con git y GITHUB; Proyecto, Espacio de Trabajo y Versiones

(Commit); Arboles y Ramas de un

proyecto; Repositorios Remoto y

colaboración a través de GITHUB

Ejercicio P2P Obligatorio

Tema 11. Parámetro oculto y method

- Modulo 6. Proyecto Quiz I: Patrón Modelo-Vista-Controlador (MVC); generación del proyecto con expressgenerator; Primera Página y Primera Pregunta; Despliegue en la nube (Heroku)
- Datos (DB), Tablas, sequelize.js y SQLite; Despliegue en Heroku utilizando Postgres; Presentación de Listas de Quizes y Autoload Modulo 8. Proyecto Quiz III: Gestión de

Modulo 7. Proyecto Quiz II: La Base de

- Listas de Quizes, Creación, Edición y Borrado Modulo 9. Proyecto Quiz IV: Creación y
- Moderación de Comentarios a Quizes; Relaciones entre Tablas de la Base de Datos; Sesiones, Autenticación y Autorización; HTTP Seguro (HTTPS)

Modulo 4. Introducción a HTTP y a los Servidores Web; Introducción a express y al Middleware Static; Introducción a REST; Aplicaciones express.js y Composición de Middlewares; Formularios GET y POST; Parámetros Ocultos

Prueba realizada Tu resultado en el test ha sido: 100%

Has superado el test.

Tus respuestas

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

var app = express(); app.get('/coche', function (req, res){res.send('Coche');});

var express = require('express');

app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80);

- "Coche" "Casa"
- "Nada" Error-NoHayServidor
- 7 Correct

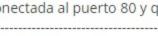
var app = express(); app.get('/coche', function (req, res){res.send('Coche');});

app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80);

- "Coche" "Casa" "Nada" Error-NoHayServidor Correct
- app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80);

Como respondera esta aplicación a la siguiente URL: http://localhost/coche/coche

- "Casa" "Nada"
- Correct



var express = require('express');

var app = express(); app.get('/coche', function (req, res){res.send('Coche');}); app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80);

"Coche"

- "Casa"
- "Nada"
- Correct

var express = require('express'); var app = express();

app.get('/coche', function (req, res){res.send('Coche');}); app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80);

"Coche"

"Casa" "Nada" Error-NoHayServidor Correct

var express = require('express'); var app = express();

app.get('/casa/*', function (req, res){res.send('Casa');}); app.get('*', function (req, res){res.send('Nada');}); app.listen(80); Como respondera esta aplicación a la siguiente URL: http://localhost:8000/casa

app.get('/coche', function (req, res){res.send('Coche');});

"Coche" "Casa"

- "Nada"
- Error-NoHayServidor Correct

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

var express = require('express');

Error-NoHayServidor

var app = express(); app.get('/coche', function (req, res){res.send('Coche');}); app.get('/casa/*', function (req, res){res.send('Casa');});

app.get('*', function (req, res){res.send('Nada');}); app.listen(80); Como respondera esta aplicación a la siguiente URL: http://localhost/coche/casa "Coche"

"Casa" "Nada"

7 Correct Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

var express = require('express'); var app = express(); app.get('/coche', function (req, res){res.send('Coche');});

app.listen(80); Como respondera esta aplicación a la siguiente URL: http://localhost:8080

app.get('/casa/*', function (req, res){res.send('Casa');});

"Nada" Error-NoHayServidor

Correct

var app = express(); app.get('/coche', function (req, res){res.send('Coche');}); app.get('/casa/*', function (req, res){res.send('Casa');});

"Coche" "Casa"

"Nada" Error-NoHayServidor

Correct

anterior

2012-2015 Miríada X

Telefónica Educación Digital

Aviso legal

Política de cookies

Telefonica

Santander

Política de privacidad

uni>ersia

Siguiente 2

Como respondera esta aplicación a la siguiente URL: http://localhost/casa/coche Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto: var express = require('express');

Como respondera esta aplicación a la siguiente URL: http://localhost

var express = require('express'); var app = express(); app.get('/coche', function (req, res){res.send('Coche');});

conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express

"Coche" Error-NoHayServidor

conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express

Como respondera esta aplicación a la siguiente URL: http://localhost:12/coche

Error-NoHayServidor

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

Como respondera esta aplicación a la siguiente URL: http://localhost/casa/casa

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto:

app.get('*', function (req, res){res.send('Nada');});

"Coche" "Casa"

Supongamos que en nuestro ordenador local arrancamos la siguiente aplicación express conectada al puerto 80 y que no hay ningún otro servidor en ningún otro puerto: var express = require('express');

app.get('*', function (req, res){res.send('Nada');}); app.listen(80); Como respondera esta aplicación a la siguiente URL: http://localhost/coche