Curso de NodeJS

Unidad Didáctica 05: API Rest

Ayuntamiento de Vitoria-Gasteiz
Vitoria-Gasteizko Udala

http://cursosdedesarrollo.com/

Índice de contenidos

- Introducción
- Servidor Node
- Rutas en Express
- Creación de Controladores
- Restify
- Conclusiones

http://cursosdedesarrollo.com/

Introducción

¿Para qué nos sirve un API Rest?

http://cursosdedesarrollo.com/

Introducción

Un API Rest nos va a permitir gestionar las peticiones que vayan a manejar los datos de la aplicación independientemente de donde se vayan a visualizar esos datos

http://cursosdedesarrollo.com/

Introducción

Un API Rest se va a consultar desde aquellos clientes Rest que necesiten acceder a la información que gestiona el API

http://cursosdedesarrollo.com/

Introducción

Los clientes pueden ser otras Aplicaciones Web o Clientes Javascript que quieran consultar datos

http://cursosdedesarrollo.com/

Servidor Node

El Servidor Node nos permite aceptar peticiones web
y ofrecer una determinada respuesta

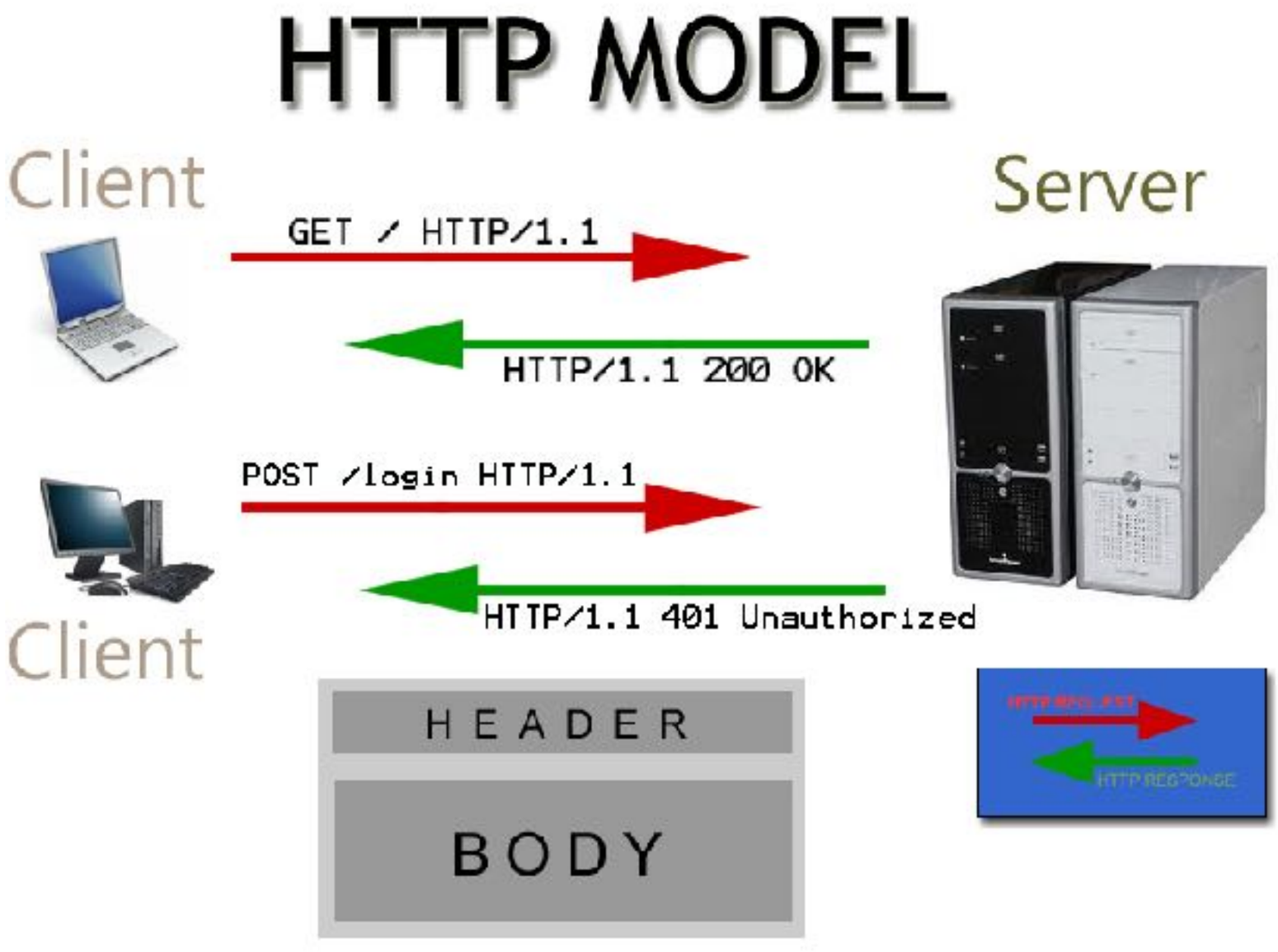
http://cursosdedesarrollo.com/

Servidor Node

El servidor funciona con el protocolo HTTP o HTTPs por lo que debe soportar las distintas combinaciones de peticiones HTTP

http://cursosdedesarrollo.com/

Servidor Node



http://cursosdedesarrollo.com/

Servidor Node

Los principales datos que se envían con la petición son:

URL

Método HTTP

Parámetros

http://cursosdedesarrollo.com/

Servidor Node

Dependiendo de los datos de la petición nuestra aplicación node deberá presentar un dato u otro

http://cursosdedesarrollo.com/

Servidor Node

Se puede manejar la URL con el módulo url

```
var url = require('url');
```

http://cursosdedesarrollo.com/

Servidor Node

Éste módulo permite gestionar la url de la petición

var pathName = url.parse(petic.url).pathname;

http://cursosdedesarrollo.com/

Servidor Node

Dependiendo de la ruta podríamos redirigir a distintos apartados dentro del código

http://cursosdedesarrollo.com/

Rutas en Express

Dentro del Framework de Express disponemos de una gestión de rutas

http://cursosdedesarrollo.com/

Rutas en Express

Veamos cómo se gestiona de base una app express

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

var app=express();

    app.get('/', function (req, res) {

      res.send('Hello World!');

    });

    app.listen(3000, function () {

      console.log('Example app listening on port 3000!');

    });
```

http://cursosdedesarrollo.com/

Rutas en Express

Como se puede ver se gestiona una ruta mediante el método get del objeto app

En el método se define la ruta de entrada y la función que la maneja

http://cursosdedesarrollo.com/

Rutas en Express

Existen funciones para todos los métodos HTTP:

get, post, put, delete, etc…

all representa a todos los métodos

http://cursosdedesarrollo.com/

Rutas en Express

Dentro de la definición de la ruta se pueden utilizar expresiones regulares para el acceso a un determinado método:

http://expressjs.com/es/guide/routing.html

Rutas en Express

Se puede gestionar distintas funciones asociadas a la gestión de una petición, se pasan como parámetros a la función, get por ejemplo.

También se pueden pasar como un array

http://cursosdedesarrollo.com/

Rutas en Express

También se puede utilizar el método next() para que se pase de una función manejada a la siguiente

http://cursosdedesarrollo.com/

Rutas en Express

Este tipo de encadenamiento de funciones permite el uso de filtros en las peticiones ya que van pasando de una función a otra

http://cursosdedesarrollo.com/

Rutas en Express

Dentro de las funciones de manejo se pueden controlar los siguientes métodos de respuesta:

Método	Descripción
res.download()	Solicita un archivo para descargarlo.
res.end()	Finaliza el proceso de respuesta.
res.json()	Envía una respuesta JSON.
res.jsonp()	Envía una respuesta JSON con soporte JSONP.
res.redirect()	Redirecciona una solicitud.
res.render()	Representa una plantilla de vista.
res.send()	Envía una respuesta de varios tipos.
res.sendFile	Envía un archivo como una secuencia de octetos.
res.sendStatus()	Establece el código de estado de la respuesta y envía su representación de serie como el cuerpo de respuesta.

http://cursosdedesarrollo.com/

Rutas en Express

En Express tenemos también a nuestra disposición la posibilidad de utilizar un route

http://cursosdedesarrollo.com/

Rutas en Express

El route permite la agrupación de distintas funciones manejadas bajo una misma ruta

http://cursosdedesarrollo.com/

Rutas en Express

```
app.route('/book')
  .get(function(req, res) {
    res.send('Get a random book');
  })
        .post(function(req, res) {
          res.send('Add a book');
        })
        .put(function(req, res) {
          res.send('Update the book');
        });
```

http://cursosdedesarrollo.com/

Rutas en Express

También tenemos acceso a un router que permite agrupar varias rutas bajo otra ruta principal

http://cursosdedesarrollo.com/

Rutas en Express

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

var router = express.Router();

// middleware that is specific to this router

router.use(function timeLog(req, res, next) {

  console.log('Time: ', Date.now());

    next();});

// define the home page route

router.get('/', function(req, res) {

  res.send('Birds home page');});

// define the about route

router.get('/about', function(req, res) {

  res.send('About birds');});

module.exports = router;
```

http://cursosdedesarrollo.com/

Rutas en Express

```
var birds = require('./birds');

...

app.use('/birds', birds);
```

http://cursosdedesarrollo.com/

Creación de Controladores

Los controladores nos van a permitir la asociación de funcionalidades agrupadas con respecto a las rutas definidas

http://cursosdedesarrollo.com/

Creación de Controladores

authorsController.js

```
// Display list of all Authors

exports.author_list = function(req, res, next) {

    res.send('NOT IMPLEMENTED: Author list');

};
```

http://cursosdedesarrollo.com/

Creación de Controladores

index.js

```
var author_controller = require(‘../controllers/
authorController');

/* GET request for list of all Authors. */

router.get('/authors', author_controller.author_list);
```

http://cursosdedesarrollo.com/

Creación de Controladores

Referencia de Controladores:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes

http://cursosdedesarrollo.com/

Restify

Restify es un motor de servidor usado principalmente para la construcción de servicio rest rápidos

http://restify.com/

Restify

```
var restify = require('restify');

function respond(req, res, next) {
  res.send('hello ' + req.params.name);
  next();
}

var server = restify.createServer();

server.get('/hello/:name', respond);

server.head('/hello/:name', respond);

server.listen(8080, function() {
  console.log('%s listening at %s', server.name, server.url);
});
```

http://cursosdedesarrollo.com/

Restify

Las rutas funcionan de una manera muy parecida a como la hacen en NodeJS o Express

http://cursosdedesarrollo.com/

Restify

```
function send(req, res, next) {
   res.send('hello ' + req.params.name);
   return next();
 }
server.post('/hello', function create(req, res, next) {
   res.send(201, Math.random().toString(36).substr(3, 8));
   return next(); });
server.put('/hello', send);
server.get('/hello/:name', send);
server.head('/hello/:name', send);
server.del('hello/:name', function rm(req, res, next) {
   res.send(204);
   return next();});
```

http://cursosdedesarrollo.com/

Restify

También se pueden encadenar funciones

http://cursosdedesarrollo.com/

Restify

```
server.get(
    '/foo/:id',
    function(req, res, next) {
        console.log('Authenticate');
        return next();
    },
    function(req, res, next) {
        res.send(200);
        return next();
    }
);
```

http://cursosdedesarrollo.com/

Conclusiones

http://cursosdedesarrollo.com/

Datos de Contacto

http://www.cursosdedesarrollo.com
david@cursosdedesarrollo.com

http://cursosdedesarrollo.com/

Licencia

David Vaquero Santiago

http://cursosdedesarrollo.com/