UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 2: Express

Profesor: Ing. Leandro Rodolfo Gil Carrano
Email: leangilutn@gmail.com

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

API REST

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

REST cambió por completo la ingeniería de software a partir del 2000. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por Roy Fielding, el padre de la especificación HTTP y uno los referentes internacionales en todo lo relacionado con la Arquitectura de Redes, en su disertación 'Architectural Styles and the Design of Network-based Software Architectures'. En el campo de las APIs, REST (Representational State Transfer- Transferencia de Estado Representacional) es, a día de hoy, el alfa y omega del desarrollo de servicios de aplicaciones.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software. Twitter, YouTube, los sistemas de identificación con Facebook… hay cientos de empresas que generan negocio gracias a REST y las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST.

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

API REST
Características

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento él identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

- Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

- Uso de hipermedios: hipermedia es un término acuñado por Ted Nelson en 1965 y que es una extensión del concepto de hipertexto. Ese concepto llevado al desarrollo de páginas web es lo que permite que el usuario puede navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones de proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

API REST
Ventajas

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

•Separación entre el cliente y el servidor: el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.

•Visibilidad, fiabilidad y escalabilidad. La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el front y el back y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

- La API REST siempre es independiente del tipo de plataformas o lenguajes: la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

A continuación vemos un ejemplo de request y response con la api de Mercado Libre:

Puedes consultar qué modos de envío tiene habilitado el usuario de la siguiente manera:

```
GET https://api.mercadolibre.com/users/:user_id?access_token=
```

Esto devolverá gran cantidad de información sobre el usuario autenticado, incluido el atributo shipping_modes.

Respuesta:

```
"shipping_modes":[
        "custom",
        "not_specified",
        "me1",
        "me2",

]
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Express

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – API REST

Sin duda el éxito de express radica en lo sencillo que es usarlo, y además abarca un sin número de aspectos que muchos desconocen pero son necesarios.

De entre las tantas cosas que tiene este framework podemos destacar:

- Session Handler
- 11 middleware poderosos así como de terceros.
- cookieParser, bodyParser…
- vhost
- router

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Primeros Pasos

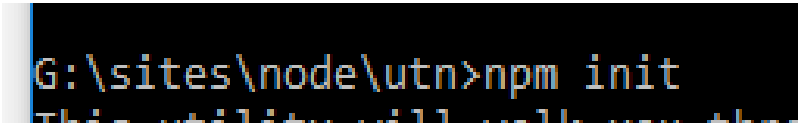
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Express primeros pasos

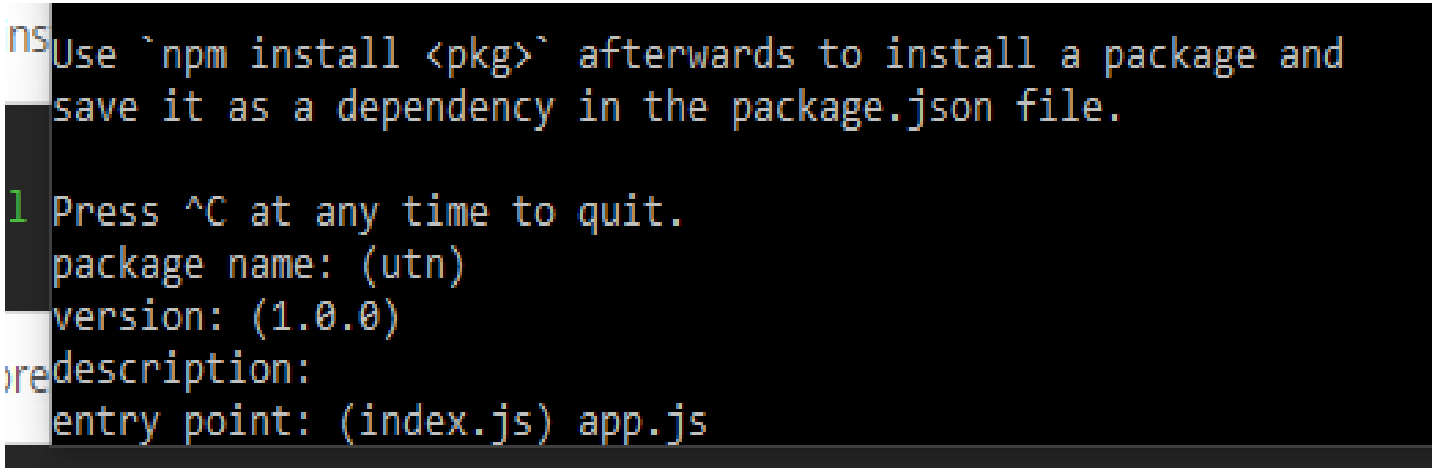
Para instalar express debemos haber instalado correctamente node js (unidad anterior)

Abrimos la consola y seguimos los siguientes pasos:

•Nos ubicamos en el directorio en el cual crearemos nuestra aplicación express y ejecutamos **npm init**



Introducimos el nombre deseado para el archivo principal, en este caso **app.js**



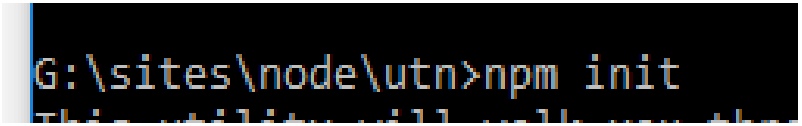
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Express primeros pasos

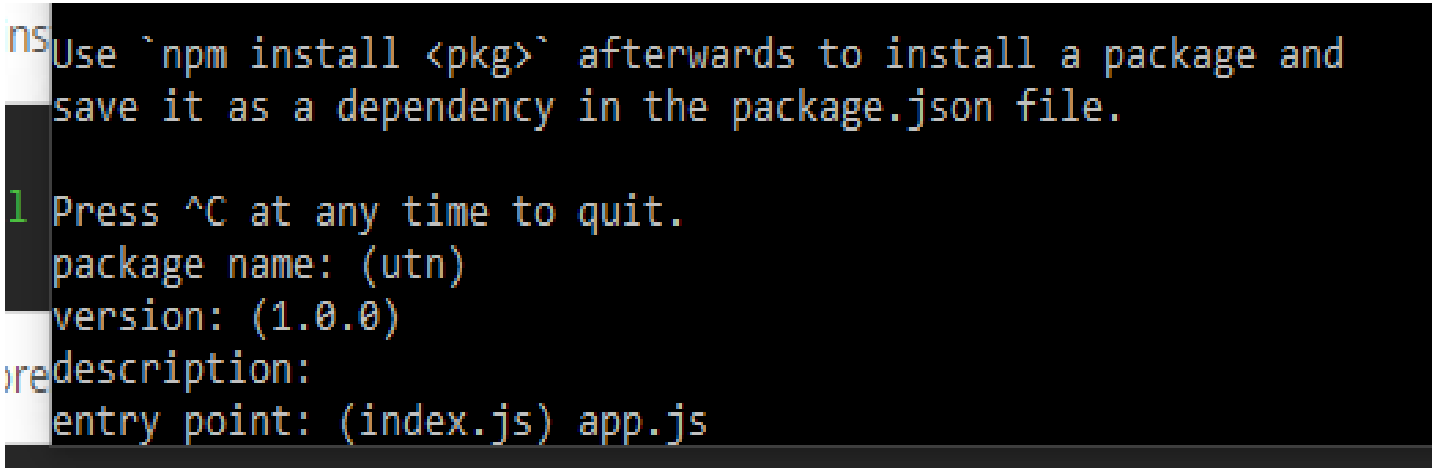
Para instalar express debemos haber instalado correctamente node js (unidad anterior)

Abrimos la consola y seguimos los siguientes pasos:

•Nos ubicamos en el directorio en el cual crearemos nuestra aplicación express y ejecutamos **npm init**



Introducimos el nombre deseado para el archivo principal, en este caso **app.js**



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Generator

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Express primeros pasos

Para instalar esta herramienta debemos ejecutar:

npm install express-generator -g

```
G:\sites\node\utn>npm install express-generator -g
```

Con el comando **express –h** podemos visualizar las opciones que nos brinda esta herramienta:

```
G:\sites\node\utn>express -h

  Usage: express [options] [dir]


  Options:

        --version        output the version number
    -e, --ejs            add ejs engine support
        --pug            add pug engine support
        --hbs            add handlebars engine support
    -H, --hogan          add hogan.js engine support
    -v, --view <engine>  add view <engine> support (dust|ejs|hbs|hjs|jade|pug|twig|vash) (defaults to jade)
        --no-view        use static html instead of view engine
    -c, --css <engine>   add stylesheet <engine> support (less|stylus|compass|sass) (defaults to plain css)
        --git            add .gitignore
    -f, --force          force on non-empty directory
    -h, --help           output usage information
```

Crear una aplicación

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Crear aplicación

Abrimos la consola y nos ubicamos en el directorio en el cual queremos crear nuestra aplicación, luego ejecutamos:

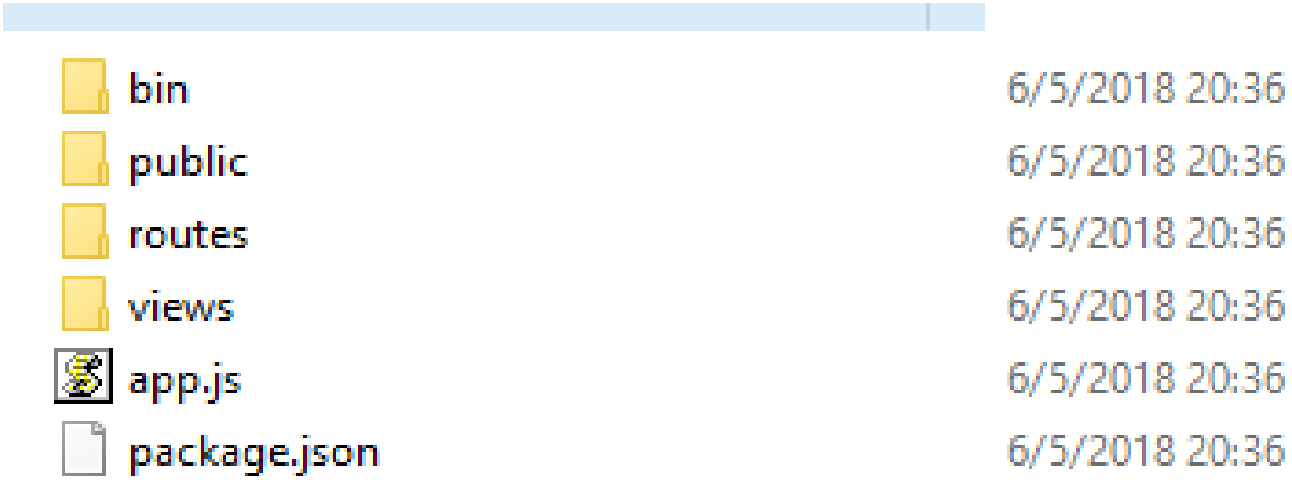
```
G:\sites\node>express --view=ejs myapp_utn

   create : myapp_utn\
   create : myapp_utn\public\
   create : myapp_utn\public\javascripts\
   create : myapp_utn\public\images\
   create : myapp_utn\public\stylesheets\
   create : myapp_utn\public\stylesheets\style.css
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Crear aplicación

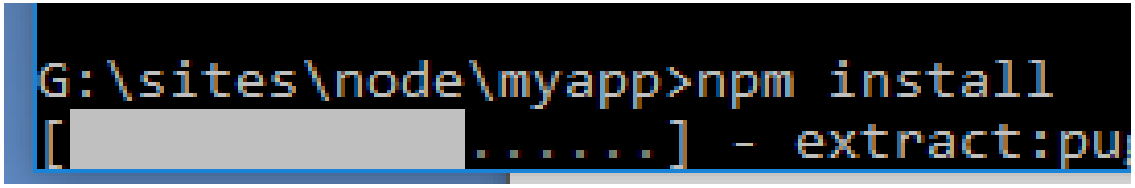
Se creara el directorio de nuestra app utilizando las plantillas de vista **pub**



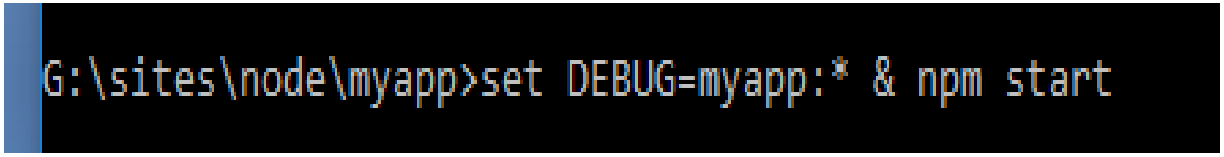
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Crear aplicación

Ingresamos al directorio en el cual se creó nuestra aplicación (desde líneas de comandos) y ejecutamos: **npm install**



Por último ejecutamos:

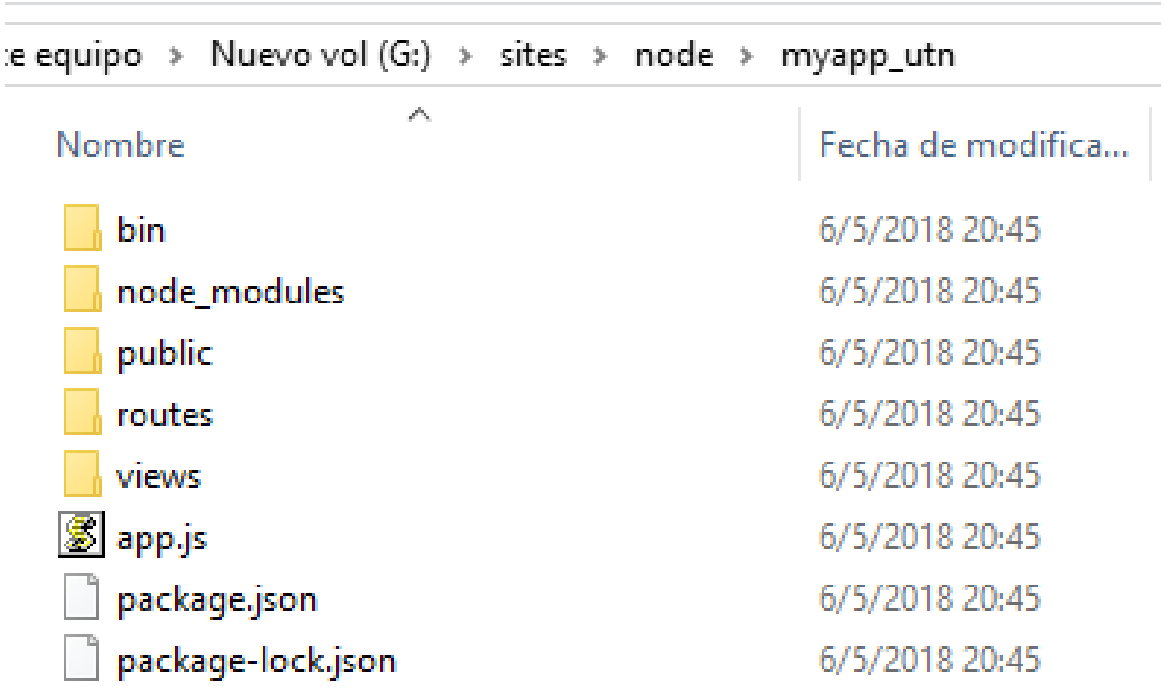


Estructura de directorio

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Estructura de directorio

Al crear nuestra aplicación en node visualizaremos la siguiente estructura de directorio:



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Estructura de directorio

- **bin:** Directorio propio de express, en el cual podremos visualizar la creación de un servidor en node js
- **node_modules:** Carpeta propia de node, allí se alojaran todos los modulos instalados con **npm install**
- **public:** Se aloja el contenido publico
  - **Imágenes**
  - **Javascript**
  - **CSS**
- **routes:** Se alojan los archivos que oficiaran de “ruteadores”, es decir que serán visualizados de acuerdo a la URL que accedamos
- **views**: Se alojan las vistas de nuestra aplicación. En este caso son de tipo **ejs**

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Direccionamiento

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Direccionamiento

El direccionamiento hace referencia a la determinación de cómo responde una aplicación a una solicitud de cliente en un determinado punto final, que es un URI (o una vía de acceso) y un método de solicitud HTTP específico (GET, POST, etc.). Cada ruta puede tener una o varias funciones de manejador, que se excluyen cuando se correlaciona la ruta. La definición de ruta tiene la siguiente estructura:

```
app.METHOD(PATH, HANDLER)
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Direccionamiento

```
app.METHOD(PATH, HANDLER)
```

- app es una instancia de express.
- METHOD es un método de solicitud HTTP (GET, POST, PUT, DELETE).
- PATH es una vía de acceso en el servidor. Ejemplo **“/”, “users”, “users/1”**
- HANDLER es la función que se ejecuta cuando se correlaciona la ruta. Es la función de callback que se ejecutara al ingresar por esa ruta.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Direccionamiento

Por ejemplo en el directorio **route** encontramos el archivo **index.js**, el mismo tendrá la definición de nuestra ruta por default (es decir cuando accedemos a la home de nuestro sitio “/”)

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Direccionamiento

En el siguiente ejemplo vemos un caso similar pero para una petición de tipo POST:

```
// POST method route
app.post('/', function (req, res) {
  res.send('POST request to the homepage');
});
```

Los objetos **req** y **res** contendrán la información del request y reponse. Como vemos en ambos casos el método **send** corresponde al objeto **response**

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Middleware

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Middleware

Las funciones de middleware son funciones que se ejecutaran en cada request y response que reciba nuestra aplicación, por cada una de estas interacciones se ejecutaran las funciones definidas
Para ello utilizamos el método **all**

```
app.all('/secret', function (req, res, next) {
  console.log('Accessing the secret section ...');
  next(); // pass control to the next handler
});
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Middleware

En el siguiente ejemplo, el manejador se ejecutará para las solicitudes a "/secret", tanto si utiliza GET, POST, PUT, DELETE, como cualquier otro método de solicitud HTTP soportado. Con el método **next** se ejecuta el contenido del manejador definido para la petición realizada, es decir primero se ejecutara esta función y luego la definida en el route para dicha petición.

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Rutas con Expresiones regulares

Node JS – Expresiones regulares

Esta vía de acceso de ruta coincidirá con acd y abcd.

```
app.get('/ab?cd', function(req, res) {
  res.send('ab?cd');
});
```

Esta vía de acceso de ruta coincidirá con abcd, abbcd, abbbcd, etc.

```
app.get('/ab+cd', function(req, res) {
  res.send('ab+cd');
});
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Expresiones regulares

Esta vía de acceso de ruta coincidirá con abcd, abxcd, abRABDOMcd, ab123cd, etc.

```
app.get('/ab(cd)?e', function(req, res) {
 res.send('ab(cd)?e');
});
```

Funciones de llamadas

Node JS – Funciones de llamadas

Podremos definir una pila de funciones de llamadas encadenadas, por ejemplo:
Una función de devolución de llamada individual puede manejar una ruta.

```
app.get('/example/a', function (req, res) {
  res.send('Hello from A!');
});
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Funciones de llamadas

Más de una función de devolución de llamada puede manejar una ruta (asegúrese de especificar el objeto next).

```
app.get('/example/b', function (req, res, next) {
  console.log('the response will be sent by the next function ...');
  next();
}, function (req, res) {
  res.send('Hello from B!');
});
```

Se ejecuta primero el contenido de la primer funciona (donde está el console.log()) y luego el de la segunda función (en donde está el send()) **En este caso es importante utilizar el llamado a la función next().**

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Funciones de llamadas

Una matriz de funciones de devolución de llamada puede manejar una ruta.

```
var cb0 = function (req, res, next) {
  console.log('CB0');
  next();
}

var cb1 = function (req, res, next) {
  console.log('CB1');
  next();
}

var cb2 = function (req, res) {
  res.send('Hello from C!');
}

app.get('/example/c', [cb0, cb1, cb2]);
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Métodos de Respuesta

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – 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.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Router

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Router

Utilice la clase express.Router para crear manejadores de rutas montables y modulares. Una instancia Router es un sistema de middleware y direccionamiento completo; por este motivo, a menudo se conoce como una “miniaplicación”.

El siguiente ejemplo crea un direccionador como un módulo, carga una función de middleware en él, define algunas rutas y monta el módulo de direccionador en una vía de acceso en la aplicación principal.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Router

```
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;
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Router

```
var birds = require('./birds');
...
app.use('/birds', birds);
```

Como vemos con el método **use** podemos definir el middleware para esta ruta

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Ejercicio

Desarrollar una página de inicio de un ecommerce. En la misma se deben mostrar 4 productos maquetados con los siguientes datos:

- Nombre
- Descripción
- Imagen
- Precio
- Precio con descuento
- Código

Al hacer clic en el producto la aplicación nos debe redirigir a la página de detalle del mismo en la cual visualizaremos la misma información de la home.