

Pasos

Se trata de un framework de node.js

Instalacion npm install express --save

Importando express const express = require('express');

Invocando express const app = express();

Entry point: Debemos crear un .js donde al correrlo, se creara el servidor. Este archivo .js se llamara

app.js

Configurar package.json para que inicie app.js y no index.js

Levantando el servidor

Metodo listen Pone a correr el servidor.
`app.listen(/port, callback)`

Metodo get Le pasamos al servidor que tiene que correr.
`app.get(/ruta, function ())`

Levantando el servidor (cont)

req, res **req:** Tendra todas las propiedades y metodos de la peticion que se envio.
res: Tendra todas las propiedades y metodos de la peticion que se respondio..

```
app.listen( 3030, ( ) => {console.log ("El servidor esta corriendo perfectamente" )})
```

```
app.get( '/', function (req, res) => { res.send( 'Bienvenido a la pagina web' )})
```

```
app.get( '/contact', function ( req, res) => { res.send( 'Bienvenido a la seccion de contacto' )})
```

Bases

Se trata de un framework de node.js

Instalacion En un framework inicializado de node, escribo;

METHODS

GET

POST

SEND

Enviando archivos

sendFile() Nos permite enviar archivos existentes en nuestro servidor como respuesta a los clientes.

path **sendFile()** precisa, como parametro, una *ruta absoluta*. Para eso hacemos uso del metodo **path**. Se trata de un modulo nativo.

```
const path = require('path');
app.get('/', function (req,res) =>{
res.sendFile(path.join(__dirname, "views/index.html"))}
```

Ejemplo de codigo

publicPath Lo utilizamos para establecer la ruta absoluta de la carpeta donde queremos estar parados.

```
const express = require('express');
const path = require('path');
const PORT = 3030;
const app = express();
```

```
const publicPath = path.resolve(__dirname, './public');
```

```
app.use(express.static(publicPath));
```

```
app.get('/', (req, res) => {res.sendFile(path.join(__dirname, './views/home.html'))})
```

```
app.listen(PORT, ( ) => {console.log ('Server listen in ${PORT}')}})
```

