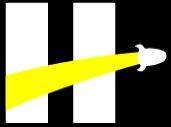
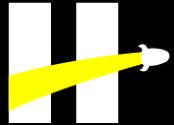


HENRY



Express

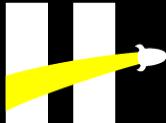


Express

Es un framework web rápido, minimalista y
flexible para [Node.js](#)



```
1 npm install express --save
```



Express

```
var express = require('express');
var app = express();
app.get('/', function(req, res, next) {
  next();
})
app.listen(3000);
```

HTTP method for which the middleware function applies.

Path (route) for which the middleware function applies.

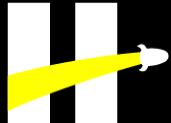
The middleware function.

Callback argument to the middleware function, called "next" by convention.

HTTP response argument to the middleware function, called "res" by convention.

HTTP request argument to the middleware function, called "req" by convention.

```
graph LR; A[HTTP method for which the middleware function applies.] --> B[app.get()]; C[Path (route) for which the middleware function applies.] --> D[/]; E[The middleware function.] --> F[function(req, res, next){};]; G[Callback argument to the middleware function, called "next" by convention.] --> H[next()]; I[HTTP response argument to the middleware function, called "res" by convention.] --> J[res]; K[HTTP request argument to the middleware function, called "req" by convention.] --> L[req];
```



Express - Routing



```
1 // Esta ruta matcheará acd y abcd
2 app.get('/ab?cd', function(req, res) {
3   res.send('ab?cd');
4 });
5
6 // Esta ruta matcheará abcd, abbcd, abbbcd, y así sucesivamente
7 app.get('/ab*cd', function(req, res) {
8   res.send('ab*cd');
9 });
10
11 // pasando parámetros
12 app.get('/api/:id', function(req, res) {
13   res.json( { parametro: req.params.id } );
14 });
```

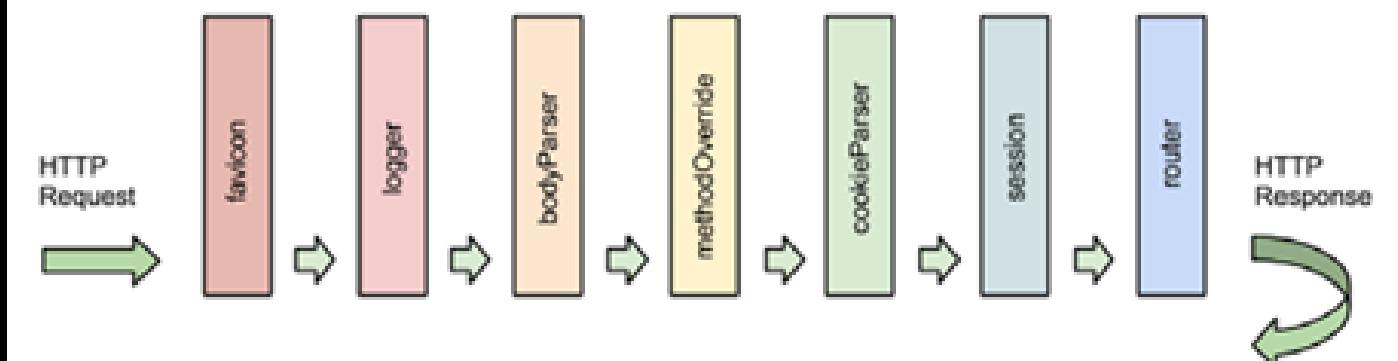
Express nos ofrece muchísimas soluciones para varios problemas, es por eso que vamos a tener que aprender a leer y buscar en la documentación (que por cierto es muy buena) de *express*.

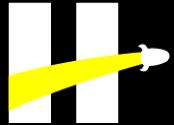


Express -MiddleWares

The Express Middleware Stack

An HTTP request may pass through a series of Express middlewares before one of the middlewares or route handlers returns the HTTP response.





Express -MiddleWares



```
1 app.use('/', function(req,res, next ){
2     console.log('Hicieron un Request a '+ req.url);
3     next();
4});
```



Express - Enviando Datos al server

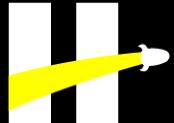
Query String

Una forma de enviar datos es hacerlo en la URL a la que apuntamos el request. Para ello nos valemos de una serie de parámetros o datos que se incluyen en la URL. Normalmente distinguimos en la URL por un nombre y un valor separados por el signo igual, y se separan del endpoint por el carácter ?, y entre cada variable por el signo &. Por ejemplo:

```
GET /?id=4&page=3 HTTP/1.1
Host: www.learnwebdev.net
Cookie: username=abc;name=Tony
```



```
1 app.get('/datos/', function(req, res) {
2   res.json( req.query );
3 });
```



Express - Enviando Datos al server

Otras formas de enviar datos al servidor es a través de Formularios. Ahora usaremos otro verbo HTTP, el **POST**.

POST / HTTP/1.1

Host: www.learnwebdev.net

Content-Type: application/x-www-form-urlencoded

Cookie: num=4;page=2

username=Tony&password=pwd



```
1 var bodyParser = require('body-parser');
2
3 var urlencodedParser = bodyParser.urlencoded({ extended: false })
4 app.post('/form', urlencodedParser, function (req, res) {
5   res.json( req.body )
6 });
```



Express - Enviando Datos al server

También podemos enviar datos al servidor en formato JSON, donde también usamos POST, pero el content-type es ahora application/json. Por ejemplo en un request generado por AJAX. En este caso los datos también estarán en el body del request.

POST / HTTP/1.1
Host: www.learnwebdev.net
Content-Type: application/json
Cookie: num=4;page=2

```
{  
  "username": "Tony",  
  "password": "pwd"  
}
```



```
1 var jsonParser = bodyParser.json()  
2 app.post('/formjson', jsonParser, function (req, res) {  
3   res.json( req.body )  
4 });
```