

## Install

NPM install console

```
$ npm install express --save
```

## Express Methods

Require and create an Express application code

```
const express = require( 'express' );
const app = express( ) ;
```

**express.static()** - serves static files code

```
app.use(express.static(__dirname + '/public'));
```

**app.listen(port, [callback])** - Binds port, listens for connections

```
app.listen(8080, () => { console.log(/*...*/); });
```

## Middleware

**Middleware** - fn(err, req, res, next){}

Define middleware, call `next()` to proceed code

```
const myLogger = function(req, res, next ) {
  next(); ...
```

**Loading middleware with `app.use` or `router.use`** code

```
<app|router>.use(logger);
```

**Loading middleware inline on an endpoint** code

```
<app|router>.get('/cars' , myLogger, (req, res) => {
  res.send('Hello World');
```

Error-handling middleware - fn(err, req, res, next){} code

```
<app|router>.use(function (err, req, res, next) {
  next(); ...
```

## Routing

**Routing** - how the app responds to a request to a particular **endpoint**.

**Endpoint** - a HTTP request method (GET, POST etc) plus a Path

Each HTTP method corresponds to an express route methods

HTTP	GET	POST	PUT	DELETE
Express	.get()	.post()	.put()	.delete()

**<app|router>.METHOD(path, [callback, ...] callback)**

Sample .post() endpoint code

```
<app|router>.post('/cars', (req, res) => { /*...*/ })
```

## Request Object

**req.params** - properties mapped to **named route parameters**

```
$ curl 'localhost:8080/cars/toyota'
```

```
<app|router>.get('/cars/:make', (req, res) => {
  req.params.make; // toyota
```

**req.query** - properties mapped to **query string** key/val pairs

```
$ curl 'localhost:8080/cars/?sort=year&order=desc'
```

```
<app|router>.get('/cars/', (req, res) => {
  const {sort, order} = req.query; // year desc
```

**req.body** - object mapped to key/val from message body

Requires body-parser middleware package

```
$ npm install body-parser --save
```

Require and use middleware

```
const bodyParser = require('body-parser');
<app|router>.use( bodyParser.json() );
```

Object mapped to key/val from message body

```
$ curl -X POST -H "Content-type:application/json"
-d '{"make":"kia","model":"soul"}' localhost:8080
```

```
<app|router>.post('/cars/', (req, res) => {
  const {make, model} = req.body; // kia soul
```

**req.get** - get specified HTTP request header

```
$ curl -H "Accept:text/plain" localhost:8080
```

```
app.get('/cars/', (req, res) => {
  req.get('Accept'); // text/plain ...
```

**req.cookies** - object that contains cookies sent by the request

Requires cookie-parser middleware

```
$ npm install cookie-parser --save
```

Require and use middleware

```
const cookieParser = require( 'cookie-parser' );
<app|router>.use( cookieParser() );
```

**req.cookies** - object that contains cookies sent by the request

```
$ curl --cookie "token=SeAx1aVyKdw" localhost:8080
```

```
<app|router>.get('/cars/', (req, res) => {
  const {token} = req.cookies; // SeAx1aVyKdw
```

## Response Object

**res.send( [body] )** - Sends the HTTP response code

```
res.send('<p>some html</p>');
```

**res.json( [body] )** - Sends a JSON response code

```
res.json( {foo: 'bar'} );
```

**res.sendStatus(code)** - Sets and sends the HTTP response status

```
res.sendStatus(200); // res.status(200).send('OK')
```

**res.status(code)** - Sets the HTTP status - does **NOT** send code

```
res.status(400).send( 'Bad Request' );
res.status(500).json( {error: 'it blew up'} );
```

**res.location(path)** - Sets Location header - does **NOT** send code

```
res.location('http://example.com/item/35');
```

**res.cookie(name, value [, options])** - Sets cookie name, value

```
res.cookie('token', '123456', {
  path: '/auth, secure: true, httpOnly: true ...
```

**res.set(field [,value])** - Sets a header - does **NOT** send code

```
res.set('Content-Type', 'text/plain');
```

**res.end([data] [, encoding])** - Ends response and sends code

```
res.status(404).end();
```

**res.sendFile(path [, options] [, fn])** - Sends file code

```
res.sendFile('/public/index.html');
```

**res.redirect([status,] path)** - Redirects to the URL code

```
res.redirect('/admin');
```

## Router (guide)

**express.router()** - isolated instance of middleware and routes code

```
const express = require('express')
const router = express.Router()
```

```
router.use(function(req, res, next) {
  next();
});
```

```
router.get('/', (req, res) => { /*...*/ });
```

```
module.exports = router;
```