



## INSTALLATION

Install the latest Express.js locally  
`$ sudo npm install express`

Install Express.js v4.2.0 locally and save to *package.json*  
`$ sudo npm install express@4.2.0 --save`

Install Express.js command-line generator v4.0.0  
`$ sudo npm install -g express-generator@4.0.0`

## GENERATOR

Usage  
`$ express [options] [dir]`

Print the usage information  
`-h`

Print the express-generator version number  
`-V`

Add ejs engine support, defaults to jade if omitted  
`-e`

Add hogan.js engine support  
`-H`

Add CSS support for (less|stylus|compass), defaults to plain CSS if omitted  
`-c <library>`

Generate into a non-empty directory  
`-f`

## BASICS

Include the module  
`var express = require('express')`

Create an instance  
`var app = express()`

Start the Express.js server  
`app.listen(portNumber, callback)`

Start the Express.js server  
`http.createServer(app).listen(portNumber, callback)`

Set a property value by the key  
`app.set(key, value)`

Get a property value by the key  
`app.get(key)`

## ROUTES AND MIDDLEWARE

`app.get(url, requestHandler, [requestHandler2, ...])`

`app.post(url, requestHandler, [requestHandler2, ...])`

`app.put(url, requestHandler, [requestHandler2, ...])`

`app.del(url, requestHandler, [requestHandler2, ...])`

`app.all([url,] requestHandler, [requestHandler2, ...])`

`app.param([name,] callback)`

`app.use([url,] requestHandler, [requestHandler2, ...])`

## REQUEST

Parameters middleware  
`request.params`

Extract one parameter  
`request.param`

Extract query string parameter  
`request.query`

Return route string  
`request.route`

Cookies, requires *cookieParser*  
`request.cookies`

Signed cookies, requires *cookie-parser* `-c <library>`  
`request.signedCookies`

Payload, requires *body-parser*  
`request.body`

## RESPONSE

Redirect request  
`response.redirect(status, url)`

Send response  
`response.send(status, data)`

Send JSON and force proper headers  
`response.json(status, data)`

Send a file  
`response.sendFile(path, options, callback)`

Render a template  
`response.render(templateName, locals, callback)`

Pass data to template  
`response.locals`

## REQUEST HEADER SHORTCUTS

Value for the header key  
`request.get(headerKey)`

Checks if the type is accepted  
`request.accepts(type)`

Checks language  
`request.acceptsLanguage(language)`

Checks charset  
`request.acceptsCharset(charset)`

Checks the type  
`request.is(type)`

IP address  
`request.ip`

IP addresses (with trust-proxy on)  
`request.ips`

URL path  
`request.path`

Host without port number  
`request.host`

Checks freshness  
`request.fresh`

Checks staleness  
`request.stale`

True for AJAX-y requests  
`request.xhr`

Returns HTTP protocol  
`request.protocol`

Checks if protocol is *https*  
`request.secure`

Array of subdomains  
`request.subdomains`

Original URL  
`request.originalUrl`

## HANDLERS SIGNATURES

Request handler signature  
`function(request, response, next) {}`

Error handler signature  
`function(error, request, response, next) {}`



## STYLUS AND JADE

```
app.set('views', path.join(__dirname, 'views'))
app.set('view engine', 'jade')
```

```
app.use(require('stylus').middleware(path.join(__dirname, 'public')))
```

## BODY

```
var bodyParser = require('body-parser')
app.use(bodyParser.json())
app.use(bodyParser.urlencoded())
```

## STATIC

```
app.use(express.static(path.join(__dirname, 'public')))
```

## OTHER POPULAR MIDDLEWARE

Analogous to *cookieParser*  
**cookies and keygrip**

Raw body  
**raw-body**

Connect middleware for multipart, connect middleware for busboy  
**connect-multipart, connect-busboy**

Analogous to *query*  
**qs**

Analogous to *staticCache*  
**st, connect-static**

Validation  
**express-validator**

LESS CSS  
**less**

Authentication library  
**passport**

Security headers  
**helmet**

CORS  
**connect-cors**

Connect Redis  
**connect-redis**

## CONNECT MIDDLEWARE

```
$ sudo npm install <package_name> --save
```

Request payload  
**body-parser**

Gzip  
**compression**

Times out the request in *ms*, defaulting to *5000*.  
**connect-timeout**

Cookies  
**cookie-parser**

Session via Cookies store  
**cookie-session**

CSRF  
**csurf**

Error handler  
**errorhandler**

Session via in-memory or other store  
**express-session**

HTTP method override  
**method-override**

Server logs  
**morgan**

Response time middleware  
**response-time**

Favicon  
**serve-favicon**

Serves pages that contain directory listings  
**serve-index**

Static content  
**serve-static**

Virtual Host  
**vhost**