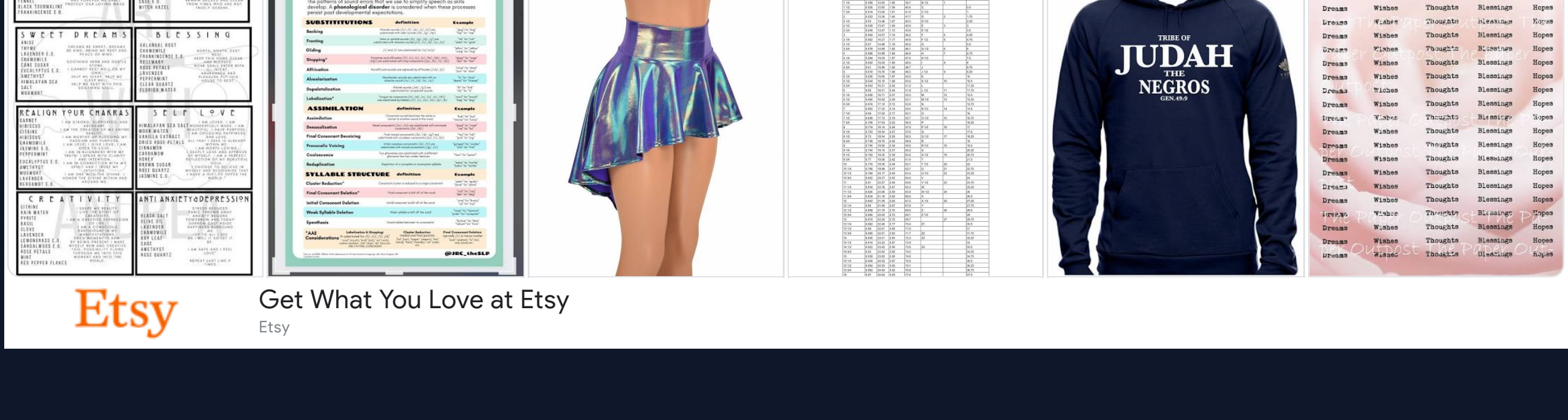


# Express cheatsheet

A quick reference cheatsheet for Express, a flexible and streamlined web framework for Node.js



## # Getting Started

</

## # Example

<div><div>Router</div><div><p>Called for any request passed to this router</p><pre>router.use(function (req, res, next) {   //.. some logic here .. like any other middleware   next() })</pre><p>will handle any request ending in /events</p><pre>//depends on where the router "use()" router.get('/events', (req, res, next) =&gt; {   //.. })</pre></div></div>	<div><div>Response</div><div><p>The <b>res</b> object represents the HTTP response sent by the Express application when it receives an HTTP request</p><pre>app.get('/user/:id', (req, res) =&gt; {   res.send('user' + req.params.id) })</pre></div></div>	<div><div>Request</div><div><p>A <b>req</b> object represents an HTTP request and has properties for the request query string, parameters, body, HTTP headers, etc.</p><pre>app.get('/user/:id', (req, res) =&gt; {   res.send('user' + req.params.id) })</pre></div></div>
<div><div>res.end()</div><div><pre>res.end() res.status(404).end()</pre><p>End the response process. This method actually comes from the Node core, specifically the <code>response.end()</code> method of <code>http.ServerResponse</code></p></div></div>	<div><div>res.json([body])</div><div><pre>res.json(null) res.json({ user: 'tobi' }) res.status(500).json({ error: 'message' })</pre></div></div>	<div><div>app.all</div><div><pre>app.all('/secret', function (req, res, next) {   console.log('access secret section...')   next() // Pass control to the next handler })</pre></div></div>
<div><div>app.delete</div><div><pre>app.delete('/', function (req, res) {   res.send('DELETE request to homepage') })</pre></div></div>	<div><div>app.disable(name)</div><div><pre>app.disable('trust proxy') app.get('trust proxy') // =&gt; false</pre></div></div>	<div><div>app.disabled(name)</div><div><pre>app.disabled('trust proxy') // =&gt; true  app.enable('trust proxy') app.disabled('trust proxy') // =&gt; false</pre></div></div>
<div><div>app.engine(ext, callback)</div><div><pre>var engines = require('consolidate')  app.engine('haml', engines.haml) app.engine('html', engines.hogan)</pre></div></div>	<div><div>app.listen([port[, host[, backlog]]], [callback])</div><div><pre>var express = require('express')  var app = express() app.listen(3000)</pre></div></div>	<div><div>Routing</div><div><pre>const express = require('express') const app = express()  //Respond to "hello world" when making a GET request app.get('/', (req, res) =&gt; {   res.send('hello world') })  // GET method routing app.get('/', (req, res) =&gt; {   res.send('GET request to the homepage') })  // POST method routing app.post('/', (req, res) =&gt; {   res.send('POST request to the homepage') })</pre></div></div>
<div><div>Middleware</div><div><pre>function logOriginalUrl (req, res, next) {   console.log('ReqURL:', req.originalUrl)   next() }  function logMethod (req, res, next) {   console.log('Request Type:', req.method)   next() }  const log = [logOriginalUrl, logMethod]  app.get('/user/:id', log, (req, res, next)=&gt;{   res.send('User info') })</pre></div></div>	<div><div>Using templates</div><div><pre>app.set('view engine', 'pug')</pre><p>Create a Pug template file named <code>index.pug</code> in the views directory with the following content</p><pre>html   the head     title= title   the body     h1=message</pre><p>Create a route to render the <code>index.pug</code> file. If the <code>view</code> engine property is not set, the extension of the view file must be specified</p><pre>app.get('/', (req, res) =&gt; {   res.render('index', {     title: 'Hey', message: 'Hello there!'   }) })</pre></div></div>	



### Related Cheatsheet

ES6 Cheatsheet Quick Reference	JSON Cheatsheet Quick Reference	Homebrew Cheatsheet Quick Reference	PyTorch Cheatsheet Quick Reference
Kubernetes Cheatsheet Quick Reference	TOML Cheatsheet Quick Reference	Taskset Cheatsheet Quick Reference	EJS Cheatsheet Quick Reference

### Recent Cheatsheet