[Express](http://expressjs.com/)

Fast, unopinionated, minimalist web framework for [Node.js](http://nodejs.org/)

# Definition:

Express is a minimal and flexible Node.js web application frame work that provides robust set of features for web and mobile applications.

Express is a framework that takes Node from a barebones application and turns it into something that behaves more like the web servers we're all used to working with.

# Installing:

First create a directory to hold your application. If you haven’t already done so, and make that your working directory.

$ mkdir myapp

$ cd myapp

Create a package.json file in the directory of interest, if it does not exist already, with the npm init command.

$ npm init

Install Express in the app directory and save it in the dependencies list:

$ npm install express --save

To install Express temporarily, and not add it to the dependencies list, omit the --save option:

$ npm install express

# Express application generator:

Use the application generator tool, express, to quickly create a application skeleton.

Install it with the following command.

$ npm install express-generator -g

For example, the following creates an Express app named **myapp** in the current working directory.

$ express myapp

create : myapp

create : myapp/package.json

create : myapp/app.js

create : myapp/public

create : myapp/public/javascripts

create : myapp/public/images

create : myapp/routes

create : myapp/routes/index.js

create : myapp/routes/users.js

create : myapp/public/stylesheets

create : myapp/public/stylesheets/style.css

create : myapp/views

create : myapp/views/index.jade

create : myapp/views/layout.jade

create : myapp/views/error.jade

create : myapp/bin

create : myapp/bin/www

install dependencies:

$ cd myapp && npm install

run the app:

$ DEBUG=myapp ./bin/www

On Windows, use the following command to run the app.

> set DEBUG=myapp & node .\bin\www

The generated app directory structure looks like the following.

.

├── app.js

├── bin

│ └── www

├── package.json

├── public

│ ├── images

│ ├── javascripts

│ └── stylesheets

│ └── style.css

├── routes

│ ├── index.js

│ └── users.js

└── views

├── error.jade

├── index.jade

└── layout.jade

# Hello world example:

Here is an example of a very basic Express app.

var express = require('express')

var app = express()

app.get('/', function (req, res) {

res.send('Hello World!')

})

var server = app.listen(3000, function () {

var host = server.address().address

var port = server.address().port

console.log('Example app listening at http://%s:%s', host, port)

})

The req (request) and res (response) are the exact same objects that Node provides, so you can invoke req.pipe(), req.on('data', callback) and anything else you would do without Express involved.

The app starts a server and listens on port 3000 for connection. It will respond with "Hello World!" for requests to the homepage. For every other path, it will respond with a **404 Not Found**.

Save the code in a file named app.js and run it with the following command.

$ node app.js

Then, load <http://localhost:3000/> in a browser to see the output.

**Basic Node.js Application using express js framework:**

[Express](http://expressjs.com/) is a minimal [Node.js](http://nodejs.org/) web application framework, inspired by [Sinatra](http://www.sinatrarb.com/).  
Out of the box, Express provides features such as routing, templating/view rendering (via [Consolidate.js](https://github.com/visionmedia/consolidate.js)), dynamic CSS support (via [less](http://lesscss.org/) | [stylus](http://learnboost.github.com/stylus/)), middleware (via [Connect](http://github.com/senchalabs/connect)), session management, logging, application settings, and a command line interface (CLI).

If you already installed express. Don’t need it to install again.

npm install -g express // global install

### Create an app

Express comes with a command line interface (CLI), aptly named express.

express -h // show usage

Let's use it to create an app. To keep things simple, we'll give our app a name and otherwise just use the default options.

express express-skeleton

|  |  |
| --- | --- |
|  | $ express express-skeleton    create : express-skeleton  create : express-skeleton/package.json  create : express-skeleton/app.js  create : express-skeleton/public  create : express-skeleton/public/javascripts  create : express-skeleton/public/images  create : express-skeleton/public/stylesheets  create : express-skeleton/public/stylesheets/style.css  create : express-skeleton/routes  create : express-skeleton/routes/index.js  create : express-skeleton/views  create : express-skeleton/views/layout.jade  create : express-skeleton/views/index.jade    install dependencies:  $ cd express-skeleton && npm install    run the app:  $ node app |

As you can see from the previous console output, the next steps are to install dependencies and then run the app. Let's do just that.

cd express-skeleton && npm install

|  |  |
| --- | --- |
|  | $ cd express-skeleton && npm install  npm http GET https://registry.npmjs.org/express/3.0.0rc2  npm http GET https://registry.npmjs.org/jade  ...    jade@0.27.2 ./node\_modules/jade  ├── commander@0.6.1  └── mkdirp@0.3.0    express@3.0.0rc2 ./node\_modules/express  ├── methods@0.0.1  ├── fresh@0.1.0  ├── range-parser@0.0.4  ├── cookie@0.0.4  ├── crc@0.2.0  ├── commander@0.6.1  ├── debug@0.7.0  ├── mkdirp@0.3.3  ├── send@0.0.3 (mime@1.2.6)  └── connect@2.4.2 (bytes@0.1.0, pause@0.0.1, qs@0.4.2, formidable@1.0.11) |

That last command downloaded and installed all of our application's dependencies into a new. /node\_modules folder. Now let's start the app with the following command.

node app

|  |  |
| --- | --- |
|  | $ node app  Express server listening on port 3000 |

### Try it out

View the app via curl or in a web browser, then Ctrl-C to stop the server. curl http://localhost:3000 oropen http://localhost:3000

where you will see a welcome to Express page. You are now running your own Node JS webserver, with the Express engine and Jade HTML preprocessor installed

