

Express

Express application generator

Use the application generator tool, `express-generator`, to quickly create an application skeleton.

You can run the application generator with the `npx` command (available in Node.js 8.2.0).

```
$ npx express-generator
```

For earlier Node versions, install the application generator as a global npm package and then launch it.

```
$ npm install -g express-generator  
$ express
```

Display the command options with the `-h` option:

```
$ express -h  
  
Usage: express [options] [dir]  
  
Options:  
  
  -h, --help            output usage information  
      --version          output the version number  
  -e, --ejs             add ejs engine support  
      --hbs             add handlebars engine support  
      --pug             add pug engine support  
  -H, --hogan           add hogan.js engine support  
      --no-view          generate without view engine  
  -v, --view <engine>  add view <engine> support (ejs|hbs|hjs|jade|pug|twig|vash)  
                        (defaults to jade)  
  -c, --css <engine>   add stylesheet <engine> support (less|stylus|compass|sass)  
                        (defaults to plain css)  
      --git             add .gitignore  
  -f, --force           force on non-empty directory
```

For example, the following creates an Express app named ***myapp***. The app will be created in a folder named ***myapp*** in the current working directory and the view engine will be set to [Pug](#):

```
$ express --view=pug 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.pug
create : myapp/views/layout.pug
create : myapp/views/error.pug
create : myapp/bin
create : myapp/bin/www
```

Then install dependencies:

```
$ cd myapp
$ npm install
```

On MacOS or Linux, run the app with this command:

```
$ DEBUG=myapp:* npm start
```

On Windows Command Prompt, use this command:

```
> set DEBUG=myapp:* & npm start
```

On Windows PowerShell, use this command:

```
PS> $env:DEBUG='myapp:*'; npm start
```

Then load `http://localhost:3000/` in your browser to access the app.

The generated app has the following directory structure:

```
.
├─ app.js
├─ bin
│   └─ www
└─ package.json
```

```
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.pug
    ├── index.pug
    └── layout.pug
```

7 directories, 9 files

The app structure created by the generator is just one of many ways to structure Express apps. Feel free to use this structure or modify it to best suit your needs.

[Previous: Hello World](#) [Next: Basic routing](#)

Documentation translations provided by [StrongLoop/IBM](#): [French](#), [German](#), [Spanish](#), [Italian](#), [Japanese](#), [Russian](#), [Chinese](#), [Traditional Chinese](#), [Korean](#), [Portuguese](#).

Community translation available for: [Slovak](#), [Ukrainian](#), [Uzbek](#), [Turkish](#) and [Thai](#).



Star

[Express](#) is a project of the [OpenJS Foundation](#).

[Edit this page on GitHub](#).

Copyright © 2017 StrongLoop, IBM, and other [expressjs.com](#) contributors.



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 United States License](#).