

# **EXPRESS INTRODUCTION**

BY 程序猿小卡



- What is Express?
- Why we need Express?
- Rules of routers.
- Everything is middleware.
- Combo Application



# WHAT IS EXPRESS

Express

Fast, unopinionated, minimalist  
web framework for Node.js

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



# WHY WE NEED IT

app.js



```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  res.end('Hello World\n');  
}).listen(1337, '127.0.0.1');  
  
console.log('Server running at http://127.0.0.1:1337/');
```



127.0.0.1:1337



应用



项目



工作常用到



技术



新奇项

Hello World



# WHAT IS EXPRESS

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



The image shows a code editor with two tabs: 'app.js — server-with-node' and 'app.js — server-with-express'. The 'server-with-express' tab is active, displaying the following JavaScript code:

```
1 var express = require('express'),
2   app = express();
3
4 app.use(function(req, res, next) {
5   res.send('Hello World\n');
6 });
7 app.listen(1337);
8
9 console.log('Server running & listening on port 1337');
```

Below the code editor, a web browser window is visible. The address bar shows '127.0.0.1:1337'. The browser's tab bar includes icons for '应用' (Applications), '项目' (Projects), '工作常用到' (Work常用到), and '技术' (Technology). The main content area of the browser displays 'Hello World'.

← → ↻ 🏠 127.0.0.1:1337

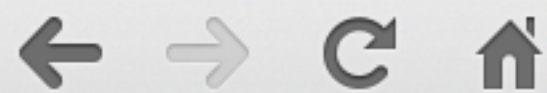
应用 项目 工作常用到 技术

Hello World



server-with-static-resources/

└─ app.js  
└─ public  
 └─ a.js  
 └─ b.js



127.0.0.1:1337/a.js



应用



项目



工作常用到



技术



新奇项目

```
module.exports = function() {  
    console.log('I am A !');  
};
```

```
15  var exp  
16  app };  
17  
18  app.use  
19  app.use(function(req, res, next) {  
20      res.send('Hello World\n');  
21  });  
22  app.listen(1337);  
23  
24  console.log('Server running at http://127.0.0.1:1337/');
```



# WHY WE NEED EXPRESS

```
npm install express
```

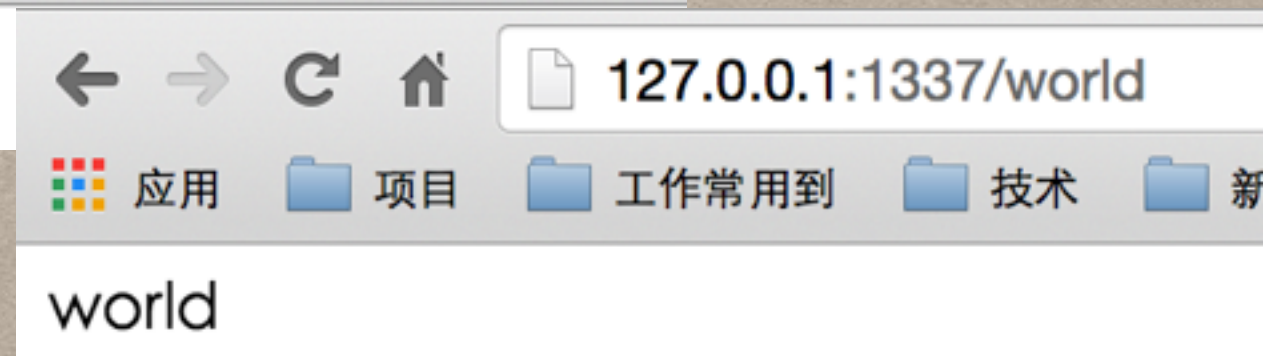
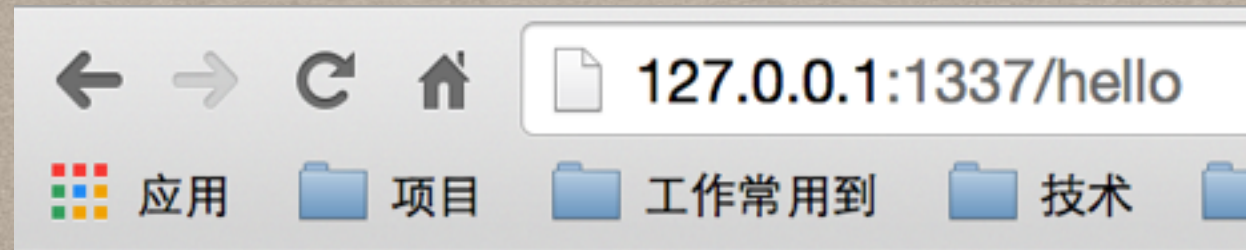
The image shows a code editor on the left and a web browser's developer console on the right. The code editor contains the following JavaScript code:

```
1 var express = require('express');
2 app = express();
3 // ...
4
5 app.use(bodyParser.json());
6 app.use(bodyParser.urlencoded({ extended: true }));
7 app.use(express.static(path.resolve(__dirname, 'public')));
8
9 app.listen(1337, function() {
10   console.log('Server running at http://127.0.0.1:1337/');
11 });
```

The browser's developer console is open to the Network tab, showing a list of resources. The resource 'a.js' is selected. The 'Headers' sub-tab is active, displaying the 'Response Headers' section. The 'Content-Encoding: gzip' header is highlighted with a red box. Other visible headers include 'Accept-Ranges: bytes', 'Cache-Control: public, max-age=0', 'Connection: keep-alive', 'Content-Type: application/javascript', 'Date: Thu, 23 Apr 2015 14:41:23 GMT', 'ETag: W/"39-2067554222"', 'Last-Modified: Thu, 23 Apr 2015 14:39:08 GMT', 'Transfer-Encoding: chunked', 'Vary: Accept-Encoding', and 'X-Powered-By: Express'.



# RULES OF ROUTER



```
$ curl -X POST http://127.0.0.1:1337/login
```

```
login API requested!
```



```
1 var express = require('express'),
2   app = express(),
3   compression = require('compression');
4
5 // routers based on path
6 app.get('/hello', function(req, res, next) {
7   res.send('hello');
8 });
9
10 app.get('/world', function(req, res, next) {
11   res.send('world');
12 });
13
14 // router based on method
15 app.get('/login', function(req, res, next){
16   res.send('login page requested!');
17 });
18
19 app.post('/login', function(req, res, next){
20   res.send('login API requested! \n\n');
21 });
22
23 // router with regular expression
24 app.get(/^\/cgi-bin\/(\w+)$/, function(req, res, next){
25   res.send('CGI ' + req.params[0] + ' is called !');
26 });
27
28 app.get('/cgi-bin/agency/:cgi_name', function(req, res, next){
29   res.send('Agency CGI ' + req.params.cgi_name + ' is called !');
30 });
31
32 app.listen(1337);
33
34 console.log('Server running at http://127.0.0.1:1337/');
```

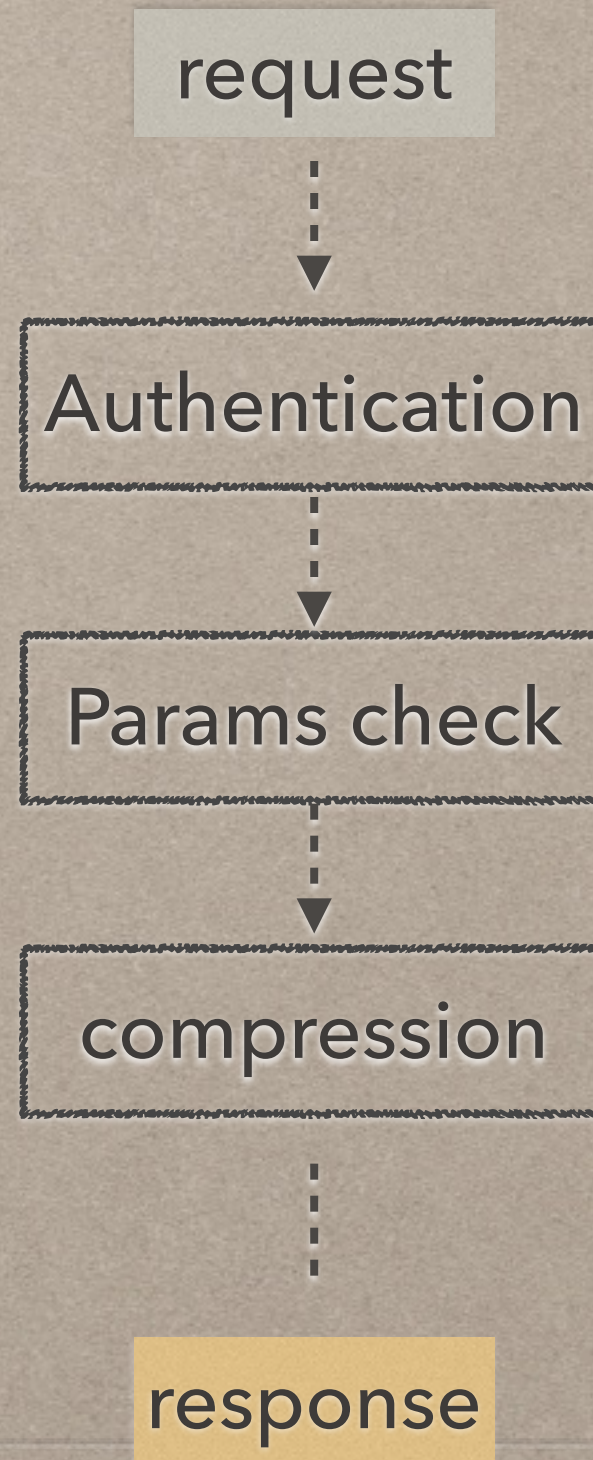
根据路径

根据请求方法

复杂路径



# EVERYTHING IS MIDDLEWARE





```
22
23 exports = module.exports = function cookieParser(secret, options){
24   return function cookieParser(req, res, next) {
25     if (req.cookies) return next();
26     var cookies = req.headers.cookie;
27
28     req.secret = secret;
29     req.cookies = Object.create(null);
30     req.signedCookies = Object.create(null);
31
32     // no cookies
33     if (!cookies) {
34       return next();
35     }
36
37     req.cookies = cookie.parse(cookies, options);
38
39     // parse signed cookies
40     if (secret) {
41       req.signedCookies = parse.signedCookies(req.cookies, secret);
42       req.signedCookies = parse.JSONCookies(req.signedCookies);
43     }
44
45     // parse JSON cookies
46     req.cookies = parse.JSONCookies(req.cookies);
47
48     next();
49   };
50 };
```



# SIMPLE MIDDLEWARE


```
// routers based on path
app.get('/hello', function(req, res, next) {
  res.send('hello');
});
```





# COMBO APPLICATION

*Get the code*

 **chyingp / e-combo**

Unwatch ▾ 1



一个基于node的combo服务器 — Edit

9 commits



1 branch







0 releases

1 contributor

 branch: master ▾ **e-combo** / + 

Merge branch 'master' of https://github.com/chyingp/e-combo ...

 **chyingp** authored 23 hours ago latest commit de9c099b77 

 lib	主要文件	a day ago
 public	主要文件	a day ago
 routers	bug fix: 非合并的资源请求出错	a day ago
 README.md	Update README.md	a day ago
 index.js	添加gzip压缩	23 hours ago
 package.json	添加gzip压缩	23 hours ago



# RESOURCES

- Express demo:<https://github.com/chyingp/blog/tree/master/demo/2015.04.21-express>
- Combo App:<https://github.com/chyingp/e-combo>
- About me:<https://github.com/chyingp>