**Express开发**

# 相关

## 资源网站

<https://www.npmjs.com/package/express-generator>

npm i -g express-generator

<https://github.com/expressjs/generator>

中文站点： <http://www.expressjs.com.cn/> **<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**

## 使用express搭建第一个Web应用【Node.js初学】

<http://jingyan.baidu.com/article/bad08e1ee501e009c8512106.html>

## 通过NodeJS,Express搭建本地HTTP访问服务

<http://jingyan.baidu.com/article/2fb0ba4069e57400f3ec5f7a.html>

express 安装nodejs项目hello。

express hello

cd hello && npm install

node app.js

http://localhost:3000/ 进入访问

## nodejs中express 入门小实例

<http://blog.csdn.net/qq_28666081/article/details/52063197>

**源码下载：https://yunpan.cn/c6HIDhDhNdvy7  访问密码 286d**

## node.js基于express框架搭建一个简单的注册登录Web功能

<http://www.cnblogs.com/Leo_wl/p/4361289.html>

**需要代码的可移步至Github：**  <https://github.com/imwtr/nodejs_express_login_register>

## 插件

**http://www.cnblogs.com/Darren\_code/p/node\_express.html**

**Node的小基友supervisor 和 forever 不要忘记了，相信你会喜欢他们的:)**

|  |
| --- |
| 1、安装express4 npm install -g express@4 npm install -g express-generator@4 express -V 查看版本 2、建立项目 比方说在e:\project中建立 firstApp那就  cd e:\project express -e firstApp    (-e代表  ejs 模板  -J 代表 jade 模板) 然后根据提示   cd firstApp && npm install  (安装各依赖插件) 再输入  set DEBUG=firstApp:\* & npm start (在express4中 npm start  代替了原先的方法  node app.js) 这时会有提示  firstApp：server Listening on port 3000 +0ms 可以访问 localhost:3000 插件 1、socket.io    已集成在express中 2、pm2    [pm2 是一个带有负载均衡功能的Node应用的进程管理器. 当你要把你的独立代码利用全部的服务器上的所有CPU，并保证进程永远都活着，0秒的重载， PM2是完美的](http://blog.sina.com.cn/s/blog_537517170102w1w4.html) 3、generic-pool  (github 项目为 node-pool) [一个数据库连接池，可连接mysql/redis等多种常见数据库](https://github.com/coopernurse/node-pool) 4、promise [promise 解决nodejs 无穷无尽回调的一种标准](http://blog.sina.com.cn/s/blog_537517170102w1w7.html) 目前主流的方案有 1>、Q 使用人最多，文档最多的，项目地址 https://github.com/kriskowal/q 2>、co 需要nodejs0.11 以上版本，新东西自有他的特色 3>、bluebird 很多人推荐，不明白为什么用日本车牌子  项目地址 https://www.npmjs.com/package/bluebird 4>、then 非常轻的，个人感觉不错  项目地址  https://github.com/zensh/then.js 5>、async 最简单的，但其实是最接近nodejs原生callback的，项目地址 [https://github.com/caolan/async](https://github.com/caolan/async/) 不错的教程   http://blog.fens.me/nodejs-async/ 6>、when 比较轻小的，项目地址 https://github.com/cujojs/when 一个示例  http://blog.sina.com.cn/s/blog\_537517170102w1w8.html 5、orm  持久化框架 1> orm2       项目地址 https://github.com/dresende/node-orm2 6、KOA ，据说可以取代express    详情 http://koa.bootcss.com/ |

# 开发

## 插件

Morgan: HTTP request logger middleware.

body-parser: body-parsing middleware that is used to parse the request body, and it supports various request types.

method-override: provides HTTP verb support such as PUT or DELETE in places where the client doesn't support it.

Compression: a compression middleware that is used to compress the response data using gzip/deflate.

express.static: This middleware used to serve static files.

cookie-parser: a cookie-parsing middleware that populates the req.cookies object.

Session: a session middleware used to support persistent sessions.

**自带插件页面**

[**https://github.com/senchalabs/connect#middleware**](https://github.com/senchalabs/connect#middleware)

**第三方插件页面**

[**https://github.com/senchalabs/connect/wiki**](https://github.com/senchalabs/connect/wiki)**.**

**测试数据：**JSON:

{

"firstName": "First",

"lastName": "Last",

"email": "user@example.com",

"username": "username",

"password": "password"

}

**测试方法：**

**$ curl -X POST -H "Content-Type: application/json" -d '{"firstName":"First", "lastName":"Last","email":"user@example.com","username":"username","password":"password"}' localhost:3000/users**