

零、Nodejs安装

Node.js 安装包及源码下载地址为：<https://nodejs.org/en/download/>。

msi安装

LTS
Recommended For Most Users

Current
Latest Features

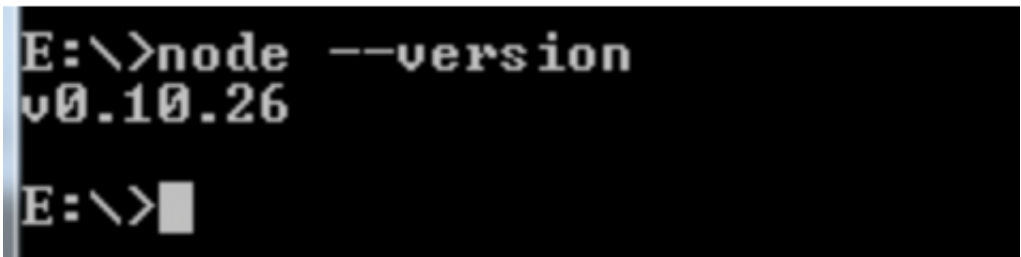

Windows Installer
node-v16.13.2-x86.msi


macOS Installer
node-v16.13.2.pkg


Source Code
node-v16.13.2.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.13.2.tar.gz	

检查Node.js版本



二进制：64 位安装包下载地址：<http://nodejs.org/dist/v0.10.26/x64/node.exe>

NPM命令

NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：

- 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
- 允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
- 允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。

由于新版的nodejs已经集成了npm，所以之前npm也一并安装好了。同样可以通过输入 **"npm -v"** 来测试是否成功安装。命令如下，出现版本提示表示安装成功：

```
$ npm -v
2.3.0
```

如果你安装的是旧版本的 npm，可以很容易得通过 npm 命令来升级，命令如下：

```
$ sudo npm install npm -g
/usr/local/bin/npm -> /usr/local/lib/node_modules/npm/bin/npm-cli.js
npm@2.14.2 /usr/local/lib/node_modules/npm
```

如果是 Window 系统使用以下命令即可：

```
npm install npm -g
```

一、安装json-server模块

```
npm install -g json-server
```

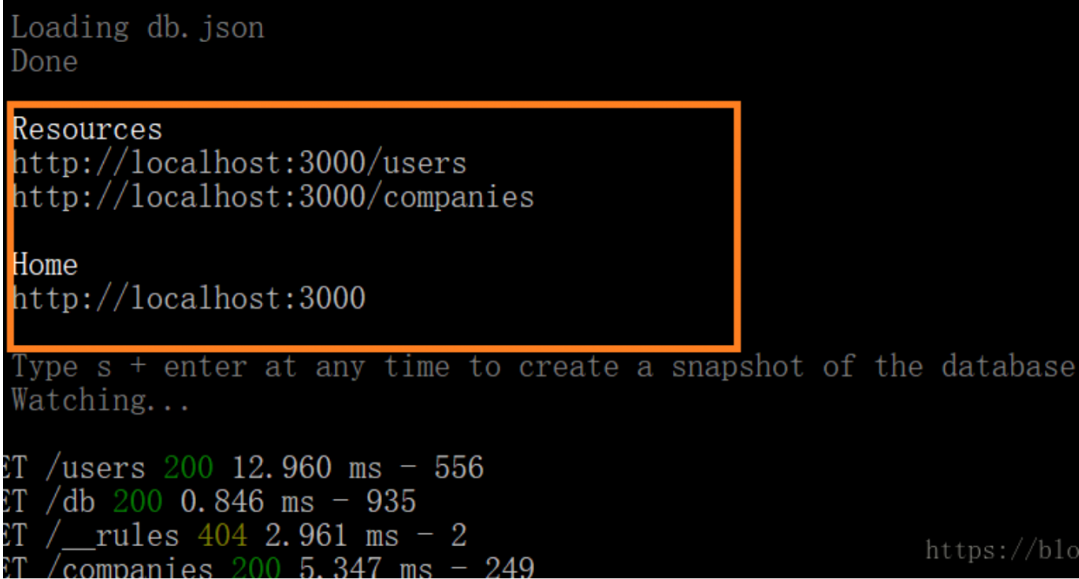
注意：window无需密码，mac加sudo。

二、在任意目录新建文件 `db.json`，拷贝如下内容：

```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ],
  "profile": { "name": "typicode" }
}
```

三、进入该目录，使用命令启动json-server。如下图所示启动成功。

```
json-server --watch db.json
```



```
Loading db.json
Done
Resources
http://localhost:3000/users
http://localhost:3000/companies
Home
http://localhost:3000
Type s + enter at any time to create a snapshot of the database
Watching...
ET /users 200 12.960 ms - 556
ET /db 200 0.846 ms - 935
ET /__rules 404 2.961 ms - 2
ET /companies 200 5.347 ms - 249
```

The screenshot shows a terminal window with a black background. The text is white and green. A yellow rectangle highlights the 'Resources' section, which lists two URLs: 'http://localhost:3000/users' and 'http://localhost:3000/companies'. Below this, the 'Home' URL 'http://localhost:3000' is shown. At the bottom, there is a list of HTTP requests and their responses, including status codes and response times.

<https://blog.csdn.net/lhjuejiang>

四、在浏览器访问地址：`http://localhost:3000`

- Resources中显示的是可以查看的json数据（数据表）。
- 下面是可以支持的restFul访问方式，具体分类为：
 - get是查询
 - post是添加
 - put是全体修改
 - patch是局部修改
 - delete是删除
 - options预检请求

五、可以使用get方式查询数据。例如：`http://localhost:3000/posts/1`

其他查询方式

```
1 //获取所有用户信息
2 http://localhost:3000/users
3 //获取id为1的用户信息
4 http://localhost:3000/users/1
5 //获取公司的所有信息
6 http://localhost:3000/companies
7 //获取单个公司的信息
8 http://localhost:3000/companies/1
9 //获取所有公司id为3的用户
10 http://localhost:3000/companies/3/users
11 //根据公司名字获取信息
12 http://localhost:3000/companies?name=Google
13 //根据多个名字获取公司信息
14 http://localhost:3000/companies?name=Google&name=Apple
15 //获取一页中只有两条数据
16 http://localhost:3000/companies?_page=1&_limit=2
17 //升序排序 desc降序 asc升序
18 http://localhost:3000/companies?_sort=name&_order=asc
19 //获取年龄为30以及30以上的
20 http://localhost:3000/users?age_gte=30
21 //获取年龄为30到40之间的
22 http://localhost:3000/users?age_gte=30&age_gte=40
23 //搜索用户信息
24 http://localhost:3000/users?q=d
```

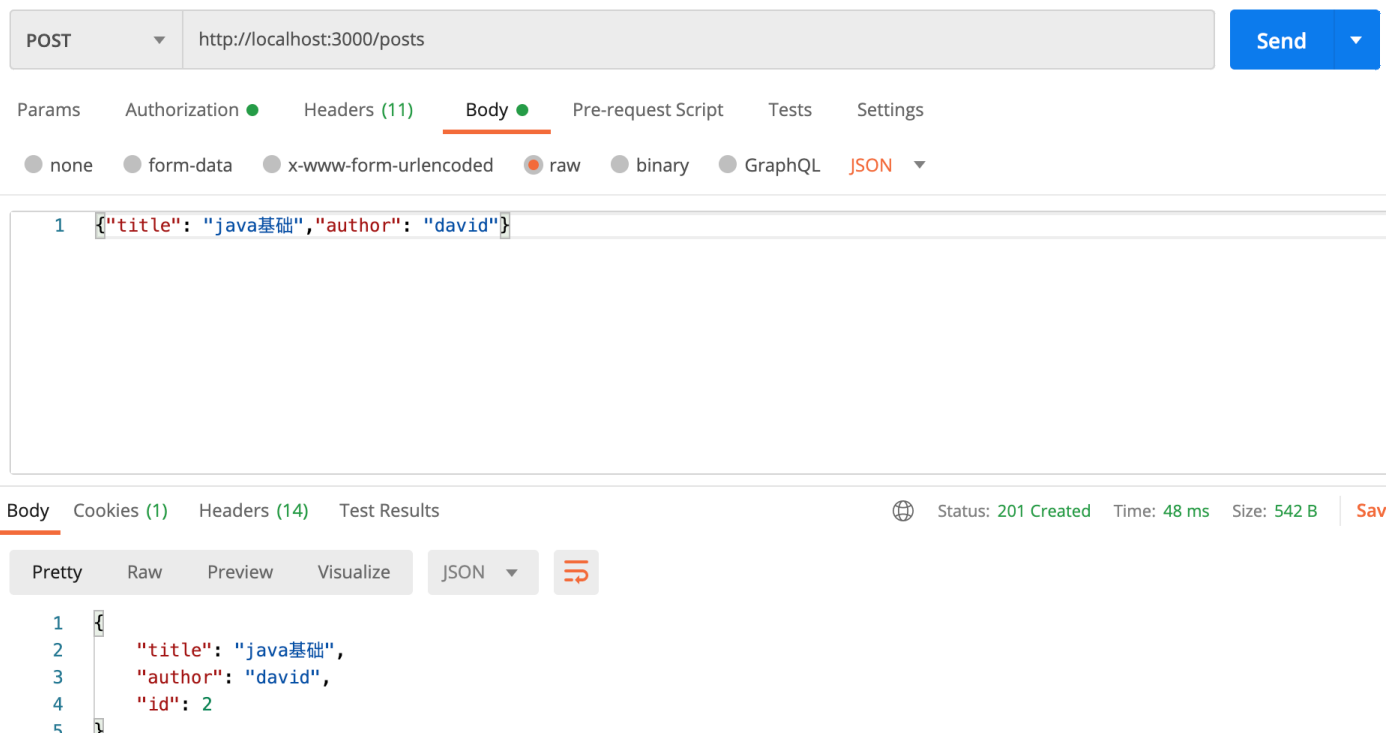
1. <http://localhost:3000/db> 访问的是db.json文件下的所有内容；
2. <http://localhost:3000/layoutList?categoryName=> 模拟接口参数可筛选该目录下内容
3. 分页查询 参数为 *start*, *_end*, *_limit*, 并可添加其它参数筛选条件
如: <http://localhost:3000/posts?start=6&limit=3>
http://localhost:3000/posts?start=3&_end=6

- 4.
5. 排序 参数为sort, order
如: <http://localhost:3000/posts?sort=id&order=asc>
http://localhost:3000/posts?sort=user,views&_order=desc,asc
6. 操作符 _gte, _lte, _ne, _like
_gte大于, lte小于, _ne非, _like模糊查询
7. q全局搜索 (模糊查询)

六、使用postman实现其他请求方式操作。(增删改)

添加:

- 地址写为表名。name=david&pass=123
- post方式为添加。
- body 选择为 raw。
- 内容可以不写id选项, 添加自动生成。



删除:

- 地址写为表名。
- delete方式为删除。
- 内容可以不写id选项, 添加自动生成。

DELETE

http://localhost:3000/posts/1

Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

● none● form-data● x-www-form-urlencoded● raw● binary● GraphQLJSON

1

BodyCookies (1)Headers (12)Test Results

⌐ Status: 200 OKTime: 32 msSize: 372 B

Sav

PrettyRawPreviewVisualizeJSON

1 {}

修改：

- 地址写为表名。
- patch方式为修改（put也可以）。
- body 选择为 raw。
- 内容可以无id选项，但是url中必须制定id。

PATCH

http://localhost:3000/posts/2

Send

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

● none● form-data● x-www-form-urlencoded● raw● binary● GraphQLJSON

1 {"title": "java基础", "author": "king"}

BodyCookies (1)Headers (12)Test Results

⌐ Status: 200 OKTime: 17 msSize: 430 B

Sav

PrettyRawPreviewVisualizeJSON

1 {
2 "title": "java基础",
3 "author": "king",
4 "id": 2
5 }

<https://www.npmjs.com/package/json-server#add-custom-routes>