学习nodejs

<http://nodejs.cn/>

# NPM包管理器

在 2017 年 1 月时，npm 仓库中就已有超过 350000 个软件包，这使其成为世界上最大的单一语言代码仓库，并且可以确定几乎有可用于一切的软件包。

node –v 查看node 版本

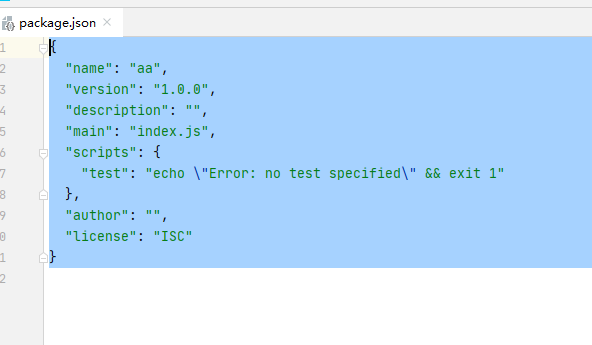
npm init 初始化一个package.json包描述文件

node main.js 运行一个node模块

## package.json

package.json 项目描述文件，可以手动创建，也可以使用命令更具提示自动创建。

package.json 文件其实就是对项目或者模块包的描述，里面包含许多元信息。比如项目名称，项目版本，项目执行入口文件，项目贡献者等等。npm install 命令会根据这个文件下载所有依赖模块。



package.json文件配置详解

|  |
| --- |
| { //项目名称 “name”: “demo”, //version是版本（遵守“大版本.次要版本.小版本”的格式） “version”: “1.0.0”, //description描述你的模块,或者搜索 “description”: “vue.js js iview “, //main字段指定了加载的入口文件 “main”: “app.js”, // scripts指定了运行脚本命令的npm命令行缩写，比比如start指定了运行npm run start时，所要执行的命令。 “scripts”: { “start”: “node index.js” }, //repository(仓库)指定一个代码存放地址 “repository”: { “type”: “git”, “url”: “git+<https://github.com/XXXX>” }, “author”: “mayuan”, “license”: “ISC”, “bugs”: { “url”: “<https://github.com/XXXX>” }, // 一个字符串数组，方便别人搜索到本模块 “keywords”: [ “vue”,”iview” ], “entry”:{ “main”: “./web/index.js” }, //项目主页url “homepage”: “<https://github.com/XXXX>“, //devDependencies指定项目开发所需要的模块 “devDependencies”: { “babel-core”: “^6.23.1”, “babel-loader”: “^6.3.2”, “babel-preset-es2015”: “^6.22.0”, “vue-html-loader”: “^1.0.0”, “vue-loader”: “^8.5.2”, “vue-style-loader”: “^1.0.0”, “webpack”: “^1.13.2” }， //dependencies字段指定了项目运行所依赖的模块 “dependencies”: { “underscore”: “^1.8.3”, “vue”: “^2.1.4” }, } |

## package-lock.json

锁定安装时的包的版本号，并且需要上传到git，以保证其他人在npm install时大家的依赖能保证一致。

## NPM常用命令

### 安装一个包

npm i express express是包名。

### 更新一个包

npm update express

### 卸载一个包

npm uninstall express

### 全局安装一个包

npm install express -g

### 在项目描述文件中添加依赖记录

npm install express -save

--save 安装并添加条目到 package.json 文件的 dependencies。

--save-dev 安装并添加条目到 package.json 文件的 devDependencies。

### 全局安装添加依赖记录

npm install express -g -save

## NPM切换到淘宝镜像

由于 npm 安装速度慢，本教程使用了淘宝的镜像及其命令 cnpm

切换命令：

npm install -g cnpm --registry=https://registry.npm.taobao.org

## 切换为官方的NPM

npm config delete registry

# 使用模版引擎合并数据

模板引擎有很多，ejs 是其中一种，因为它使用起来十分简单，而且与 express 集成良好，

所以我们使用 ejs。

**安装 ejs： npm i ejs –save**

通过调用 res.render 函数渲染 ejs 模板，res.render 第一个参数是模板的名字，第二个参数是一个对象，里面是渲染模版需要的数据。

## 模板文件

<!DOCTYPE html>  
 <**html** lang=**"en"**>  
 <**head**>  
 <**meta** charset=**"UTF-8"**>  
 <**title**>XXX欢迎您</**title**>  
 </**head**>  
 <**body**>  
 <**h1**>你的ip是:**<%=myip%>**</**h1**>  
 </**body**>  
 </**html**>

## 数据内容

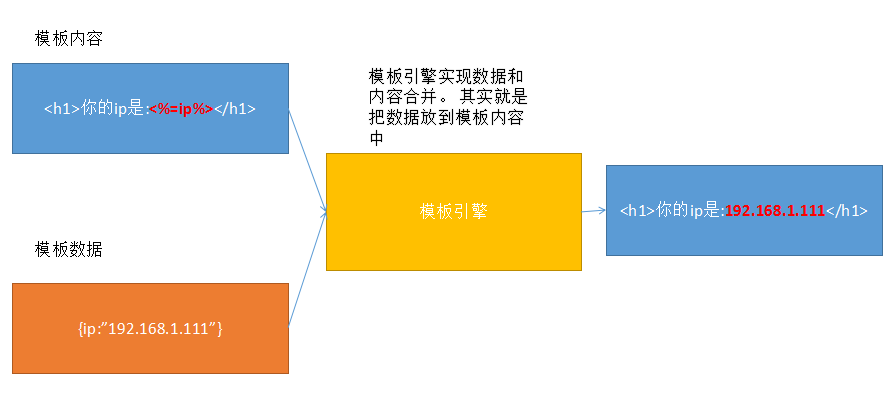
{

myip:“192.168.1.112”,

name:”王小花”

}

## 模板引擎合并生成数据



*//ejs模板引擎***var** ejs = require(**"ejs"**);  
*//模板内容和数据进行合并*ejs.renderFile(**"templateDemo.ejs"**,{**ip**:**"192.168.1.222"**},

**function**(err, str){  
 **console**.log(str);  
});

## 三种常用标签

这个 <%= xxx %> 是什么东西呢？

ejs 有 3 种常用标签：

<% code %>：运行 JavaScript 代码，不输出

<%= code %>：显示转义后的 HTML内容

<%- code %>：显示原始 HTML 内容

注意：<%= code %> 和 <%- code %> 都可以是 JavaScript 表达式生成的字符串，

当变量 code 为普通字符串时，两者没有区别。

当 code 比如为 <h1>hello</h1> 这种字符串时，

<%= code %>  会原样输出 <h1>hello</h1>，

<%- code %>  则会显示 H1 大的 hello 字符串。

完整例子

模板代码：

day04-template.ejs

|  |
| --- |
| <!DOCTYPE html> <**html** lang=**"en"**> <**head**>  <**meta** charset=**"UTF-8"**>  <**title**>欢迎您</**title**> </**head**> <**body**> 欢迎您<%=ip%>，性别<%=sex%> 你的爱好是: <% **for**(**var *i***=0;***i***<hobbys.**length**;***i***++) { %> <**h1**> <%=hobbys[***i***]%></**h1**> <% } %> </**body**> </**html**> |

后端代码：

|  |
| --- |
| **var *http*** = ***require***(**"http"**); **var *ejs*** = ***require***(**"ejs"**); **var *server*** = ***http***.createServer(**function**(req,res){  *//每个客户端请求的ip都不一样* **var** ip = req.socket.remoteAddress;  res.writeHead(200,{  **"Content-Type"**:**"text/html;charset=utf-8"** });  *//res.write("123");  //res.end();* ***ejs***.renderFile(**"template.ejs"**,{**"ip"**:ip,**"age"**:20,**"sex"**:**"男"**,**hobbys**:[**"写代码"**,**"看代码"**,**"抄代码"**]},**function**(err,data){  **if**(!err){  *//把合成后内容输出到浏览器。* res.write(data);  res.end();  }**else**{  *// console.log(err.toString());* res.write(err.toString());  res.end();  }  });  }); ***server***.listen(3000); |

# Express 介绍

## Express介绍

Express 是一个基于 **Node.js 平台的极简、灵活的 web 应用开发框架（具有一定结构和功能的代码）**，它提供一系列强大的特性，帮助你创建各种 Web 和移动设备应用。

敏捷开发。

Express 框架核心特性：

* 可以设置**中间件**来响应 HTTP 请求。
* 定义了**路由表**用于执行不同的 HTTP 请求动作。

（url = 资源） 映射

* 可以通过向模板传递参数来动态渲染 HTML 页面。

## 使用步骤

1. 打开cmd ,输入npm i **express-generator -g ，全局安装express-generator 。**
2. **在D盘根目录新建一个文件夹code**
3. **Cd进入到这个目录。**
4. **express -e 创建一个express项目，采用的模板是ejs.**
5. **用webstorm打开项目，运行bin目录下面的www文件**

**6、执行npm i 命令，安装依赖的包。**

**7、访问127.0.0.1:3000**

## 新增一个新闻模块

|  |
| --- |
| var express = require('express'); var router = express.*Router*();  */\* GET home page. \*/* router.get('/', function(req, res, next) {  res.render('newsadd', { title: 'Express' }); }); router.get('/newsadd/', function(req, res, next) {  res.render('newsadd', { title: 'Express' }); });  module.exports = router; |

## 主模块挂载和路由配置

