**Express**

**1、简介**

（1）Express 是一个基于 Node.js 平台的极简、灵活的 web 应用开发框架，它提供一系列强大的特性，帮助你创建各种 Web 应用(相当于原生js与jQuery的关系)

（2）使用 Express 可以快速地搭建一个完整功能的网站（通过express-generator 脚手架 快速的构建express项目）

**2、express的使用及安装（安装到项目中）**

（1）创建一个js文件 如：app.js

（2）创建包package.json 命令行输入cnpm init

（3）安装express 命令行输入cnpm install --save express

（4）创建服务器

（5）post请求时需要导入处理post请求的中间件

命令行输入cnpm install --save body-parser

cdn 表示请求别人服务器的资源

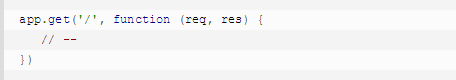


express框架实例1

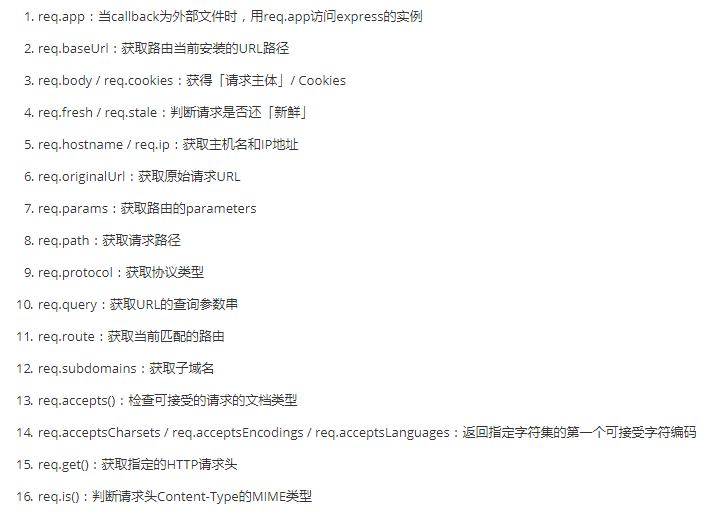


## 3、请求和响应

(1)Express 应用使用回调函数的参数： request 和 response 对象来处理请求和响应的数据



1)Request 对象 - request 对象表示 HTTP 请求，包含了请求查询字符串，参数，内容，HTTP 头部等属性。常见属性有：



2)Response 对象 - response 对象表示 HTTP 响应，即在接收到请求时向客户端发送的 HTTP 响应数据。常见属性有：



## 4、路由

介绍：在HTTP请求中，路由决定了由谁(指定脚本)去响应客户端请求（及不同的路径访问不同的内容）

1. git请求如下：



**根据以上创建好js文件后，在浏览器地址栏输入localhost:8888/相应的路径 如：localhost:8888/del\_user 页面会输出 删除页面四字…**

1. post请求如下：

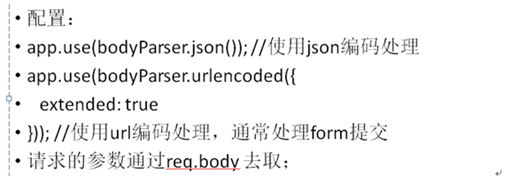
（1）利用body-parser插件

命令行输入cnpm install --save body-parser

（2）加载静态文件

1）Express 提供了内置的中间件 express.static 来设置静态文件如：图片， CSS, JavaScript 等





****

****

****

## 5、自动重启服务器工具node supervisor

（1）创建node supervisor

命令行输入cnpm install –g node-supervisor

1. 启动

命令行输入supervisor js文件名 如：supervisor app.js

**6、调试工具**

（1）第一种方法

1. 安装 npm install -g node-inspector
2. 再启动项目，添加 --debug 参数 如: node --debug-brk app2.js
3. 再新开cmd，输入node-inspector & ，执行成功后会显示一个地址，使用浏览器打开这个地址，浏览器需要是Chrome
4. 第二种方法
5. 浏览器打开网页

地址chrome://flags/#enable-devtools-experiments

2）点击开启（#enable-devtools-experiments）---再重启---命令行输入node --inspect app ---- F12 选中source---点击右边的connect

**7、NodeJS通过mongoose框架连接mongoDB数据库**



**Nodejs创建app.js服务器并连接mongoDB数据库：**

1、引入需要的各个库，如：QQ截图20170621153940.png

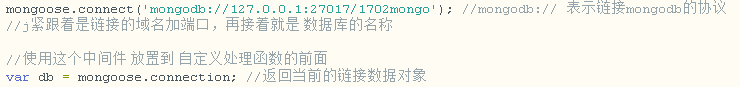
2、创建express实例：

QQ截图20170621153940.png

1. 引入model方法创建的用户集合模型：

QQ截图20170621153940.png

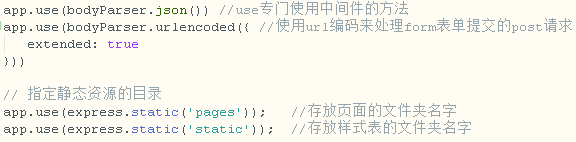
1. 通过mongoos的connect方法连接到指定的数据库，并用connection返回一个当前对象：



使用open事件测试是否连接成功：

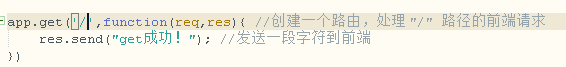
QQ截图20170621153940.png

1. 通过固定的USE语句来使用中间件body-parser加载post的和form表单的请求。并指定静态资源的目录：



1. express实例的get和post 方法：

get：



Post：



1. 添加监听端口：

QQ截图20170621153940.png

**Nodejs使用ejs模板加载：**

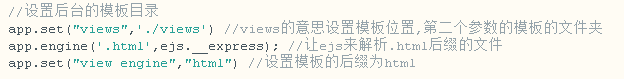
安装模板： npm install --save ejs

在app.js中引入中间件： ejs=require(“ejs”)

在工程文件夹下创建一个views文件夹，存放后台的模板和其他的一些页面。这里将部分静态页面放入这个文件夹中就是将逻辑判断由前端判断跳转转换为后端判断。

模板采用HTML文件，相当于一个通过ejs模板写的静态的html网页，用来动态加载各数据。

在app.js中使用ejs中间件方法加载模板：



由于大部分编辑器和浏览器不认识ejs文件，所以采用set方法将模板的后缀设置为html文件来解析。

这里有两个方法来进行后台的跳转：

res.render('comment',{username:username}) //向响应端发送comment.html 并渲染。Comment是由ejs模板编写，第二个参数是在模板中传入变量为username的参数。

res.redirect('/comment'); //在已经渲染了的模板页面中，重新进行跳转。

**Nodejs使用session保存当前用户数据：**

安装模板： npm install --save express-session

在app.js中引入中间件： ejs=require(“express-session”)

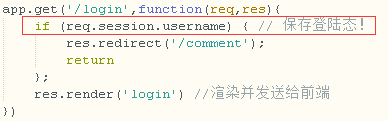
使用use方法调用session中间件，secret是自己配置的一个字符串，结合保存的时间等其他条件生成一个唯一的session—cookie码；maxAge是这个session在当没有操作的情况下保存的时间，单位是毫秒：



在需要保存的地方，进行session保存。Session会被保存在res响应部分中：



Session也可以作为判断条件，看是否有session保存的数据：



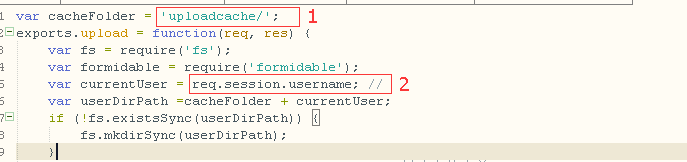
**Nodejs中使用fromdata第三方插件来保存上传的图片：**

在html中只有form表单才能提交二进制流文件，我们可以通过ajax异步上传和使用FromData第三方插件来上传，其大概原理是将图片外包一个from进行异步上传。

先下载安装fromdata插件： npm install --save formdata

在工程文件下创建一个uploadcache文件夹来存放上传的图片，一般图片都存放在另外的位置上，而数据库保存的只是图片的路径。

然后在工程项目文件夹中创建一个util文件夹，保存upload.js 文件，这个文件保存的是上传图片的方法函数：upload在这个文件中需要设置2个基本配置：

****

1处是保存上传图片的路径。

2处是当前用户上传的一个属性值。

调用方法：

QQ截图20170623165537.png

在前端页面上传时使用如下格式：



**Nodejs中使用websocket：**

WebSocket是HTML5中新增协议，是一个基于TCP的协议，它由通信协议和编程API组成，WebSocket能够在浏览器和服务器之间建立双向连接， 以基于事件的方式，赋予浏览器实时通信能力。既然是双向通信，就意味着服务器端和客户端可以同时发送并响应请求，而不再像HTTP的请求和响应。

Websocket的安装： npm i --save ws

再安装socket.io的方法： npm i --save socket.io

安装express : npm i express --save

ssh-keygen -t rsa -b 4096 -C "zhanghaohbt@163.com"