# Express框架

基于 [Node.js](https://nodejs.org/en/) 平台，快速、开放、极简的 Web 开发框架

//安装

//npm install express --save

//引包

var express = require('express');

//创建你的服务器应用程序（也就是原来的http.createServer()）

var app = express();

//当服务器收到get请求 / 的时候，执行回调函数

app.get('/',function(req,res){

    // res.end('hello express!!!');

    res.end(`

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <title>Document</title>

    </head>

    <body>

        <h2>hello express 你好</h2>

    </body>

    </html>

    `)

});

app.get('/about',function(req,res){

    //express框架不需要解析中文

    // res.setHeader('Content-type','text/html;charset=utf-8');

    res.end('我是 about!!!');

});

//公开指定目录

//这样就可以直接通过/public/xx的方式访问public目录下的所有资源

//public/js/main.js直接访问

//一个express中开放的资源就是一个API

app.use('/public/',express.static('./public/'));

app.use('/node\_modules/',express.static('./node\_modules/'));

app.listen(8080,function(){

    console.log('http://localhost:8080');

});

# 路由

## Get

//当你以get方法请求/的时候，执行对应的处理函数

App.get(‘/’,function(req,res){

Res.end(‘hello world!!!’);

})

## Post

//当你以post方法请求/的时候，执行对应的处理函数

App.post(‘/’,function(req,res){

Res.end(‘hello world!!!’);

})

# 静态服务

//公开指定目录

//方法1

//这样就可以直接通过/public/xx的方式访问public目录下的所有资源

//public/js/main.js直接访问

//一个express中开放的资源就是一个API

app.use('/public/',express.static('./public/'));

app.use('/node\_modules/',express.static('./node\_modules/'));

//方法2

//访问资源的时候直接/a/js/main.js 就可以，a相当于public的别名

app.use('/a/',express.static('./public/'));

//方法3

//当省略第一个参数的时候

//访问资源的时候直接js/main.js 就可以，可以省略public

app.use(express.static('./public/'));

# Express中配置使用art-template模板引擎

<https://aui.github.io/art-template/>

## 安装：

npm install art-template --save

npm install express-art-template --save

## 配置

//配置使用art-template模板引擎

//第一参数表示，当渲染以.art,.html结尾的文件的时候，使用art-template模板引擎

// express-art-template是专门用来在express中把art-template整合到express中

//安装

//npm install art-template --save  安装模板引擎

//npm install express-art-template --save

// app.engine('art',require('express-art-template'))

 app.engine('html',require('express-art-template'))

## 使用

//express为response响应对象提供了一个方法：render

//render方法默认是不可以使用的，但是安装了模板引擎就可以使用

//render('html模板名',{模板数据})

//第一个参数不能写路径，默认会去项目中views目录查找该模板文件

//所有的视图文件都放在views目录中

//如果想要修改默认的views，则可以 app.set('views',render函数的默认路径)

App.get(‘/’,function(req,res){

Res.render(‘index.html’,{

Title:’hello world’

})

})

app.get('/doadd',function(req,res){

var mag = req.query //获取用户输入的对象

mags.push(mag) //尾部插入数据

//重定向 必须的

// res.statusCode = 302;

// res.setHeader('localhost','/')

res.redirect('/')

})

# 在express获取表单GET请求数据

在express中有内置获取表单GET请求体的API,直接res.query获取

# 在express获取表单POST请求数据

在express中没有内置获取表单POST请求体的API，这里我们需要使用一个第三方包：body-parser

## 下载安装包

npm install body-parser

## 配置

var express -requires(‘express’);

//引包

var bodyParser = require(‘body-parser’);

//服务器

var app =express();

//配置body-parser 插件

//只要加入这个配置，则在req请求对象会多出一个属性：body

//也就是说可以直接通过req.body来获取表单POST请求体中的数据了

app.use(bodyParser.urlencoded({ extended:false }));

app.use(bodyParser.json());

## 使用

app.use(function(req,res){

res.setHeader(‘Content-Type’,’texl/plain’)

res.write(‘hello world’)

//可以通过req.body来获取POST请求体数据

Res.end(JSON.stringfy(req.body ,null ,2 ))

})

app.post(‘/post’,function(req,res){

Console.log(req.body);//用户输入的数据

})

## 案例 （提交的表单为POST）

