

Node.js 企业级实战在线课(从入门到精通) - 第四章

概述

1. Express框架的基本使用
2. 路由的概念与设置方式
3. 中间件的概念与设置方式
4. 模块supervisor的使用
5. 项目构造器express-generator

Express框架的基本使用

概念

1. Express框架是基于Node.js平台的快速、开放、极简的web开发框架
2. 良好的路由机制、丰富的HTTP快捷方法和任意排列组合的Connect中间件
3. Express在Node.js基础上进行了web所需功能扩展
4. 良好的社区以及案例应用环境

下载与基本使用

```
// 下载模块
npm install express

// 加载模块
var express = require('express');

// 创建应用实例
var app = express();

// 设置首页路由
app.get('/', function(req, res){
    // req 请求
    // res 响应
});

// 设置监听的端口号和监听IP地址
app.listen(port, IP, callback);
```

视图模板引擎 ejs

```
// 下载模块
npm install ejs

// 加载模块
var ejs = require('ejs');
var path = require('path');

// 设置ejs模板引擎
app.set('view engine', ejs);
// 设置模板存储目录
app.set('views', path.join(__dirname, 'views'));

// 响应视图模板
res.render('index');
```

静态资源托管

```
// express.static() 中间件的使用
app.use(express.static(path.join(__dirname, 'public')));
```

注：在模板视图文件中加载静态资源时 / 就表示public目录

路由的概念与设置方式

1. 路由是指如何定义应用的端点以及如何响应客户端的请求
2. 路由由URL和HTTP请求(GET/POST)和若干个句柄组成

路由方法

```
// 常用方法有get和post两种
app.get()           // 接收get方式的请求
app.post()          // 接收post方式的请求
```

路由路径

注：就是大家约定好的字符串访问方式（我说鸡蛋你说要）

```
// 设置字符串路径
app.get('/about', callback)
// 字符串模式路由
app.get('/ab?', callback)
// 正则表达式路由
app.get(/a/, callback)
```

路由句柄

```
// 允许设置一个或多个句柄处理该路由，指定多个路由时注意移交处理(next)
app.get(rouoter,cbA,cbB...)
```

中间件的概念与设置方式

概念

中间件(Middleware)是一个函数，可以访问请求对象(req),响应对象(res)和web应用中处于请求-响应循环流程中的中间件，一般命名为next变量

功能

1. 执行任何代码
2. 修改请求和响应对象
3. 终止请求和响应循环
4. 调用堆栈中的另一个中间件

中间件类型划分

1. 应用级中间件
2. 路由级中间件
3. 错误处理中间件
4. 内置中间件
5. 第三方中间件

supervisor模块

概念：supervisor模块是一个监测文件变化的模块

下载安装

```
// 为在任何一个目录都可以使用该模块，进行全局安装
npm install supervisor -g
```

使用

```
supervisor node app.js
```

项目构造器 **express-generator**

概述：该模块是用于快速创建express框架项目结构的模块

下载安装

```
// 全局安装  
npm install express-generator -g
```

基本使用

生成项目结构

```
// 使用express命令快速创建项目结构，项目文件夹名称blog，使用模板引擎为ejs  
express --view=ejs blog
```

进入目录，下载依赖包

```
cd blog && npm install
```

运行项目

```
// 默认项目在本机的3000端口运行  
npm start
```