1.示例项目库

mfresh.codeboy.com

源码：mfresh.codeboy.com/mfresh.zip

m.mfresh.codeboy.com

源码：m.mfresh.codeboy.com/mfresh-m.zip

--------------------------------------------------

iweb.codeboy.com

源码：iweb.codeboy.com/iweb.zip

m.iweb.codeboy.com

源码：m.iweb.codeboy.com/iweb-m.zip

---------------------------------------------------

ulift.codeboy.com

源码：ulift.codeboy.com/ulift.zip

m.ulift.codeboy.com

源码：m.ulift.codeboy.com/ulift-m.zip

---------------------------------------------------

irose.codeboy.com

源码：irose.codeboy.com/irose.zip

m.irose.codeboy.com

源码：m.irose.codeboy.com/irose-m.zip

---------------------------------------------------

学员作品库：www.codeboy.com/pro

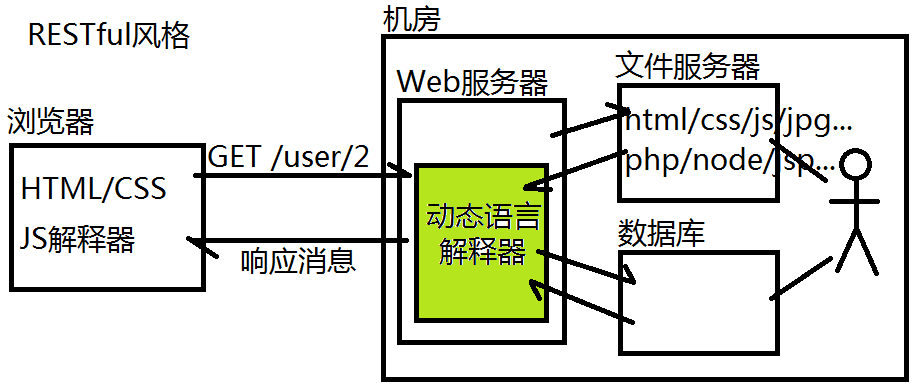
80%代码 <=> 20%时间

20%代码 <=> 80%时间

2.项目介绍

预览：mfresh.codeboy.com

技术：mysql + node + vue



|  |  |
| --- | --- |
| JS和Node.js区别： | |
| JS | Node.js |
| 客户端/浏览器技术 | 服务器端技术，竞争对手是PHP/JSP |
| 有严重的兼容性问题 | 没有兼容性问题(ES6/7) |
| 数据类型：  值类型：...  引用类型：  ES原生对象  宿主/浏览器对象(B&D)  自定义对象 | 数据类型：  值类型：...  引用类型：  ES原生对象  Node.js扩展对象(手册)  自定义对象 |

学习Node.js：

(1)架构体系——语法 (2)API中的对象——单词

3.Node.js核心模块——http模块

http模块用于创建HTTP协议的服务器端应用(代替Apache+PHP功能)——接收客户端请求，解析并处理请求，返回响应消息

http模块还用于创建HTTP协议的客户端应用(模拟浏览器功能)——向其他的服务器发起请求，等待返回响应消息

Node.js中使用http模块创建服务器端应用：

const http = require('http');

let server = http.createServer(function(req, res){

//读取请求消息 req.method/url/headers

//输出响应消息 res.writeHead()/write()/end()

});

server.listen(80);

午间练习：使用HTTP模块接收不同的请求URL，服务器返回不同的响应消息主体

/index

/user/login

/user/register

原生DOM操作(复杂) => 第三方库jQuery

Node.js http模块(复杂) => 第三方框架Express/Koa

4. Node.js http模块的第三方扩展框架——Express

官网：expressjs.com

中文网：expressjs.com.cn

Express本质是一个HTTP请求消息处理函数，可以解析请求消息，处理请求数据，向客户端输出响应消息

使用步骤：

(1)使用NPM工具到npmjs.org下载最新的Express模块

npm i express

(2)使用http模块创建Web服务器，运行express，让其承担请求监听器的任务

var app = express();

http.createServer( app ).listen(80);

(3)为app对象添加特定URL的请求处理

....

|  |
| --- |
| Express三大核心概念：  (1)路由  (2)中间件  (3)模板引擎 |

5.Express核心概念之一： 路由

Route：路由，针对客户端的某个请求方法所请求的特定URL，如何给出响应消息的过程，称为“路由”

**路由 = 请求方法 + 请求URL + 处理函数**

一个典型的路由形如：

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

//接收请求消息中的数据

req.query //获取查询字符串中的数据{ }

req.params //获取请求参数(手册自学)

req.on('data', function(buf){}) //获取请求主体中的数据

//发送响应消息

res.send(HTML文本);

res.sendFile(任意类型的文件);

res.json(对象/数组数据);

})

课后任务：

创建Node.js/Express服务器端应用，仿照mfresh.codeboy.com实现其中的如下功能点：

GET /index

GET /contact

GET /register

POST /register

GET /login

POST /login

必需的静态资源可以直接从codeboy.com服务器上抓取