Node 应用开发

Node 简介

Node 是服务器的 JavaScript 运行环境,提供 API 与操作系统互动。

主要用途:

- 开发前端应用
- 快速搭建服务
- 架设网站

npm

安装 Node 的时候,会同时安装 npm。

\$ npm -v

它是 Node 的模块管理器,开发 Node 项目的必备工具。

课堂练习:Node 的简单应用

进入 demos/simple-app-demo 目录,参考《操作指南》,自己动手在 Node 里面,编写并编译一个前端脚本。

Node 开发前端脚本的好处

- 1. 模块机制
- 2. 版本管理
- 3. 对外发布
- 4. 持续集成的标准开发流程

REST API

REST 是浏览器与服务器通信方式的一种设计风格。

它的全称是"REpresentational State Transfer",中文意为"表现层状态转换"。

• Resource:资源

• Representation: 表现层

• State: 状态

• Transfer:转换

REST 的核心概念

- 1. 互联网上所有可以访问的内容,都是资源。
- 2. 服务器保存资源,客户端请求资源。
- 3. 同一个资源,有多种表现形式。
- 4. 协议本身不带有状态,通信时客户端必须通过参数,表示请求 不同状态的资源。
- 5. 状态转换通过HTTP动词表示。

URL 设计

URL 是资源的唯一识别符。

- /store/1
- /store/2
- /store/1/employee/2

查询字符串

查询字符串表示对所请求资源的约束条件。

- GET /zoos/1/animals?limit=10
- GET /zoos/1/animals?limit=10&offset=10
- GET /animals?zoo_id=1

操作 SQL方法 HTTP动词

HTTP 动词

操作	SQL方法	HTTP动词
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT/PATCH
DELETE	DELETE	DELETE

GET /v1/stores/1234 PUT /v1/stores/1234 POST /v1/stores DELETE /v1/stores/1234

课堂练习:REST API

打开 demos/rest-api-demo ,按照《操作指南》,熟悉 REST API 的基本用法。

Express

Express 是最常用的 Node 框架,用来搭建 Web 应用。



课堂练习: Express 搭建 Web 应用

进入 demos/express-demo 目录,按照《操作指南》,学习使用 Express 搭建 Web 应用。

定义一个 Web 应用实例,并且启动它。

```
var express = require('express');
var app = express();

var port = process.env.PORT || 8080;

app.listen(port);
console.log('Magic happens on port ' + port);
```

定义一个路由

```
var router = express.Router();

router.get('/', function(req, res) {
  res.send('<h1>Hello World</h1>');
});

app.use('/home', router);
```

中间件:对HTTP请求进行加工。

```
router.use(function(req, res, next) {
  console.log('Thers is a requesting.');
  next();
});
```