复习

MVC （Model-View-Controller，模型视图控制器）是一种软件的设计模式，把一个复杂的软件工程分解为三个层面：模 型、视图和控制器。 

模型是对象及其数据结构的实现，通常包含数据库操作。 

视图表示用户界面，在网站中通常就是 HTML 的组织结构。 

控制器用于处理用户请求和数据流、复杂模型，将输出传递给视图。

1、认识express

Express 是Node.js的开发框架，因为它是目前稳定、使用广泛，而且 Node.js 官 方推荐的唯一一个 Web 开发框架。

对nodejs扩展了Web 应用所需的基本功能

功能：

 路由控制； 模板解析支持； 动态视图； 用户会话； CSRF 保护； 静态文件服务； 错误控制器； 访问日志； 缓存； 插件支持。

1. 安装express

npm install express node\_modules

安装并且构建一个项目目录

microblog

/package.json

/app.js

/public

/javascripts

/images

/stylesheets

/routes

/index.js

/views

/layout.ejs

/index.ejs

1. 简单配置一个express路由
2. 路由是什么

路由是指如何定义应用的端点（URIs）以及如何响应客户端的请求。

路由是由一个 URI、HTTP 请求（GET、POST等）和若干个句柄组成，它的结构如下： app.METHOD(path, [callback...])

1. 路由方法

GET：请求获取指定资源。 

HEAD：请求指定资源的响应头。 

POST：向指定资源提交数据。 

PUT：请求服务器存储一个资源。 

DELETE：请求服务器删除指定资源。 

TRACE：回显服务器收到的请求，主要用于测试或诊断。 

CONNECT：HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。  OPTIONS：返回服务器支持的HTTP请求方法。

主要用：

GET：获取  POST：新增  PUT：更新  DELETE：删除

路由句柄

express.Router 创建模块化、可挂载的路由句柄

app.route() 创建路由路径的链式路由句柄

1. 响应方法

res.json(); 发送一个JSON格式的响应

res.jsonp(); 发送一个支持 JSONP 的 JSON 格式的响应

res.download(); 提示下载文件

res.end(); 终结响应处理流程

res.redirect(); 重定向请求

res.render(); 渲染视图模版

res.send(); 发送各种类型的响应

res.sendFile(); 以八位字节流的形式发送文件

res.sendStatus(); 设置响应状态代码，并将其以字符串形式作为响应体的一部分发送

1. 使用中间件

Express 是一个自身功能极简，完全是由路由和中间件构成一个的 web 开发框架：从本质上来说，一个 Express 应用就是在调用各种中间件。

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

应用级中间件

应用级中间件绑定到 app 对象 使用 app.use() 和 app.METHOD()；

例如：app.use(function (req, res, next) {

console.log('Time:', Date.now());

next();

});

路由级中间件

路由级中间件

路由级中间件和应用级中间件一样，只是它绑定的对象为 express.Router()。

例如：router.use(function (req, res, next) {

console.log('Time:', Date.now());

next();

});

错误处理中间件

错误处理中间件有 4 个参数，定义错误处理中间件时必须使用这 4 个参数。即使不需要 next 对象，也必须在签名中声明它，否则中间件会被识别为一个常规中间件，不能处理错误。

例如：app.use(function(err, req, res, next) {

console.error(err.stack);

res.status(500).send('Something broke!');

});

内置中间件

例如：express.static(root, [options])

express.static 是 Express 唯一内置的中间件。它基于 serve-static，负责在 Express 应用中提托管静态资源。

第三方中间件

安装所需功能的 node 模块，并在应用中加载，可以在应用级加载，也可以在路由级加载。

例如：cookie-parser

$ npm install cookie-parser

1. ejs模版引擎

模板引擎（Template Engine）是一个从页面模板根据一定的规则生成 HTML 的工具。 动态生成 HTML.

模板引擎的功能是将页面模板和要显示的数据结合起来生成 HTML 页面。它既可以运 行在服务器端又可以运行在客户端，大多数时候它都在服务器端直接被解析为 HTML，解析 完成后再传输给客户端，因此客户端甚至无法判断页面是否是模板引擎生成的。

1. 模版引擎配合路由使用

app.set('views', \_\_dirname + '/views');

app.set('view engine', 'ejs');

res.render 的功能是调用模板引擎，

res.render('index', { title: 'Express' });

<h1><%= title %></h1>

<p>Welcome to <%= title %></p>

ejs 的标签系统非常简单，它只有以下3种标签。 

<% code %>：JavaScript 代码。 

<%= code %>：显示替换过 HTML 特殊字符的内容。 

<%- code %>：显示原始 HTML 内容。

我们可以用它们实现页面模板系统能实现的任何内容。

1. 搭建几个页面使用路由配置好
2. 错误处理

错误处理中间件