复习：

Node/Express：是一个第三方应用程序框架，本质是一个HTTP请求处理程序，负责接收/处理请求消息，返回响应消息。

Express三个核心概念：

(1)路由 = 请求方法 + 请求URL + 处理函数

(2)中间件

(3)模板引擎

1.中间件(Middleware)

Express中的中间件就是函数，形如：

function(req, res, next){

}

注意：路由处理函数本质也是一个中间件。

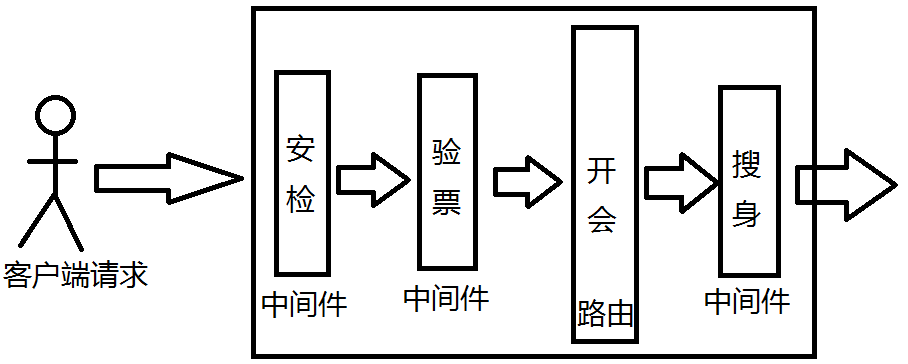
中间件的功能：

(1)执行任何代码。

(2)修改请求和响应对象。

(3)终结请求-响应循环。

(4)调用堆栈中的下一个中间件。



一个典型的“中间件栈”：

//前置中间件

app.use( function(req, res, next){ ...**next();** })

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

res.json(...);

**next();**

})

//后置中间件

app.use( function(req, res, next){ ...**next();** })

项目中常用的中间件

(1)向客户端返回指定目录下的所有静态资源文件

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

(2)将客户端POST请求主体中的数据封装到req.body中

npm i body-parser

const bodyParser = require('body-parser');

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

(3)将请求头部中的Cookie数据封装到req.cookies中

npm i cookie-parser

const cookieParser = require('cookie-parser');

app.use(cookieParser());

|  |
| --- |
| PHP: $\_REQUEST = $\_GET + $\_POST + $\_COOKIE |
| H5 sessionStorage技术：把k=v数据保存在客户端  服务器端session技术：把每个客户端会话数据保存在服务器端，客户端仅用Cookie保存一个SessionID号。$\_SESSION['loginName'] = 'tom'; |

(4)服务器的响应消息中发送Set-Cookie提供SessionID，客户端再次请求时可以自动提交该Cookie，出示自己的SessionID；服务器接收该SessionID，查找当前客户端对应的服务器端存储数据——封装在req.session对象中。

npm i express-session

const session = require('express-session');

app.use( session( {

resave: false,

saveUninitialized: true,

secret: 'xxxyyyyzzz'

} ) );

...

req.session

(5)使用cors中间件修改响应消息的头部，以允许指定来源的跨域请求

午间复习：

什么是跨域请求？如何实现跨域请求？CORS方案是什么？

http://www.ruanyifeng.com/blog/2016/04/cors.html

当前URL与目标URL的“协议名或主机名或端口”三者中有一项不同，目标URL请求就称为“跨域请求”。

浏览器允许跨域的情形： <img><script><link><iframe>

浏览器禁止跨域的情形： XHR请求

XHR跨域被禁止的解决方案：(1)JSONP (2)CORS (3)... .... (8)...

2.Express中的“路由器”

Router：路由器，是一个可以包含多个路由声明的对象，可以相关的路由（Route）放置到同一个路由器（Router）中

|  |
| --- |
| //news.js  let router = require('express').Router();  router.get(**'/list'**, function(req, res){}) //news/list  router.get(**'/detail'**, function(req, res){}) //news/detail  module.exports = router; |
| //app.js  let routerNews = require('./news');  app.use('**/news'**, routerNews); |

3.Node/Express项目中使用MySQL数据库

必须下载第三方的mysql模块：

npm i mysql

const mysql = require('mysql');

let pool = mysql.createPool({

host: '127.0.0.1',

user: 'root',

password: "",

database: "mfresh",

connectionLimit: 10

});

pool.query("SQL.....", function(err, result, cols){

})

课后任务：

完成“mfresh-api项目”所有的数据API。包括如下四个模块：

(1)新闻模块

(2)用户模块

(3)产品模块

(4)购物车模块