**1，中间件**

用来拦截对路由的请求，为路由服务的。在中间件中既可以获取到**请求**，也可以做出**响应**。

中间件分为应用级中间件、路由级中间件、内置中间件、第三方中间件、错误处理中间件

**（1）应用级中间件**

也称为自定义中间件

app.use( url, (req,res,next)=>{ } )

url: 表示要拦截的URL ，一旦拦截到自动执行回调函数；在回调函数中可以获取到请求以及做出响应，next是一个函数，next表示要执行下一个中间件或者路由。

练习：创建购物车的路由（get /shopping）,添加中间件拦截对购物车的请求，在中间件中对传递的商品价格price打8折；最后在路由中响应‘商品的价格为：xxx’

|  |
| --- |
| Error: Can't set headers after they are sent.  已经响应结束了（send,sendFile,redirect..），不允许再次响应 |

**(2)路由级中间件**

就是路由器的使用

**(3)内置中间件**

只有一个，托管静态资源

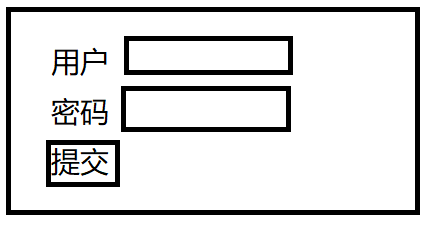
静态资源(html，css，js，图像...)

如果浏览器端要请求服务器端的文件，不需要使用路由响应文件，而是让浏览器端自动到某个目录下寻找。

app.use( express.static('目录名称') )

练习：再次托管静态资源到files目录，如果public和files出现相同文件名称显示哪一个。

练习：编写文件03\_three.js，创建web服务器，设置端口；托管静态资源到public目录，在该目录下创建文件login.html；点击提交，向服务器发请求(post /login)，创建对应的路由，响应‘登录成功’；获取到post传递的数据



**(4)第三方中间件**

也属于第三方模块，需要先下载安装并引入

|  |
| --- |
| //1.引入parser中间件  const bodyParser=require('body-parser');  //2.应用body-parser中间件，将post请求的数据解析为对象  app.use( bodyParser.urlencoded({  extended:false //不使用扩展的qs模块，而是使用核心模块querystring  }) );  //3.在路由中获取post传递数据，格式为对象  req.body |

**2.mysql模块**

是Node.js下专门用于操作mysql数据库的一个第三方模块

mysql数据库使用步骤

启动mysql

code.tarena.com.cn

tarenacode

code\_2013

执行mysql命令

脚本模式 mysql -uroot < 拖拽过来 回车

交互模式

mysql -uroot 回车 进入到交互模式

mysql.exe -h127.0.0.1 -P3306 -uroot -p

show databases; 显示所有的数据库

use 数据库名; 进入指定的数据库

show tables; 显示当前数据库所有的表

desc 表名称; 描述表中的列

select \* from 表名称; 查询数据

insert into 表名称 values( ... ); 插入数据

update 表名称 set 列名称=值,... where 条件; 修改数据

delete from 表名称 where 条件; 删除数据

//插入对象形式的数据

//SQL语句不属于标准的，mysql模块中才可以使用

**insert into emp set ?**

**(1)普通连接**

|  |
| --- |
| const c=mysql.createConnection(); //创建连接对象  c.connect(); //建立连接 （可省略）  //执行SQL语句，会自动建立连接  c.query(sql, callback)  sql 要执行的SQL语句  callback 使用回调函数获取结果  err 可能产生的错误  result SQL语句的执行结果 |

**(2)使用连接池**

|  |
| --- |
| const pool=mysql.createPool( {...} );//创建连接池对象  //执行SQL语句，和之前普通连接执行方式一样  pool.query(sql, callback) |

数据库的常用攻击方式：SQL注入

SQL注入：在用户提供的数据中含有SQL语法。

查询是否登录成功

select \* from user where uname='**range'** and upwd=**'123456' or 1=1;**

select \* from emp where eid='1' or 1=1;

or1=1——或者任何数据都满足1

**使用占位符(?)，可以对数据进行过滤，避免数据库攻击**

课后任务

(1)复习今天内容，整理思维导图

(2)创建web服务器，托管静态资源到public目录，在该目录下创建add.html，点击提交按钮，向服务器发送请求(get /add)，创建对应路由获取表单传递的数据，把这个数据插入到mysql数据库下的tedu下的dept表中，响应‘部门添加成功’

