# day05

## 1.中间件

（1）应用级中间件

（2）路由级中间件

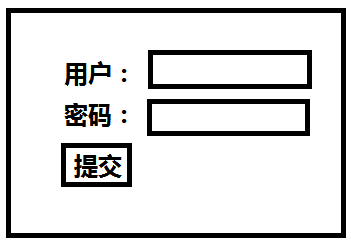
路由器的使用 app.use(‘use’,用户路由器)

（3）内置中间件

app.use( express.static(‘托管的目录’) );

把所有的静态资源托管到某个目录，如果浏览器请求文件，会自动到该目录下查找，不需要再创建单独路由响应文件。

练习：托管静态资源到public下，向服务器端请求login.html,点击提交，再次想服务器发请求post /losin 在服务器端获取提交的数据



（4）第三方中间件

body-parser中间件的使用

用于将post请求的数据格式化为对象

|  |
| --- |
| //1.下载安装并引入  const bodyParser=require(‘body-parser’);  //2.使用中间件  app.use(bodyParser.urlencoded({  extended:false //不使用第三方的qs模块格式化为对象，而是使用核心模块querystring  }));  //3.在post请求的路由中  req.body 返回一个对象 |

（5）错误中间件

## 2.mysql模块

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

INSER INTO emp VALUES(…)

DELETE FROM emp WHERE eid=5;

UPDATE emp SET sex=0,salary=8000 WHERE eid=6;

SELECT \* FROM emp;

|  |
| --- |
| var connection=mysql.createConnection({ })创建连接对象，在参数中传递连接mysql服务器的相关信息  connection.connect() 建立连接  connection.query(aql,callback) 执行SQL语句，参数sql就是要执行的SQL语句，callback回调函数用于获取SQL语句的结果  connection.end() 关闭连接 |

查询数据，返回数组

插入/修改/删除数据，返回对象，通过affectedRows可以判断是否执行成功，如果大于0成功。

SQL注入

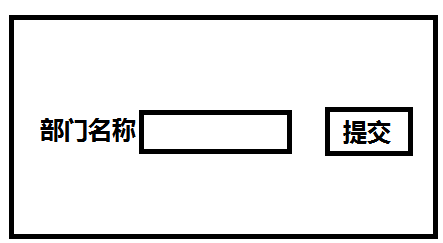
SELECT \* FROM user WHERE uname=’root’ and upwd=123456 **OR 1=1**;

使用连接池

|  |
| --- |
| var pool=mysql.createPool({ }) 创建连接池，使用connectionLimit设置数量，默认是15个  pool.query(sql,[],callback) 执行SQL语句，sql执行的SQL语句，第2个参数用于替换SQL语句中的值，callback回调函数，获取执行的结果 |

练习：使用连接池中的连接向员工表插入数据

练习：创建web服务器，托管资源到public目录下，创建文件add.html,点击提交，向服务器发请求get,/add,服务器端获取该数据



课后任务：

（1）复习，整理思维导图

（2）练习：创建web服务器，托管静态资源，创建emp\_add.html,点击提交,将数据添加到tedu下的emp表中。点击提交请求 get /add

