

1. 创建登录页面 —— page

当使用浏览器访问登录页面时，从客户端发送请求到服务端

服务器接收到请求之后，“生产”出一个登录需要使用的页面，返回给客户端（浏览器）

我们在浏览器上看到登录页面

2. 点击登录完成登录操作 —— page 跳转到 login

在上一次请求返回的登录页面中的输入框内，输入账号密码，然后点击登录按钮

此时会发出第二次请求，从客户端发送到服务端，同时会带去输入的内容（账号和密码）

服务器接收到请求之后，将收到的账号密码，拿去找数据库进行匹配，看是否存在

3. 在点击登录之后进性用户登录的效验 —— login

核心功能——比较账号密码

找数据库进行比较

如果用户存在，则“跳转”到登录成功页面，否则继续输入账号密码

4. 项目实现中需要注意的细节：

mysql驱动包需要存放在 Test\_Login\WebContent\WEB-INF\lib 路径下，并且需要添加到 build path 中

mysql版本对应相关的驱动 jar 包

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","root123");

中需要使用正确的

地址 —— localhost

端口号 —— 3306

数据库名称 —— test

数据库账号 —— root

数据库密码 —— root123

数据库操作对象中的SQL 语句必须没有SQL语法错误

查询结果分离时，必须先创建User类型的对象

checkLoginDao() 需要在方法内实现异常处理，否则会对整个项目造成通篇修改的“灾难”

注意类内的局部变量和全局变量的使用，主要是LoginDao类中

SQL 建表和插入数据

CREATE TABLE t\_user (

user\_id int(11) NOT NULL,

user\_name varchar(30) NOT NULL,

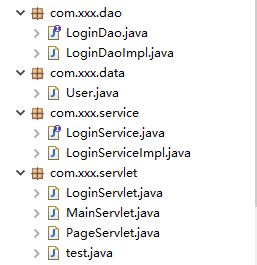
user\_password varchar(30) NOT NULL,

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

INSERT INTO t\_user VALUES(1,'qwer','123');

INSERT INTO t\_user VALUES(2,'小明','222');

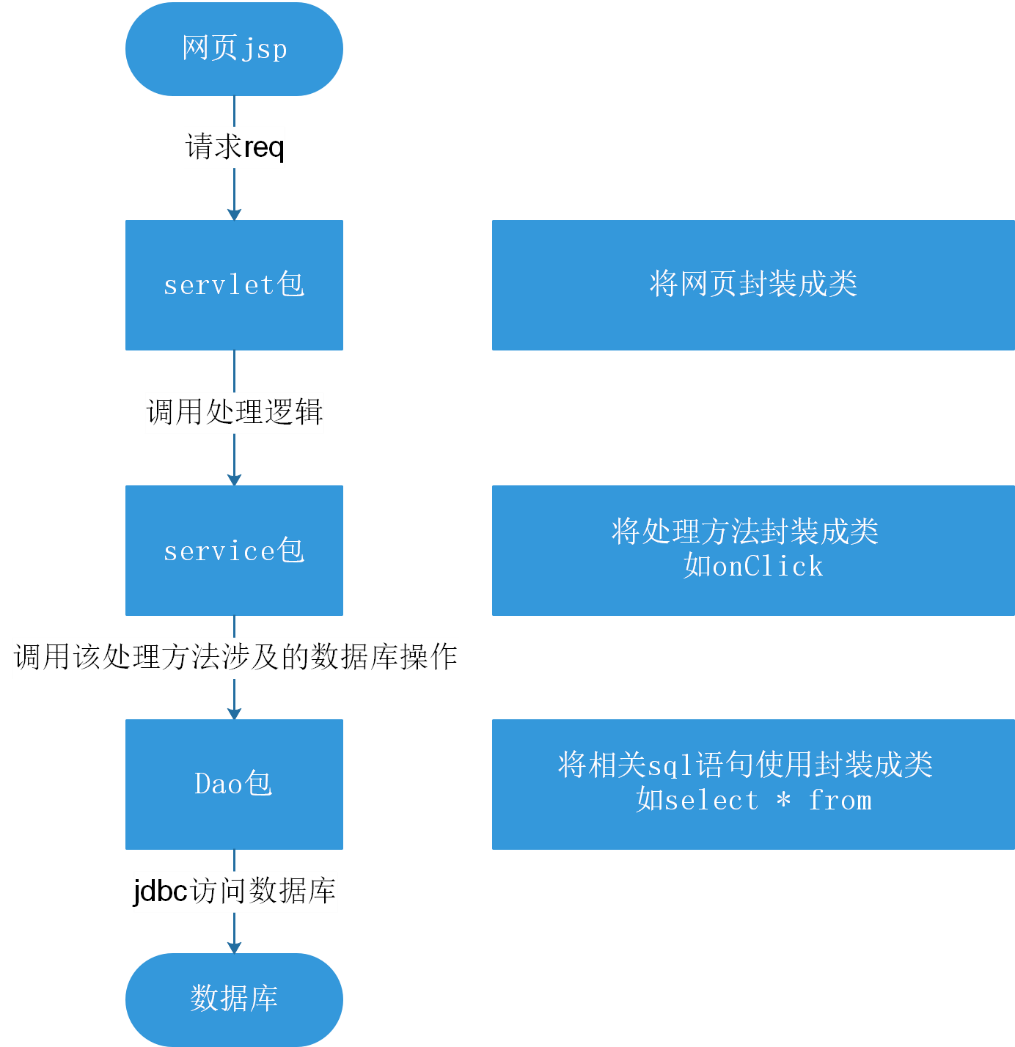


Dao: 在后台负责进行数据库的访问 （数据库操作对象）

Data：存放实体

Servlet： 网页显示与交互跳转，负责请求与响应，常调用service业务层处理方法

Service： 业务逻辑处理，常调用Dao数据操作层操作数据库



HTML负责发送请求数据和接收响应数据  
  
Servlet负责接受请求数据和回写响应数据，进行一些简单的逻辑判断确定回写的数据  
  
Service负责核心逻辑的编写及对数据的处理和封装  
  
Dao负责对数据库数据的查询和修改