# 实训五Servlet的使用

**一、实训目的**

1．掌握Servlet的创建与配置；

2．熟悉过滤器的应用范围；

3．掌握创建和配置过滤器的基本步骤

**二、实训条件**

1．需配备安装有Windows7操作系统、IE浏览器的计算机，每人一台台。

**三、实训要求及注意事项**

1． Servlet是一种服务器端的Java Web技术，可以用来生成动态的Web页面,其对Web请求的处理能力非常强大。Servlet实质上就是按Servlet规范编写的Java类,与Java普通程序不同的是,Servlet的运行需要Servlet容器的支持。通常所说的Servlet就是指HttpServlet，在开发Servlet时，可以直接继承javax.servlet.http.HttpServlet。用于处理Http请求的方法有doGet()、doPost()、service()等。

2．Servlet是javax.servlet包中HttpServlet类的子类，由服务器完成该子类的创建和初始化。Servlet的生命周期定义了一个Servlet如何被加载、初始化，以及它怎样接收请求、响应请求、提供服务。

3.在标准的MVC模式中，Servlet仅作为控制器（Control）使用。Java EE应用框架也正是遵循了MVC模式，对其而言，JSP仅作为表现层（View）技术。Model通常由JavaBean来充当，所有业务逻辑、数据访问逻辑都在Model中实现，实际上隐藏在Model层中还有很多丰富的组件，例如DAO组件、领域对象等。MVC的工作模式简要概述为：所有用户请求都发送给Servlet, Servlet调用Model来处理用户请求，并调用JSP来呈现处理结果；或者Servlet直接调用JSP将应用的状态数据呈现给用户。

**四、实训内容**

1． 熟悉MVC模式，利用Servlet作为控制器实现一个简单的登录验证。

2． 本项目共需要3个程序文件：

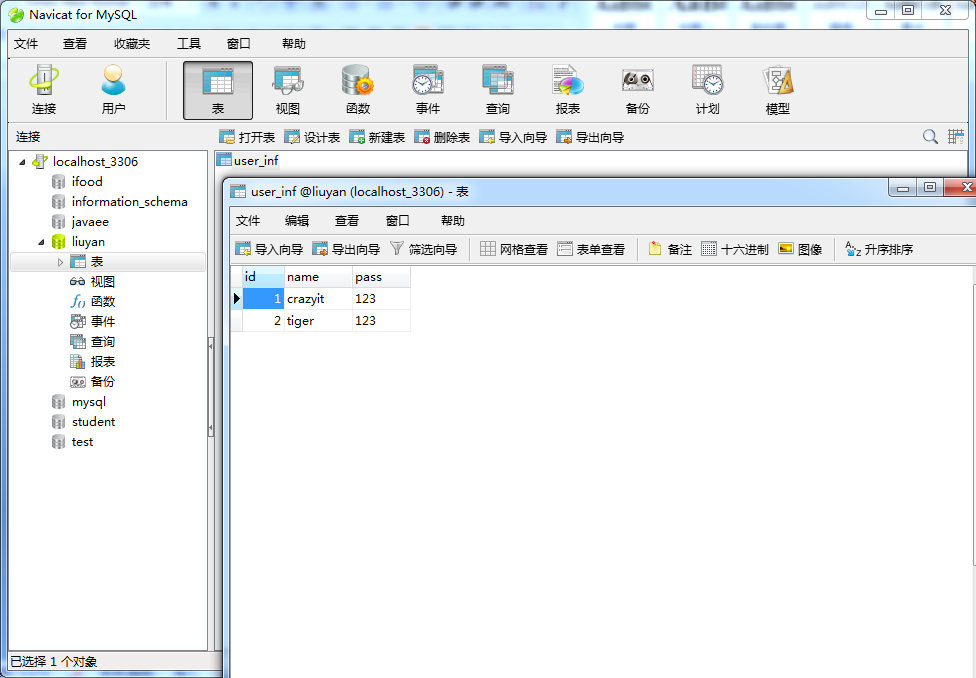
1）视图层：登录页面：Login.jsp

2）控制器：LoginServlet.java

3）模型层：DAO组件：DbDao.java

**五、实训步骤**

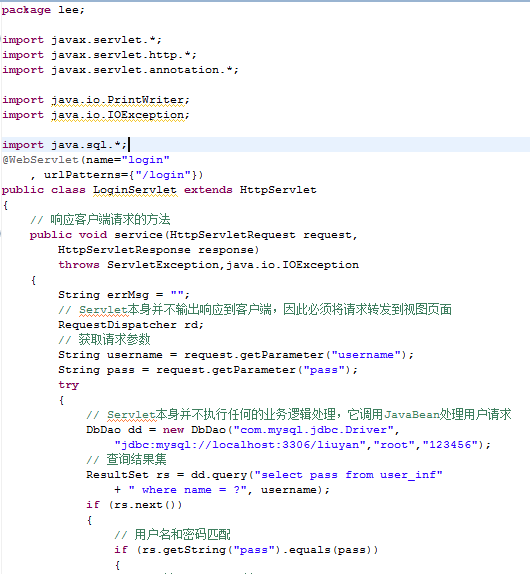
1． 创建MYSQL数据库名为“liuyan”，并创建表“user\_inf”，如下图所示，将记录加入表中。



2、设计login.jsp页面



2、设计Servlet





3、设计DAO

**package** lee;

**import** java.sql.\*;

**public** **class** DbDao

{

**private** Connection conn;

**private** String driver;

**private** String url;

**private** String username;

**private** String pass;

**public** DbDao()

{

}

**public** DbDao(String driver , String url

, String username , String pass)

{

**this**.driver = driver;

**this**.url = url;

**this**.username = username;

**this**.pass = pass;

}

// 下面是各个成员属性的setter和getter方法

**public** **void** setDriver(String driver) {

**this**.driver = driver;

}

**public** **void** setUrl(String url) {

**this**.url = url;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** **void** setPass(String pass) {

**this**.pass = pass;

}

**public** String getDriver() {

**return** (**this**.driver);

}

**public** String getUrl() {

**return** (**this**.url);

}

**public** String getUsername() {

**return** (**this**.username);

}

**public** String getPass() {

**return** (**this**.pass);

}

// 获取数据库连接

**public** Connection getConnection() **throws** Exception

{

**if** (conn == **null**)

{

Class.*forName*(**this**.driver);

conn = DriverManager.*getConnection*(url,username,

**this**. pass);

}

**return** conn;

}

// 插入记录

**public** **boolean** insert(String sql , Object... args)

**throws** Exception

{

PreparedStatement pstmt = getConnection().prepareStatement(sql);

**for** (**int** i = 0; i < args.length ; i++ )

{

pstmt.setObject( i + 1 , args[i]);

}

**if** (pstmt.executeUpdate() != 1)

{

**return** **false**;

}

pstmt.close();

**return** **true**;

}

// 执行查询

**public** ResultSet query(String sql , Object... args)

**throws** Exception

{

PreparedStatement pstmt = getConnection().prepareStatement(sql);

**for** (**int** i = 0; i < args.length ; i++ )

{

pstmt.setObject( i + 1 , args[i]);

}

**return** pstmt.executeQuery();

}

// 执行修改

**public** **void** modify(String sql , Object... args)

**throws** Exception

{

PreparedStatement pstmt = getConnection().prepareStatement(sql);

**for** (**int** i = 0; i < args.length ; i++ )

{

pstmt.setObject( i + 1 , args[i]);

}

pstmt.executeUpdate();

pstmt.close();

}

// 关闭数据库连接的方法

**public** **void** closeConn()

**throws** Exception

{

**if** (conn != **null** && !conn.isClosed())

{

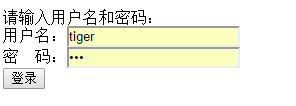
conn.close();

}

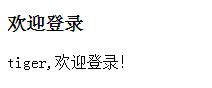
}}

**六、数据记录及处理**

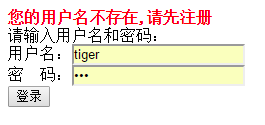
1． 登录页面



2．成功登录效果



3．账号密码出错的情况



**六、考核方式**

实训报告+考勤