



分层思想.doc
16.89KB

一、JavaWeb开发模式

C/S: 客户端 / 服务器 （胖客户端）

B/S: 浏览器 / 服务器 （瘦客户端）

JavaBean:

就是一个普通类（实体bean），包含三样标准：一个无参构造、私有属性、公共的getter和setter方法

1、Model1模式

JSP + JavaBean



2、Model2模式

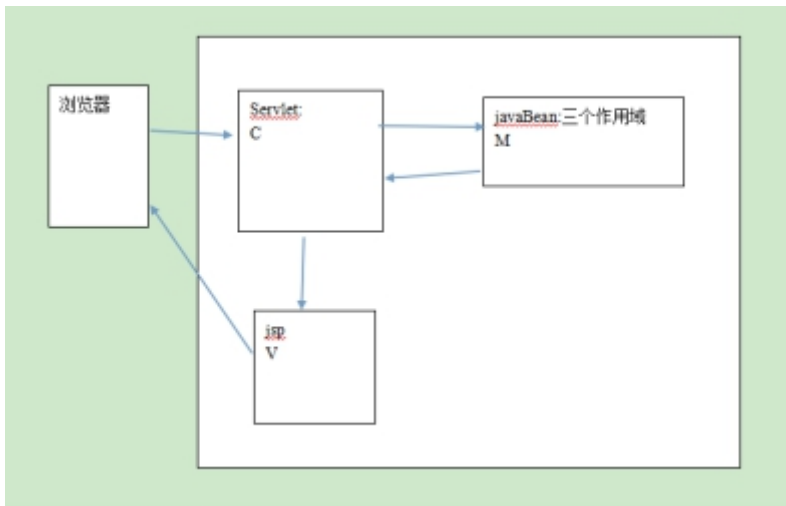
JSP + Servlet + JavaBean

MVC:开发模式

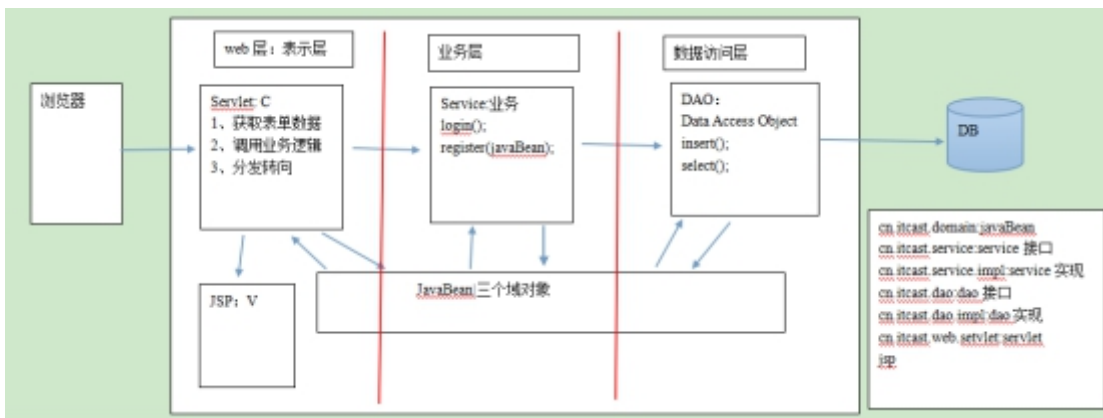
M: Model模型 JavaBean|四种作用域

V: view视图 JSP

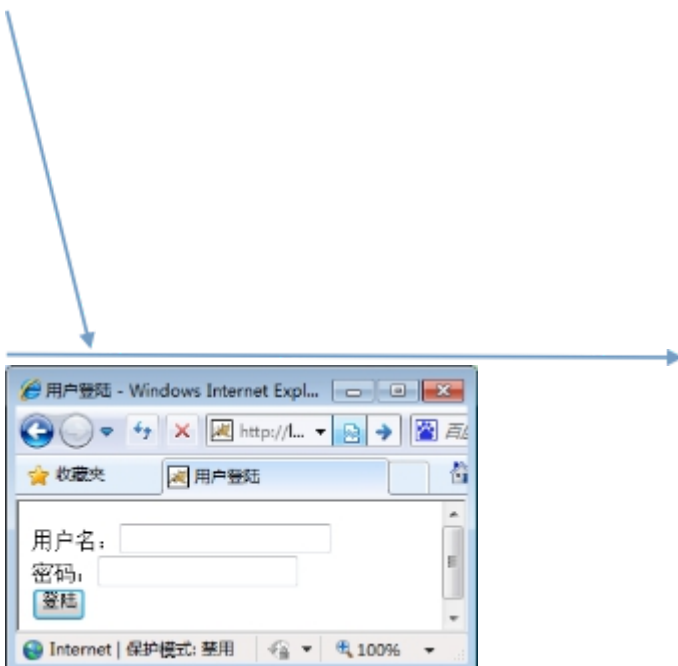
C:Controller控制器 Servlet

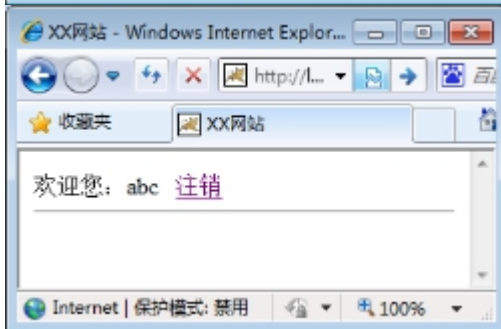
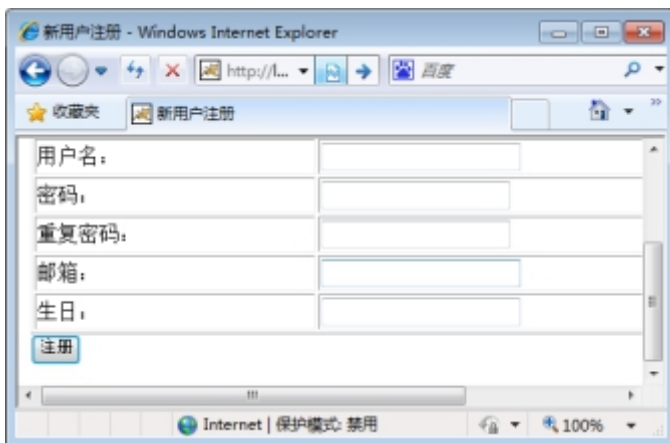
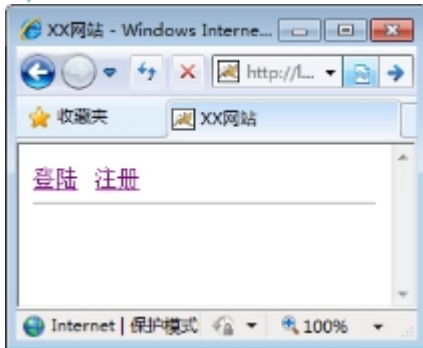


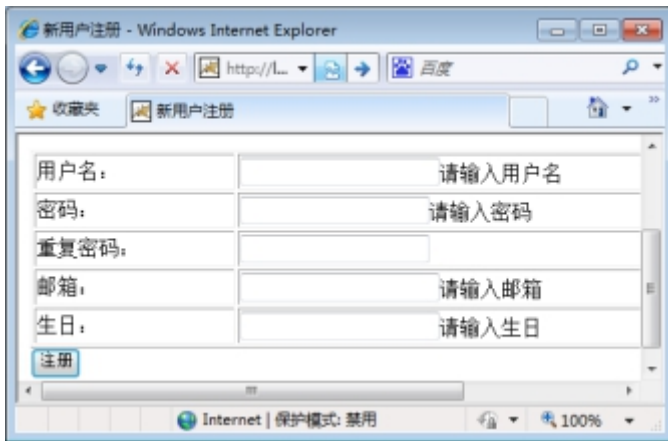
分层思想：强内聚、弱耦合



二、案例：用户的注册和登陆







三、开发步骤:

开发时注意事项:

约定优于编码。列名、字段名、表单中的属性名要一至

1、创建数据库及表

```
DROP DATABASE day12;

CREATE DATABASE day12;

USE day12;

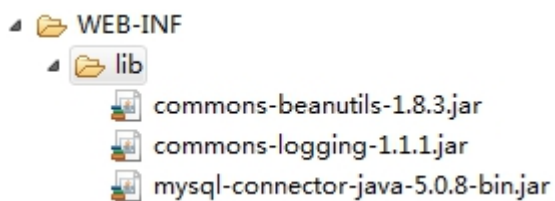
CREATE TABLE users(
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    PASSWORD VARCHAR(50),
    email VARCHAR(200),
    birthday DATE
)

SELECT * FROM users;

INSERT INTO users(username,PASSWORD,email,birthday) VALUES ('tom','123','tom@163.com','2015-08-28');
```

2、开发web应用

a、搭建开发环境：添加jar包





b、创建实现类（javaBean）

```
import java.io.Serializable;
import java.util.Date;
//实体bean。用于封装数据
public class User implements Serializable{
    private int id;
    private String username;
    private String password;
    private String email;
    private Date birthday;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
```

c、service层（业务层）

接口

```
public interface UserService {
    /**
     * 根据用户名和密码实现登录
     * @param username
     * @param password
     * @return 如果没找到返回null
     */
    public User login(User user);

    /**
     * 注册用户信息
     * @param user
     */
    public void register(User user);
}
```

实现类

```

public class UserServiceImpl implements UserService {
    private UserDao dao = new UserDaoImpl();
    public User login(User user) {
        return dao.findUser(user);
    }

    public void register(User user) {
        dao.insert(user);
    }
}

```

d、DAO（数据访问层）

接口

```

public interface UserDao {
    /**
     * 查找用户
     * @param user
     * @return
     */
    public User findUser(User user);

    /**
     * 添加用户
     */
    public void insert(User user);
}

```

实现类

添加工具类：DBUtils

cn.itcast.util
DBUtils.java

info.properties

```

public User findUser(User user) {
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    User u = null;
    if(user==null){
        System.out.println("你的用户数据呢？");
    }
    try {
        //创建连接
        conn = DBUtils.getConnection();
        //得到执行sql语句的对象PreparedStatement
        String sql = "select username,password,email,birthday,id from";
        ps = conn.prepareStatement(sql);
        //给sql赋值
    }
}

```

```

public void insert(User user) {
    Connection conn = null;
    PreparedStatement ps = null;
    try {
        //创建连接
        conn = DBUtils.getConnection();
        //得到执行sql语句的对象PreparedStatement
        String sql = "INSERT INTO users(username,PASSWO
        ps = conn.prepareStatement(sql);
        //给?赋值
        ps.setString(1, user.getUsername());
        ps.setString(2, user.getPassword());
        ps.setString(3, user.getEmail());
        SimpleDateFormat sdf = new SimpleDateFormat("yy
        ps.setString(4, sdf.format(user.getBirthday()))
        //执行sql语句,并返回结果集
        ps.executeUpdate();
    }
}

```

e、表示层（JSP、Servlet）

要注意：

注册页面--》regServlet

登录页面--》loginServlet