1. 事物

**什么是事务？？？**

数据库事务(database transaction) 是指作为**单个逻辑单元内的一系列操作**，要么全部成功，要么全部失败！

**不管是事务回滚还是手动提交，都需要将事务的自动提交关闭**

**connection.setAutoCommit(false);**

**如果事务操作中没有产生任何异常，就执行提交操作，影响数据库中的数据**

connection.commit();

**如果事务操作中产生了异常，则执行事务回滚，回滚到最初状态**

connection.rollback();

**事务的四大特性（ACID四个特性）**

1. 原子性（Atomicity）  
   原子性是指事务包含的所有操作要么全部成功，要么全部失败回滚，因此事务的操作如果成功就必须要完全应用到数据库，如果操作失败则不能对数据库有任何影响。
2. 一致性（Consistency）  
   一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。
3. 隔离性（Isolation）  
   隔离性是当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。
4. 持久性（Durability）  
   持久性是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，即便是在数据库系统遇到故障的情况下也不会丢失提交事务的操作。

**事务的四大隔离级别（由低 - -高 排序）**

**Read uncommitted 脏读的问题：**

读未提交，顾名思义，就是一个事务可以读取另一个未提交事务的数据

**Read committed 不可重复读：**

读提交，顾名思义，就是一个事务要等另一个事务提交后才能读取数据

**Repeatable read 幻读：**

重复读，就是在开始读取数据（事务开启）时，不再允许修改操作

**Serializable 序列化:**

**Serializable 是最高的事务隔离级别，在该级别下，事务串行化顺序执行，可以避免脏读、不可重复读与幻读。但是这种事务隔离级别效率低下，比较耗数据库性能，一般不使用。**

软件架构的分类：bs cs

**bs模型，browser server 浏览器服务器模型**

**cs模型，client server 客户端服务器模型**

服务端的认知

1. **什么是服务端？？？**

**服务端是为客户端服务的，服务的内容包括为客户端提供需要的数据、保存客户端的数据。**

1. **客户端与服务端的交互过程，就是不断的请求、响应的过程！而且请求与响应式一组概念，有请求必有响应！**
2. **具备接收请求、响应数据功能的称为服务端或者服务器！**
3. **java中的服务端的作用是什么呢？？？**

**本质上还是用来处理请求、响应数据的。**

**java中的客户端 browser java中的服务端 就是java中的后台服务器tomcat**

**页面与后台java代码的中转站！让我们可以通过java代码来处理页面发送的请求以及对页面做出数据响应！**

1. **java中使用的后台服务器是tomcat，用户群体极其广泛！！！**

tomcat的下载、安装、启动、暂停，tomcat集成到eclipse中

动态web工程的创建

servlet

什么是servlet？

**我们后台具备接收请求、响应数据功能就是servlet！！！**

服务程序或服务连接器，用Java编写的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8/100571)端程序，具有独立于平台和[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE/13020269)的特性，主要功能在于**交互式地浏览和生成数据，生成动态**[Web](https://baike.baidu.com/item/Web/150564)**内容**。

创建servlet的步骤1 2 3 4

1. 、在页面中书写表单，发送请求
2. 、创建servlet
3. (3) 、在web.xml文件中，完成对servlet的配置以及路径关联！
4. (4) 、启动tomcat，测试代码是否能够正常运行，请求是否能够顺利的发送到servlet！

## servlet的访问流程

**action---url-patter---servlet-name---- servlet-name---- servlet-class—servlet类—service重写**

在web.xml文件中对servlet进行注册绑定

<servlet>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>com.offcn.servlet.LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

servlet servlet-name servlet-class

servlet-mapping servlet-name url-pattern

servlet的访问流程

servlet的工作原理

登录案例：

1. 获取参数 request.getParameter(“请求参数的关键字”)

关键字=值&关键字=值（name=”uname”）

1. 中文乱码的处理

get在tomcat8.0之后自动解决了中文乱码

post需要设置request.setCharacterEncoding(“utf-8”)

1. 响应数据

设置响应的类型及编码 response.setContentType(“text/html;charset=utf-8”)

获取响应对象内置的打印流 response.getWriter()

通过打印流对象向页面响应内容writer.write(内容)