**周报**

1. 处理Web开发中遇到的部分安全性问题：  
    在开发中，经常遇到一个网页需要在另一个网页之后打开，即直接进入第二个界面是不被允许的，这就需要在开发的时候，对需要处理的网页进行拦截器处理；说到拦截器不得不说的就是关于页面跳转的若干种方法，主要可分为两种：第一种是重定向，第二种是转发；两者是有很重要的区别的：

重定向过程：客户浏览器发送http请求——》web服务器接受后发送302状态码响应及对应新的location给客户浏览器——》客户浏览器发现是302响应，则自动再发送一个新的http请求，请求url是新的location地址——》服务器根据此请求寻找资源并发送给客户。在这里location可以重定向到任意URL，既然是浏览器重新发出了请求，则就没有什么request传递的概念了。在客户浏览器路径栏显示的是其重定向的路径，客户可以观察到地址的变化的。重定向行为是浏览器做了至少两次的访问请求的。

转发过程：客户浏览器发送http请求——》web服务器接受此请求——》调用内部的一个方法在容器内部完成请求处理和转发动作——》将目标资源发送给客户；在这里，转发的路径必须是同一个web容器下的url，其不能转向到其他的web路径上去，中间传递的是自己的容器内的request。在客户浏览器路径栏显示的仍然是其第一次访问的路径，也就是说客户是感觉不到服务器做了转发的。转发行为是浏览器只做了一次访问请求。

所以，结合本网页的性质，你打算根据转发来进行网页之间的跳转，具体的实现正在实现之中。

1. 开发网页端的批量导入功能：  
    在单个的导入时，服务器可根据用户传输的数据流对图片信息及文档信息进行编解码，队中存储在指定的本地服务器硬盘中，但是在批量导入时，会牵扯到线程的调用问题。因为在待传输的文件数量较多时，我们一般采用多线程进行传输，但是在传输过程中又会遇到对于资源的同步问题，这里，在搜索了众多资料之后，发现JDK1.5之后的开发包中新增了Concurrent包，该包在处理多线程任务时很是出色，你打算学习使用，以下给出该开发包中的若干重要的类：
   1. 等待多个线程完成执行的CountDownLatch类：某个主线接到一个任务，起了n个子线程区完成，但是主线程需要等待这n个子线程都完成了才开始执行某个操作。
   2. 线程安全并且无阻塞的Atomic类：Atomic相关的类比如：AtomicLong，AtomicInteger类，可以分别对Long，Integer类型的数据进行数据的同步，这些类都是线程安全的，支持无阻塞无锁定的set(),get(),getAndSet(),getAndIncrement(),getAndDecrement(),getAndAdd()等操作。
   3. ReadWriteLock：读锁可以有很多个锁同时上锁，只要当前没有写锁；写锁是排他的，上了写锁，其他的线程既不能上读锁，也不能上写锁；同样，需要上写锁的前提是既没有读锁，也没有写锁。
   4. Condition接口：其基本方法是await（）和signal（）；Condition接口依赖于Lock接口，生成一个Condition的基本代码是lock.newCondition（）；调用Condition的await（）方法和signal（）方法都必须在lock的保护之内，就是说必须在lock.lock（）和lock.unlock（）之间才可以；和Object.wait（）方法一样，每次调用Condition的await（）方法，当前线程就自动释放了对当前锁的拥有权。