☆将eclips web工程导入myeclips工程的方法





☆删除仓库阻止下载文件的命令

**for /r %i in (\*.lastUpdated) do del %i**

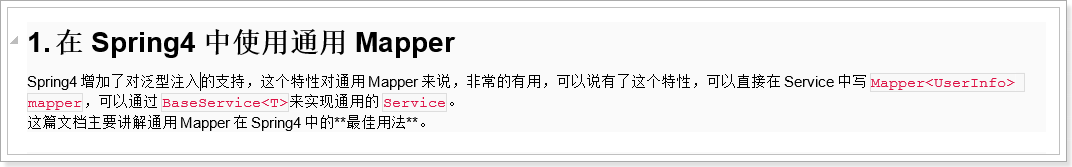
☆JPA常用注解

[**http://blog.csdn.net/wanghuan203/article/details/8698102**](http://blog.csdn.net/wanghuan203/article/details/8698102)

☆实体类实现Serializable的意义

实体类通过实现 java.io.Serializable 接口以启用其序列化功能。未实现此接口的类将无法使其任何状态序列化或反序列化。**序列化接口Serializable没有方法或字段，仅用于标识可序列化的语义**  
    **实现了Serializable接口的对象，可将它们转换成一系列字节，并可在以后完全恢复回原来的样子。**这一过程亦可通过网络进行。这意味着序列化机制能自动补偿操作系统间的差异。换句话说，可以先在Windows机器上创建一个对象，对其序列化，然后通过网络发给一台Unix机器，然后在那里准确无误地重新“装配”。不必关心数据在不同机器上如何表示，也不必关心字节的顺序或者其他任何细节。  
    **serialization主要用来支持2种主要的特性**：  
    1、RMI(Remote method invocation)。RMI允许象在本机上一样操作远程机器上的对象。当发送消息给远程对象和调用远程方法时，就需要用到serializaiton机制来发送参数和接收返回值。   
    2、保存信息。在某个时候把状态信息保存起来，以便以后某个时候能恢复这些状态信息。  
    Hibernaet和EJB中的实体Bean就用到了上面两个特性。  
    另外：保存的时候不仅能保存对象的副本，而且还会把对象里面所引用的对象也保存起来，以此类推。就像在编译某个类一样，会涉及到所用到的所有类。但是所引用的对象也必须是可序列化的，不然会抛NotSerializableException异常。

☆Spring4的新特性



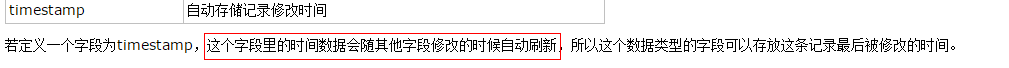
☆COOKIE



☆Spring MVC 中如果以.html结尾，那么久不会返回JSON数据

SpringMVC的规定：**在SpringMVC中如果请求以html结尾，那么就不会返回JSON数据**。

☆MYSQL可变时间格式



☆关系型数据库和非关系型数据库

[**http://blog.csdn.net/robinjwong/article/details/18502195**](http://blog.csdn.net/robinjwong/article/details/18502195)

☆JS严格模式

[**http://www.ruanyifeng.com/blog/2013/01/javascript\_strict\_mode.html**](http://www.ruanyifeng.com/blog/2013/01/javascript_strict_mode.html)

☆webservice框架cxf和Axis对比

[**http://www.cnblogs.com/snake-hand/archive/2013/06/09/3129915.html**](http://www.cnblogs.com/snake-hand/archive/2013/06/09/3129915.html)

☆Druid介绍

**www.iteye.com/magazines/90 qq群92748305**

☆Linux命令大全

**http://www.cnblogs.com/fnlingnzb-learner/p/5831284.html**

如何查看端口占用情况

        这里主要是用到windows下的DOS工具,点击"开始"--"运行",输入"cmd"后点击确定按钮,进入DOS窗口,接下来分别运行以下命令:  
        >netstat -aon | findstr "8080"  
                Proto  Local Address          Foreign Address        State            PID  
                ====  ============      ==============  ==========  ======  
                TCP    0.0.0.0:80                    0.0.0.0:0                LISTENING      1688  
可以看出80端口被进程号为1688的程序占用.  
        >tasklist | findstr "1688"  
图像名                                                PID            会话名                    会话#       内存使用  
               ========================= ====== ================ ======== ============  
               inetinfo.exe                                        1688           Console                      0              2,800 K  
很明显,是inetinfo占用了80端口;inetinfo.exe主要用于支持微软Windows IIS网络服务的除错,这个程序对你系统的正常运行是非常重要的.  
        当然,并不是只有inetinfo.exe进程会占用80端口,这只是我机器上的情况.如果你并不了解此进程是干什么用的,千万不要盲目地将其kill掉,最好先百度或Google搜索一下;当然如果你很了解它,并确定可以终止,那么继续下面的命令.  
        >taskkill /pid 1688 /F  
成功: 已终止 PID 为 1688 的进程。  
如果你很熟悉此进程,并确定可以终止,那么就直接使用上面的命令把PID为1688的进程终止.(这一步同样可以在任务管理器中执行,inetinfo.exe就是任务管理器中的映像名称,选中它,点击"结束进程"即可)  
        >tasklist | findstr "1688"  
再次确认是否成功终止,如果成功终止此次执行命令后应返回空.

**2、linux下如何查看80端口占用情况?是被哪个进程占用?如何终止等**

查询端口是否被占用，被哪个进程占用有两种方式：1、netstat -anl | grep "80" ；2、lsof -i:80

终止进程的方式：kill pid

☆注解元数据@retention@target

**E:\PersonalDoc\离线网页\java 注解 @Retention @interface 元数据 .html**

☆注解元数据@retention@target

**E:\PersonalDoc\离线网页/SprignMVC+myBatis整合(—— 基于MapperFactoryBean .html 工作流文档.html**

☆工作流文档

**E:\PersonalDoc\离线网页/工作流文档.html**

☆阿里Maven地址

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

☆Mysql 服务安装和开启

mysqld –install

net start mysql

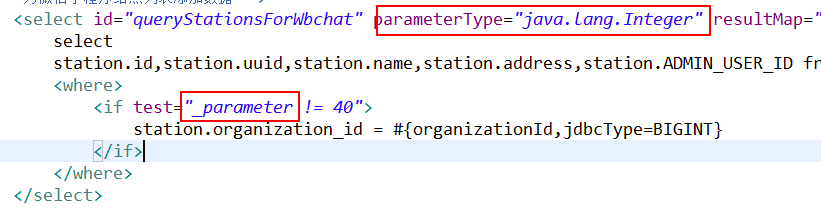
☆Mybatis restultType 和 resultMap

**·**两者都不能自动映射表字段到实体类属性；

**·**resultType : 通过as 后面接实体类属性；

**·**resultMap : 通过 <id> 和 <column> 中的 property。

☆Mybatis 单个参数的if语句注意事项



**注意：必须使用\_parameter,而不是参数名。**

☆LINUX 运行和停止.jar项目

前段时间刚刚接触了一下linux系统，本人表示第一次使用无从下手，不像windows操作系统是可视化的，简单的创建文件或文件夹都要通过写命令才能将其完成，由于工作需要，不得不自学linux ，先是自学常用的一些命令，然后再部署可执行jar，以下是具体的操作步骤。

1、执行jar包的命令和在windows操作系统上是一样的，都是java -jar xxxx.jar。  
2、将jar程序设置成后台运行，并且将标准输出的日志重定向至文件consoleMsg.log。

nohup java -jar getCimiss-surf.jar >consoleMsg.log 2>&1 &

其中：nohup命令的作用就是让程序在后台运行，不用担心关闭连接进程断掉的问题了，consoleMsg.log文件前提要创建好。  
3、如果想杀掉运行中的jar程序，查看进程命令为：

ps aux|grep getCimiss-surf.jar

将会看到此jar的进程信息

data 5796 0.0 0.0 112656 996 pts/1 S+ 09:11 0:00 grep --color=auto getCimiss-surf.jar

data 30768 6.3 0.4 35468508 576800 ? Sl 09:09 0:08 java -jar getCimiss-surf.jar

其中30768则为此jar的pid，杀掉命令为

kill -9 30768

☆SHELL 脚本检测进程是否存在，不存在则重启

如何保证服务一直运行？如何保证即使服务挂掉了也能自动重启？在写服务程序时经常会碰到这样的问题。在Linux系统中，强大的shell就可以很灵活的处理这样的事务。   
下面的shell通过一个while-do循环，用ps -ef|grep 检查loader进程是否正在运行，如果没有运行，则启动，这样就保证了崩溃挂掉的进程重新被及时启动。   
必须注意两点：   
1、ps |grep 一个进程时必须加上其路劲，否则容易grep到错误的结果；   
2、必须用 -v 从结果中去除grep命令自身，否则结果非空。

复制代码代码如下:

#!/bin/sh   
#=====================   
#YuanHui.HE   
#khler@163.com   
#=====================   
while :   
do   
echo "Current DIR is " $PWD   
stillRunning=$(ps -ef |grep "$PWD/loader" |grep -v "grep")   
if [ "$stillRunning" ] ; then   
echo "TWS service was already started by another way"   
echo "Kill it and then startup by this shell, other wise this shell will loop out this message annoyingly"   
kill -9 $pidof $PWD/loader   
else   
echo "TWS service was not started"   
echo "Starting service ..."   
$PWD/loader   
echo "TWS service was exited!"   
fi   
sleep 10   
done

如果启动此shell时发现进程已经存在，说明以别的方式启动了进程而不是此shell，那么它会持续提醒找到进程，解决办法是，要么只用此shell启动服务，要么一经发现以其他方式启动的服务即kill掉，上面的语句就是这么干的：   
kill -9 $pidof $PWD/loader