[1.开启和关闭对外开放网卡的命令 2](#_Toc12609)

[2.查看系统是否是Ubuntu还是CentOS 2](#_Toc21248)

[3.ubuntu上快速安装couchdb 2](#_Toc11682)

[安装couchdb出现错误,是由于删除旧的couchdb不彻底导致 2](#_Toc22733)

[4.怎样上次本地代码到github上 3](#_Toc30211)

[5.css遇到的问题（div1字数达到一定的宽度自动换行，以及div2始终是垂直居中于div1） 3](#_Toc4718)

[6.常见的json解析格式 4](#_Toc12406)

[7.angularjs能用map吗? 怎么用啊? 4](#_Toc20021)

[8.web.xml配置restful参考如下 5](#_Toc22630)

[9.拼接java json数据转义 6](#_Toc28345)

[10清除svn用户登陆的信息，直接删除auth下面所有的文件即可 6](#_Toc12158)

[11.open files的限制值 6](#_Toc15012)

[12.linux上安装JDK1.7的过程\*\* 8](#_Toc26308)

13.linux查找大文件

1. linux设置定时重启tomcat任务
2. Ajax的学习记录
3. Java后台web.xml加载的顺序：ServletContext -> listener -> filter -> servlet
4. 7.java中如何获得:后缀.properties的文件
5. mybatis入门文档

( <http://www.cnblogs.com/wushiqi54719880/archive/2011/07/26/2117601.html> )

## 19.[HttpClient在HTTP协议接口测试中的使用](http://www.cnblogs.com/zhangfei/p/5099036.html)

( http://www.cnblogs.com/zhangfei/p/5099036.html )

## 1.开启和关闭对外开放网卡的命令

ifup eth0

ifdown eth0

JSONArray array = JSONArray.fromObject(this.users.toArray());

## 2.查看系统是否是Ubuntu还是CentOS

lsb\_release -a

## 3.ubuntu上快速安装couchdb

http://www.xuebuyuan.com/890720.html

### 安装couchdb出现错误,是由于删除旧的couchdb不彻底导致

如下是错误：

No apport report written because the error message indicates its a followup error from a previous failure.

Errors were encountered while processing:

couchdb-server-5.5

couchdb-server

E: Sub-process /usr/bin/dpkg returned an error code (1)

彻底删除couchdb的命令

sudo apt-get autoremove couchdb\* --purge

重启couchdb

/etc/init.d/couchdb restart

解决删除couchdb参考文献

http://blog.csdn.net/suitingwei/article/details/50708334

重启mysql服务的命令

service mysqld restart

修改系统时间的命令

date -s "2015-06-15 13:30"

## 4.怎样上次本地代码到github上

http://www.jianshu.com/p/**08656eb84974**

## 5.css遇到的问题（div1字数达到一定的宽度自动换行，以及div2始终是垂直居中于div1）

<div style="margin-left: 5%;font-size: 16px;color:#666;overflow:auto;display:table;">

<span style="width:150px;

text-align:left;

white-space: normal !important;

display:table-cell;

vertical-align:middle;">呵呵呵呵呵额</span>

<span style="width:100px;margin-left:5%;

box-sizing: border-box;

-moz-box-sizing: border-box;

display:table-cell;

vertical-align:middle;">总数22</span>

</div>

## 6.常见的json解析格式

var obj = {

vos: [{

'term\_name': '山西面馆',

'quantum':'总数22',

bos: [{ 'product\_name': '50度超棉柔', 'num':'16' },

{ 'product\_name': '私人订制1', 'num':'10'}

]

},{

'term\_name': '羊蝎子京城第一家哈哈哈哈哈啊哈哈',

'quantum':'总数33',

bos: [{ 'product\_name': '50度超棉柔1', 'num':'16' },

{ 'product\_name': '私人订制3', 'num':'10'}

]

}]

}

## 7.angularjs能用map吗? 怎么用啊?

$scope.map = {key:"value"};

<h1 ng-repeat="(key,value) in map">{{value}}</h1>

## 8.web.xml配置restful参考如下

<!-- resful服务开始 -->

<filter>

<filter-name>CorsFilter</filter-name>

<filter-class>et.common.servlet.CorsFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CorsFilter</filter-name>

<url-pattern>\*</url-pattern>

</filter-mapping>

<context-param>

<param-name>resteasy.resources</param-name>

<!-- 需要修改注册resful服务 -->

<param-value>

</param-value>

</context-param>

<servlet>

<servlet-name>Resteasy</servlet-name>

<servlet-class>

org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher

</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Resteasy</servlet-name>

<url-pattern>/api/\*</url-pattern>

</servlet-mapping>

<!-- resful服务结束 -->

web.xml他的作用:

主要是配置 web项目启动时加载的信息，

比如<context-param/>配置参数，

<listener/>配置你的监听器，<filter/>配置过滤器，

<servlet/>配置你的servlet实现。

## 9.拼接java json数据转义

"[" +

"{\'infoType\':'公告',\'infotTitle\':\'今天晚上有雨\',\'iconType':\'1',\'detail1':\'记得带上雨伞\',\'detail2':\'哈哈哈哈\'},"+

"{\'infoType\':'打卡',\'infotTitle\':\'下班打卡\',\'iconType':\'2',\'detail1':\'别忘记了啊\',\'detail2':\'哈哈哈哈\'}]";

常用的json解析格式

JSONObject dataJson=new JSONObject("你的Json数据“);

JSONObject response=dataJson.getJSONObject("response");

JSONArray data=response.getJSONArray("data");

JSONObject info=data.getJSONObject(0);

String province=info.getString("province");

String city=info.getString("city");

String district=info.getString("district");

String address=info.getString("address");

System.out.println(province+city+district+address);

## 10清除svn用户登陆的信息，直接删除auth下面所有的文件即可

C:\Users\Administrator\AppData\Roaming\Subversion\auth

## 11.open files的限制值

对于一般的应用来说(象Apache、系统进程)1024完全足够使用。但是如何象squid、mysql、java等单进程处理大量请求的应用来说就有 点捉襟见肘了。如果单个进程打开的文件句柄数量超过了系统定义的值，就会提到“too many files open”的错误提示。如何知道当前进程打开了多少个文件句柄呢？下面一段小脚本可以帮你查看：

# lsof -n |awk '{print $2}'|sort|uniq -c |sort -nr|more

101 3556

94 4916

ps -ef|grep 3556

原来是一个tomcat占用的！

1.临时增大open files的限制值

可以用ulimit -n 来临时增大该限制值，

但是一旦重启服务器后，该值又会恢复到1024。

只能暂时性的解决问题。命令如下：

[root@redhat ~]# ulimit -n 65536

[root@redhat ~]

# ulimit -a

core file size

(blocks, -c) 0

data seg size

(kbytes, -d) unlimited

file size

(blocks, -f) unlimited

pending signals

(-i) 1024

max locked memory

(kbytes, -l) 32

max memory size

(kbytes, -m) unlimited

open files

(-n) 65536

pipe size

(512 bytes, -p) 8

POSIX message queues

(bytes, -q) 819200

stack size

(kbytes, -s) 10240

cpu time

(seconds, -t) unlimited

max user processes

(-u) 16384

virtual memory

(kbytes, -v) unlimited

file locks

(-x) unlimited

2.彻底增大open files的限制值

一般情况下，

我们都会通过修改系统限制文件来彻底增大open files的值。

2.1 通过VI打开/etc/security/limits.conf，

在“# End of file”上面添加如下：

\* - nofile 65536

该语句表示：每一个用户的默认打开文件数是65536

2.2 通过VI打开/etc/sysctl.conf,

在最后一行添加如下：

fs.file-max = 65536

lsof -p $java\_pid 每个文件描述符的具体属性

lsof -p $java\_pid | wc -l 当前Java进程file descriptor table中FD的总量

## 12.linux上安装JDK1.7的过程\*\*

文章来源http://www.cnblogs.com/cloudwind/archive/2012/10/19/2731263.html

Linux安装JDK1.7

Ubuntu版本：12.04 LTS

JDK版本：1.7.0\_04

安装目录：/usr/local/development/jdk1.7.0\_04

\*\*

安装JDK1.7的过程\*\*

1. 打开网址http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u4-downloads-1591156.html，下载相应的JDK文件。

2. 将下载的文件解压缩到/usr/local/development/jdk1.7.0\_04目录。

3. 创建文件/etc/profile.d/development.sh，将下面的内容复制到该文件。

export JAVA\_HOME=/usr/local/development/jdk1.7.0\_04

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=.:$JAVA\_HOME/lib:$JRE\_HOME/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$JRE\_HOME/bin:$PATH

4. 重启电脑，使用命令java -version检查JDK是否安装成功。

\*\*5.修改默认的java执行\*\*

查看当前是否存在java命令

update-alternatives --config java

update-alternatives --config javac

如果存在可以将原有的删除

update-alternatives --remove java /opt/tools/jdk1.6.0\_xx/bin/java

update-alternatives --remove javac /opt/tools/jdk1.6.0\_xx/bin/javac

或加入新的配置

update-alternatives --install /usr/bin/java java /opt/tools/jdk1.6.0\_45/bin/java 300

update-alternatives --install /usr/bin/javac javac /opt/tools/jdk1.6.0\_45/bin/javac 300

再次执行

update-alternatives --config java

update-alternatives --config javac

\*\*技巧一：使用development.sh设置环境变量\*\*

网络上很多关于设置JDK环境变量的方法，都是说修改/etc/environment，/etc/profile，~/,bash\_profile等文件。

其实如果打开/etc/profile，在最后我们可以发现有这样一段代码：

复制代码

复制代码

if [ -d /etc/profile.d ]; then

for i in /etc/profile.d/\*.sh; do

if [ -r $i ]; then

. $i

fi

done

unset i

fi

复制代码

复制代码

也就是说，在/etc/profile执行的最后，它会自动的执行/etc/profile.d目录下的所有可读的文件。这就是我们将设置JDK环境变量的工作放在/etc/profile.d/development.sh的原因，可以不用修改操作系统自带的/etc/profile内容，方便系统的移植。

\*\*技巧二：将development.sh放到/usr/local/profile.d\*\*

我的系统/usr/local挂载点是一个单独的磁盘分区，这样做的目的是方便系统的维护。/usr/local目录除了存放安装的软件以外，其实还可以将配置文件放在这个目录下面，例如上面的development.sh，这样重装系统的时候就不需要重做该文件。所以，可以新建目录/usr/local/profile.d，然后将development.sh复制到该新目录。至于怎么放到/etc/profile.d下面，我觉得可以采用符号链接的方式。

cd /ect/profile.d

sudo ln -s /usr/local/profile.d/development.sh

## 13.linux查找大文件

命令行如下

find . -type f -size +1000000k

Linux系统下查找大文件或目录的技巧

当硬盘空间不够时，我们就很关心哪些目录或文件比较大，看看能否干掉一些了，怎么才能知道呢？ 　　#已易读的格式显示指定目录或文件的大小，-s选项指定对于目录不详细显示每个子目录或文件的大小

　　du -sh [dirname|filename]

　　如：

　　当前目录的大小： 　　du -sh .

　　当前目录下个文件或目录的大小：

　　du -sh \*

[root@td-agtfl2 var]# du -sh \* 12K account 1.7M cache 16K crash 28K db 28K empty

　　显示前10个占用空间最大的文件或目录：

　　du -s \* | sort -nr | head

用root到/目录下去利用命令 du -s -h ./\* 查看哪个目录占用了这么多的磁盘，如果找到后，依次进入那个目录，利用相同的办法，就能定位到文件上去。

## 14.Linux定时重启tomcat操作如下

A.Linux控制台输入vi test.sh，如我创建的就是test.sh，可以自定义

test.sh文件的具体逻辑判断看下面代码,其中tomca-gne-0自己在linux系统中的

命名自己替换

#!/bin/sh

. /etc/profile

pid=`ps aux | grep tomcat-gne-0 | grep -v grep | grep -v retomcat | awk '{print $2}'`

echo $pid

if [ -n "$pid" ]

then

{

echo ===========shutdown================

/opt/tools/tomcat-gne-0/bin/shutdown.sh

sleep 1

pid=`ps aux | grep tomcat-gne-0 | grep -v grep | grep -v retomcat | awk '{print $2}'`

if [ -n "$pid" ]

then

{

sleep 1

echo ========kill tomcat==============

kill -9 $pid

}

fi

sleep 5

echo ===========startup.sh==============

/opt/tools/tomcat-gne-0/bin/startup.sh

}

else

echo ===========startup.sh==============

/opt/tools/tomcat-gne-0/bin/startup.sh

fi

B.我把建好的test.sh文件放在根目录下root下

C.文件test.sh建好保存没有问题后，手动执行无法操作  
需要给test.sh赋予权限命令如下：  
chmod a+x test.sh

D:接下来就要做个定时的任务来执行这个test.sh文件，用这个命令

crontab -e

输入这个命令后添加如下文件，设置启动的时间

 30 10 \* \* \* /root/test.sh

30代表分钟，10代表小时

这句话的意思就是每天的10点30启动这个test.sh文件

E:最后保存完定时文件后，启动定时文件  
命令如下：

service crond stop

service crond start

备注：准备工作都完好后，可以使用命令

crontab -l 查看定时任务进程

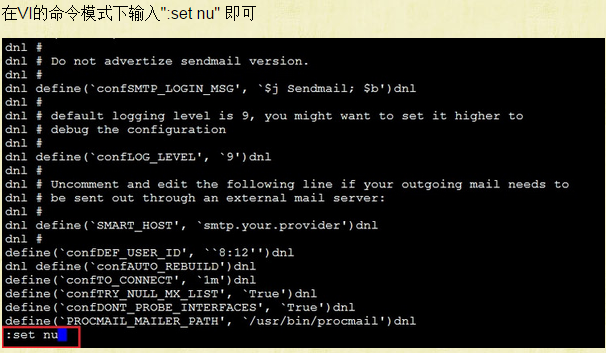
辅助命令：

查看某个文件显示行数命令如下：

cat -n test.sh

若要编辑某个文件用vi的话怎么显示编辑文件的行号呢？

如下图：



## Ajax简单叙述：

使用Ajax的步骤：

1. 创建XMLHttpRequest对象
2. Ajax向服务器发送请求(get,post方式请求)
3. 等待服务器的响应
4. Onreadystatechange响应处理事件

****XMLHttpRequest 是 AJAX 的基础****

variable=new XMLHttpRequest();

所有现代浏览器（IE7+、Firefox、Chrome、Safari 以及 Opera）均内建 XMLHttpRequest 对象。

### **老版本的 Internet Explorer （IE5 和 IE6）使用 ActiveX 对象：**

variable=new ActiveXObject("Microsoft.XMLHTTP");

****XMLHttpRequest 对象用于和服务器交换数据。****

如果需将请求发送到服务器，我们使用 XMLHttpRequest 对象的 open() 和 send() 方法：

## **GET 还是 POST？**

与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。

然而，在以下情况中，请使用 POST 请求：

* 无法使用缓存文件（更新服务器上的文件或数据库）
* 向服务器发送大量数据（POST 没有数据量限制）
* 发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

open() 方法的 *url* 参数是服务器上文件的地址：

xmlhttp.open("GET","ajax\_test.asp",true);

这里的第一个参数代表是get请求，第二个参数是请求的服务器地址，第三个参数true代表异步，false代表同步。

一般第三个参数还是设置为true比较好，除非是一些小的请求，也是可以使用false的

如当您使用 async=false 时，请不要编写 onreadystatechange 函数 - 把代码放到 send() 语句后面即可：

像这样：

xmlhttp.open("GET","test1.txt",false);

xmlhttp.send();

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

当请求被发送到服务器时，我们需要执行一些基于响应的任务。

每当 readyState 改变时，就会触发 onreadystatechange 事件。

readyState 属性存有 XMLHttpRequest 的状态信息。

下面是 XMLHttpRequest 对象的三个重要的属性：

示例代码：

xmlhttp.onreadystatechange=function()

{

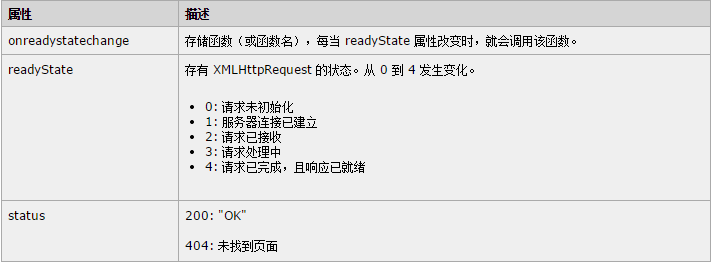
if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}



## **最后讲解：使用 Callback 函数**

callback 函数是一种以参数形式传递给另一个函数的函数。

如果您的网站上存在多个 AJAX 任务，那么您应该为创建 XMLHttpRequest 对象编写一个**标准**的函数，并为每个 AJAX 任务调用该函数。

该函数调用应该包含 URL 以及发生 onreadystatechange 事件时执行的任务（每次调用可能不尽相同）：

示例代码：

<script>

var xmlhttp;

function loadXMLDoc(url,cfunc)

{

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=cfunc;

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

function myFunction()

{

loadXMLDoc("/ajax/test1.txt",function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

});

}  
</script>

### 17.如果手动新建一个后缀为：properties的文件

### 如需获取.properties配置文件的属性用以下代码

**private** **static** ResourceBundle *visConfigResource* = ResourceBundle.*getBundle*("gne\_config");

全局配置参数

**public** **static** String get(String key) {

**return** *visConfigResource*.getString(key).trim();

}

**public** **static** **void** main(String[] args) {

System.*out*.println(GNEConfigRead.*get*("auth\_server\_url"));

}