**Java定时更新数据---定时器**

2012-10-31 15:14:23     [**我来说两句**](http://www.2cto.com/kf/201210/164940.html#comment_iframe)      作者：yanmei\_yao

在应用开发中，经常需要一些周期性的操作，比如每5分钟执行某一操作等。这次,在我们的开发中,就有这么一个需求.某个功能执行需要的时间比较长,则决定采用定时器的方式,每隔一段时间系统自动执行此功能,当界面真正触发时,只需要简单的读取结果,而不需要执行复杂的逻辑判断.那如何实现此功能呢?要想实现它,首先我们需要认识了解几个jdk封装的类.

[Java](http://www.2cto.com/kf/ware/Java/).util.Timer:一种线程设施,用于安排以后再后台线程中执行的任务.可安排任务执行一次或者定期重复执行.其中几个方法需要我们注意一下:

cancel():终止此计时器,丢弃所有当前已安排的任务.

schedule(TimerTask task, Date time) :安排在指定的时间执行指定的任务。

schedule(TimerTask task, Date firstTime, long period):安排指定的任务在指定的时间开始进行重复的固定延迟执行。

schedule(TimerTask task, long delay):安排在指定延迟后执行指定的任务。

schedule(TimerTask task, long delay, long period) :安排指定的任务从指定的延迟后开始进行重复的固定延迟执行。

scheduleAtFixedRate(TimerTask task, Date firstTime, long period) :安排指定的任务在指定的时间开始进行重复的固定速率执行。

scheduleAtFixedRate(TimerTask task, long delay, long period) :安排指定的任务在指定的延迟后开始进行重复的固定速率执行。

        在上面提到的Timer的几个方法中,参数中都涉及到了TimerTask类.那TimerTask类是干嘛的呢?他就是Timer所安排的任务.即被设置的要执行一次或多次的任务.这个类中下面这几个方法需要我们多加注意.

cancel():取消此计时器任务。

run():此计时器任务要执行的操作。这里是具体操作的代码实现.

        通过Timer和TimerTask两个类,我们就能实现任务的重复执行了.但是,我们并没有事情执行的起止点.我们不要人为的去触发事情的执行,我们要通过代码让服务自己去执行.这样,我们就需要一个监听,去监听当前是否符合执行的条件.ServletContextListener为我们实现此功能.此接口能够监听ServletContext对象的生命周期,实际上就是监听Web应用的生命周期.当Servlet容器启动或终止Web应用时,会触发ServletContextEvent事件,该事件由ServletContextListener来处理.涉及到的两个方法为:

contextInitialized(ServletContextEvent sce);初始化;

contextDestroyed(ServletContextEvent sce)销毁

       到这里,准备工作就做好了.下面我们来看一个例子定时器具体是怎样实现的.

[java]  
package com.boco.transnms.server.bo.stat;   
import java.util.Date;   
import java.util.Timer;   
import java.util.TimerTask;   
import javax.servlet.ServletContextEvent;   
import javax.servlet.ServletContextListener;   
public class NFDFlightDataTaskListener implements ServletContextListener{   
    @Override   
    public void contextDestroyed(ServletContextEvent arg0) {   
        //销毁时的代码    
    }   
    @Override   
        //在服务启动时，执行此方法。    
    public void contextInitialized(ServletContextEvent arg0) {   
        new TimerManager();   
    }   
}   
//要执行的任务    
class NFDFlightDataTimerTask extends TimerTask{   
    @Override   
    //此方法为具体要定时操作的方法    
    public void run() {   
     System.out.println("定时器测试:"+System.currentTimeMillis());   
    }   
}   
class TimerManager{   
    private static final long PERIOD\_DAY=6 \* 1000;  //每隔六秒执行一次    
    public TimerManager() {                      
        Timer timer = new Timer();     //定时器实例化     
         NFDFlightDataTimerTask task = new NFDFlightDataTimerTask();   //要执行的任务    
         //安排指定的任务在指定的时间开始进行重复的固定延迟执行。       
              timer.schedule(task,new Date(),PERIOD\_DAY);     
       }       
}

package com.boco.transnms.server.bo.stat;  
import java.util.Date;  
import java.util.Timer;  
import java.util.TimerTask;  
import javax.servlet.ServletContextEvent;  
import javax.servlet.ServletContextListener;  
public class NFDFlightDataTaskListener implements ServletContextListener{  
 @Override  
 public void contextDestroyed(ServletContextEvent arg0) {  
  //销毁时的代码  
 }  
 @Override  
        //在服务启动时，执行此方法。  
 public void contextInitialized(ServletContextEvent arg0) {  
     new TimerManager();  
 }  
}  
//要执行的任务  
class NFDFlightDataTimerTask extends TimerTask{  
 @Override  
 //此方法为具体要定时操作的方法  
 public void run() {  
  System.out.println("定时器测试:"+System.currentTimeMillis());  
 }  
}  
class TimerManager{  
 private static final long PERIOD\_DAY=6 \* 1000;  //每隔六秒执行一次  
 public TimerManager() {                    
     Timer timer = new Timer();     //定时器实例化  
      NFDFlightDataTimerTask task = new NFDFlightDataTimerTask();   //要执行的任务  
      //安排指定的任务在指定的时间开始进行重复的固定延迟执行。    
              timer.schedule(task,new Date(),PERIOD\_DAY);   
       }     
}

       既然监听是在服务启动的时候就有了，自然离不开配置文件了。那web.xml配置文件是如何配置的呢？

[[html](http://www.2cto.com/kf/qianduan/css/)]

view plaincopyprint?<listener>   
   
   <!—此处为实现监听接口的完整的包名类名-->   
  <listener-class>   
 com.boco.transnms.server.bo.stat.NFDFlightDataTaskListener   
 </listener-class>   
 </listener>

<listener>

<!—此处为实现监听接口的完整的包名类名-->

<listener-class>

  com.boco.transnms.server.bo.stat.NFDFlightDataTaskListener

</listener-class>

</listener>