tcc-fc-view: mail & schedule

tcc-fc-view repository: <http://tcci-svn/svn/src/tcc-fc/tcc-fc-view/trunk>

* **mail.readme:**

|  |
| --- |
| 模組功能:  ---------  \* 提供發送email功能, email內容為 Apache Velocity template 格式。  \* $vmformat.dateFormat: 自訂日期格式  \* $vmformat.escapeHtml: 換行(\n) -> <br/>, "<" -> &lt;, ">" -> &gt;  安裝步驟:  ---------  copy mail\\* to project\  建立email template: src\java\mail\_commonReport\_batch.vm (UTF-8 encoding)  sendmail sample code:  HashMap<String, Object> mailBean = new HashMap<String, Object>();  mailBean.put(VelocityMail.SUBJECT, "信件主旨");  // 收件者用逗號隔開  mailBean.put(VelocityMail.TO, "jimmy.lee@tcci.com.tw,gilbert.lin@tcci.com.tw");  mailBean.put("title", "異動通知"); // 在 vm 檔中用 $title取值  mailBean.put("reportItems", items); // 在 vm 檔中用 #foreach ($item in $reportItems)  mailBean.put("viewURL", viewURL);  mailBean.put("notifyURL", notifyURL);  VelocityMail.sendMail(mailBean, "mail\_commonReport\_batch.vm");  Note: 在email template中顯示日期, 多行(outlook不支援white-space:pre-line;)  $vmformat.dateFormat('yyyy-MM-dd HH:mm',$item.createtimestamp)  $vmformat.escapeHtml($item.note) |

* **schedule.readme:**

|  |
| --- |
| 模組功能:  ---------  \* 在load balance環境下AP的排程，同一時間僅一台AP會執行。  \* 可控制是否要執行排程(在資料庫設定)  安裝步驟:  ---------  copy schedule\\* to project\  DB setup: exec db-schema\tc\_schedule.sql  # 自行加上排程名稱於資料庫中  # ex: INSERT INTO TC\_SCHEDULE (NAME) VALUES ('CommonReportNotify');  修改在原始的排程facade, sample如下:  @EJB  private TcScheduleFacade scheduleFacade;  @Schedule(hour = "12,16")  public void notifyBatchSchedule() {  try {  // 30分內不要再執行  if (scheduleFacade.canExecute("CommonReportNotify", 30)) {  notifyBatch();  } else {  logger.warn("notifyBatchSchedule not execute");  }  } catch (Exception ex) {  logger.error("notifyBatchSchedule exception", ex);  }  }  停止(啟用)排程: 修改 TC\_SCHEDULE.ACTIVE欄位(0:停止, 1:啟用) |

scheduleFacade.canExecute:

|  |
| --- |
| // 排程能否執行  // true: name 要存在, 且 EXEC\_TIME 是null 或 與現在時間差大於minMinutesDiff  public boolean **canExecute**(String name, int minMinutesDiff) {  String sql = "SELECT EXEC\_TIME FROM TC\_SCHEDULE WHERE NAME=#name AND ACTIVE=1 FOR UPDATE NOWAIT";  Query q = em.createNativeQuery(sql);  q.setParameter("name", name);  Date now = new Date();  try {  Date execTime = (Date) q.getSingleResult();  boolean canExcute = (null==execTime) ? true :  (now.getTime() - execTime.getTime()) > minMinutesDiff\*60\*1000;  if (canExcute) {  sql = "UPDATE TC\_SCHEDULE SET EXEC\_TIME=#now WHERE NAME=#name";  q = em.createNativeQuery(sql);  q.setParameter("now", now);  q.setParameter("name", name);  q.executeUpdate();  logger.debug("{} schedule canExecute true", name);  return true;  } else {  logger.debug("{} schedule already executed at {}", name, execTime);  }  } catch (NoResultException ex) {  logger.warn("{} schedule not active or not exist", name);  } catch (Exception ex) {  logger.error("{} schedule canExecute exception", name, ex);  }  return false;  } |