COScheduler In Depth

Philippe Rabier - twitter.com/prabier
Sophiacom

Agenda

- Reminder: what is COScheduler?
- Job Validation
- Extend COScheduler: part 1
- Extend COScheduler: part 2
- Q&A

Use It!

Reminder
what is COScheduler?

Quartz Job Scheduler

- Very popular open source java library
- Exists for a long time (earlier than 2004)
- Can be used to create simple or complex schedules for executing tens, hundreds, or even tens-of-thousands of jobs
- Quartz 2.1.x is the current release

Quartz Main Components

- The JobDetails (JobDetail.java)
- The Triggers (Trigger.java)
- The Jobs (Job.java)
- The Scheduler (Scheduler.java)
- The Job Store (JobStore.java)

COScheduler Overview

COScheduler Goals

- Hide the (relative) complexity of Quartz
- Make simple job scheduling very simple to implement
- EOF integration
- Based on Project Wonder but with few dependancies (ERJars, ERExtensions and ERJavaMail)
- Based on the “Framework Principal” pattern (mandatory)
  http://wiki.objectstyle.org/confluence/display/WONDER/Creating+a+ERXFrameworkPrincipal+subclass

COScheduler Main Classes

- COJobDescription
  - Interface that your EOs must implement
- COSchedulerServiceFrameworkPrincipal
  - Abstract class that you have to subclass.
- COJob
  - Abstract class that you can use to implement your jobs

COScheduler Main Classes

- COJobSupervisor
  - Job running periodically which updates the list of jobs handled by Quartz
- COJobListener
  - Send informations when a job is done (emails, log, …)
  - Update the job description (firstExecutionDate, last and next)

Validation (what is it?)

Subclass COJob

- The validation methods are:
  - willDelete(final COJobDescription jobDescription)
  - willSave(final COJobDescription jobDescription)
  - validateForDelete(final COJobDescription jobDescription)
  - validateForSave(final COJobDescription jobDescription)

Example of Validation

Example of a job that executes remote procedures on MySql.

```
public void validateForSave(final COJobDescription jobDescription)
{
     // We get the list of remote procedures as a string
     String inputData = ((NOJobDescription)jobDescription).inputData();
     if (ERXStringUtilities.stringIsNullOrEmpty(inputData))
          throw new ValidationException("You have to enter at least one stored procedure.");

     else
     {
          NSArray<String> parameters = null;
          try
          {
               parameters = ERXValueUtilities.arrayValue(inputData);
          } catch (Exception e)
          {
               throw new ValidationException("The data you enter are incorrect.");
          }

          String sql = "SELECT ROUTINE_NAME FROM `information_schema`.`ROUTINES` where specific_name = ";
          EOEditingContext ec = ERXEC.newEditingContext();
          for (String aStoredProcedure : parameters)
          {
               NSArray<NSDictionary<String, String>> objects = EOUtilities.rawRowsForSQL(ec, "NOBusiness", sql + "'" + aStoredProcedure + "'", null);
               if (log.isDebugEnabled())
                    log.debug("method: validateForSave: aStoredProcedure: " + aStoredProcedure +" has been checked: " + (objects.count() > 0));
               if (objects.count() == 0)
                    throw new ValidationException("The stored procedure: " + aStoredProcedure + " doesn't exist.");
          }
     }
}
```

EO modification

- Modify the following methods
  - willDelete
  - willUpdate
  - willInsert
  - validateForDelete
  - validateForSave

Example of Validation

Example: modify the validateForSave method.

```
 public void validateForSave()
 {
      super.validateForSave();
      try
      {
           COUtilities.validateForSave(this);
      } catch (COJobInstanciationException e)
      {
           if (e.getErrorType() == ErrorType.CLASS_NOT_FOUND)
                throw new NSValidation.ValidationException("The class " + this.classPath() + " doesn't exist.");
           else
                throw new NSValidation.ValidationException("There is an error when trying to run " + this.classPath() + ".");
      }

 }
```

Code of COUtilities.validateForSave

```
 public static Job validateForSave(final COJobDescription jobDescription) throws COJobInstanciationException
 {
      Job aJob = createJobInstance(jobDescription);
      if (aJob instanceof COJob)
           ((COJob)aJob).validateForSave(jobDescription);
      return aJob;
 }
```

Extend COScheduler: part 1

Problem To Solve

Be warned when a job execution time is too long.

The solution

- Add an attribute maxDuration in our entity
- Create our own job listener by subclassing COJobListener
  - add a method isJobExecutionTooLong
  - override the methods:
    - getMailSubject
    - getMailContent
    - recipients (to force to send an email to the client)

Code of job listener

```
private boolean isJobExecutionTooLong(final JobExecutionContext jobexecutioncontext)
{
     if (jobexecutioncontext.getMergedJobDataMap().containsKey(JOB_TOO_LONG_KEY))
          return jobexecutioncontext.getMergedJobDataMap().getBoolean(JOB_TOO_LONG_KEY);

     boolean status = false;
     EOEditingContext ec = editingContext();
     ec.lock();
     try
     {
          NOJobDescription aJobDescription = (NOJobDescription) getJobDescription(jobexecutioncontext, editingContext());
          if (aJobDescription.maxDuration() != null)
          {
               // get the duration in minutes.
               long duration = jobexecutioncontext.getJobRunTime()/(1000*60);
               if (duration > aJobDescription.maxDuration())
                    status = true;
          }
     } catch (Exception e)
     {
          log.error("method: jobWasExecuted: exception when saving job description: ", e);
     } finally
     {
          ec.unlock();
     }
     jobexecutioncontext.getMergedJobDataMap().put(JOB_TOO_LONG_KEY, status);
     return status;
}
```

Code of job listener

```
public void jobToBeExecuted(final JobExecutionContext jobexecutioncontext)
{
     super.jobToBeExecuted(jobexecutioncontext);
     if (jobexecutioncontext.getMergedJobDataMap().containsKey(JOB_TOO_LONG_KEY))
          jobexecutioncontext.getMergedJobDataMap().remove(JOB_TOO_LONG_KEY);
}
```

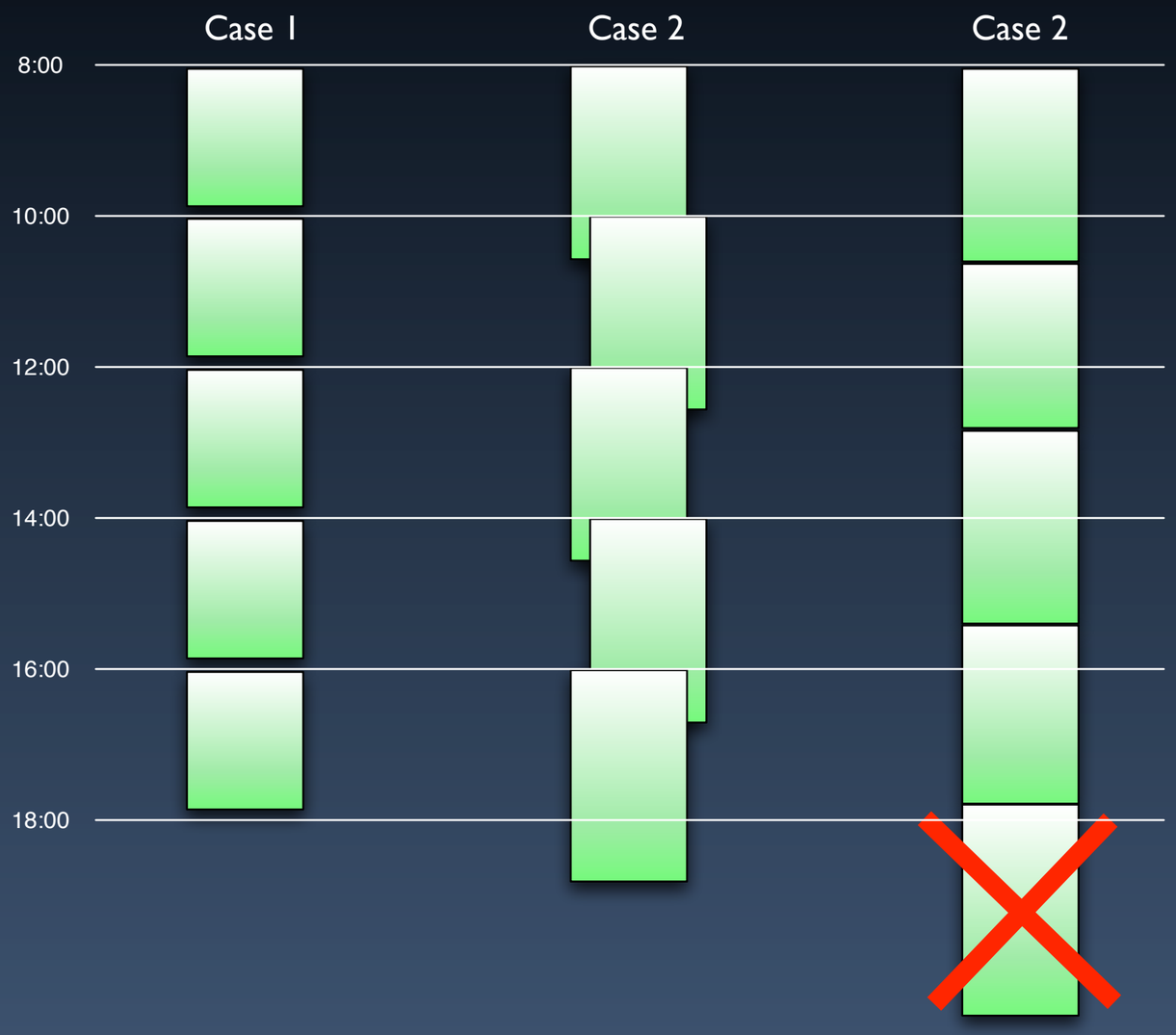
```
protected String getMailSubject(final JobExecutionContext jobexecutioncontext)
{
     if (isJobExecutionTooLong(jobexecutioncontext))
     {
          NOJobDescription aJobDescription = (NOJobDescription) getJobDescription(jobexecutioncontext, editingContext());
          return "ERROR: the expected max duration is: " + aJobDescription.maxDuration() + ". " + super.getMailSubject(jobexecutioncontext);
     }
     return super.getMailSubject(jobexecutioncontext);
}
```

Extend COScheduler: part II

Problem To Solve

Prevent a job to be executed after some time

Problem To Solve



The solution

- Add an attribute maxExecutionTime in our entity
- Create a trigger listener by subclassing COAbstractListener and implementing TriggerListener
  - only 2 methods need some works: getName and vetoJobExecution
    - getName (just return a name like this.getClass().getName()
    - vetoJobExecution (see code next slide)

Code of job listener

```
public boolean vetoJobExecution(final Trigger trigger, final JobExecutionContext context)
{
     EOEditingContext ec = editingContext();
     ec.lock();
     try
     {

          NOJobDescription jd = (NOJobDescription) getJobDescription(context, ec);
          if (jd != null && !ERXStringUtilities.stringIsNullOrEmpty(jd.maxExecutionTime()))
          {
               DateTime maxExecutionTime = NOJobDescription.hourAndSecondFormatter.parseDateTime(jd.maxExecutionTime());
               LocalTime maxExecutionLocalTime = maxExecutionTime.toLocalTime();
               LocalTime currentLocalTime = new DateTime().toLocalTime();
               boolean tooLate = currentLocalTime.compareTo(maxExecutionLocalTime) > 0;
               return tooLate;
          }
     } catch (Exception e)
     {
          log.error("method: vetoJobExecution: exception when getting the job description: ", e);
     } finally
     {
          ec.unlock();
     }
     return false;
}
```

Code of job listener

Modify your FrameworkPrincipal class and add the method finishInitialization()

```
@COMyJobListener("fr.sophiacom.ynp.schedulerservice.foundation.NSJobListener")
public class NSSchedulerServiceFrameworkPrincipal extends COSchedulerServiceFrameworkPrincipal
{
     static
     {
          log.debug("NSSchedulerServiceFrameworkPrincipal: static: ENTERED");
          setUpFrameworkPrincipalClass(NSSchedulerServiceFrameworkPrincipal.class);
          log.debug("NSSchedulerServiceFrameworkPrincipal: static: DONE");
     }

...

     public void finishInitialization()
     {
          super.finishInitialization();
          if (schedulerMustRun())
          {
               try
               {
                    // We add the trigger listener
                    TriggerListener tl = new NSTriggerListener(this);
                    getScheduler().getListenerManager().addTriggerListener(tl);
               } catch (SchedulerException e)
               {
                    log.error("method: finishInitialization: unable to add the trigger listener.", e);
               }
          }
     }
}
```

Q&A