MIND STORM SOFTWARE PVT LTD

Google App Engine Training

ExamResults – Step 6 – Cron Job

In this session, we are going to see how to write a Cron Job. A Cron Job is any code that you wish to execute repeatedly at a certain time.

Examples include:

- Fetch the latest news every 1 hour
- Update App statistics every 6 hours
- Check system to send any email reminders every 12 hours
- Take a backup every week on Wednesday at 12:00 AM

Add schedules to cron.xml

Cron Schedules are specified in a **cron.xml** file that is placed in the **/WEB-INF** directory. It can contain multiple entries and for each job, it will tell the schedule and what code to execute.

The code to be executed is specified as a URL that points to your Servlet that will get invoked automatically.

An example of the cron.xml file is shown below:

```
<cronentries>
<cron>
<url>/cron/checkstatus</url>
<description>Check Status every 60 minutes</description>
<schedule>every 60 minutes</schedule>
</cron>
</cronentries>
```

This file contains one cron entry. It is scheduled to run every 60 minutes. The scheduling is taken care of by App Engine. And when it is time for the Cron Job to fire, App Engine will simply invoke the end point that you have specified in the **<url>** element for your cron job entry above.

In our case, it will invoke the **/cron/checkstatus** url, so all we have to do next is to write a servlet that will contain our functionality and specify the mapping in the **web.xml** file with the above url.

Import the Project

We will import the Eclipse project at this stage so that you have all the working source code for this step.

1. In Eclipse, go to **File → Import**. This will bring up the Import dialog.
2. Select **General → Existing Projects into Workspace** and click on **Next**.
3. For Select root directory, click on browse and go to **C:\appengine-training\hands-on-exercises\Exam Results App\step 6**
4. Ensure that **Copy projects into workspace** is selected.
5. Click on **Finish**.

Add the Cron Job Servlet (ExamResultsEmailHandlerServlet.java)

We will write a new Servlet **CheckStatusCronJobSerlvet.java** that will only print a dummy message that it is executing. In reality, you will put in your logic over there but the idea here is to demonstrate the mechanism.

Take a look at the source code for the **CheckStatusCronJobSerlvet..java**. The entire source code is present in **hands-on-exercises/Exam Results App/step 6/src/com/mindstormsoftware/examresults/ CheckStatusCronJobSerlvet.java** file

You will find that it is a normal Java Servlet and in the doGet() method, it logs the fact that it is executed. In a real-life scenario, you will put your Cron Job functionality over here.

Add Servlet entry to web.xml

The following servlet entries are added to **web.xml**

```
<!--  Cron Jobs Handler -->
       <servlet>
             <servlet-name>CheckStatusCronJob</servlet-name>
             <servlet-
class>com.mindstormsoftware.examresults.CheckStatusCronJobServlet</servlet-class>
       </servlet>
       <servlet-mapping>
             <servlet-name>CheckStatusCronJob</servlet-name>
             <url-pattern>/cron/checkstatus</url-pattern>
      </servlet-mapping>
```

Instructions to Test locally

Once you made the changes, test out the changes locally first. Stop the Web Application if it is already running by clicking on the Stop icon in the Console window. Restart the application i.e. Right-click on Project and select **Run As → Web Application**.

The local Development Server does not ship with a Cron Scheduler, so the only way to test out your code is to execute it via the URL. i.e. **/cron/checkstatus** since that the same url that will be invoked when it is time to fire your job.

Sample output is shown below:

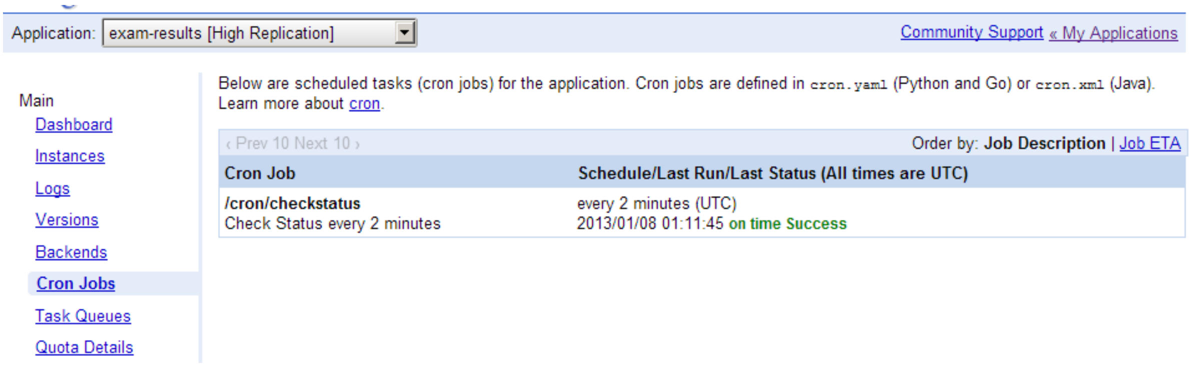
```
Jan 8, 2013 1:10:55 AM com.mindstormsoftware.examresults.CheckStatusCronJobServlet
doGet
INFO: Executing Cron Job ....
```

Instructions to Deploy and Test Live

To deploy to the cloud, right-click on project and then **Google → Deploy to App Engine.**

Visit the Admin console for your application and click on the **Main → Cron Jobs** link as shown below:

It should show you the Cron Jobs that are configured to execute including their last execution status.

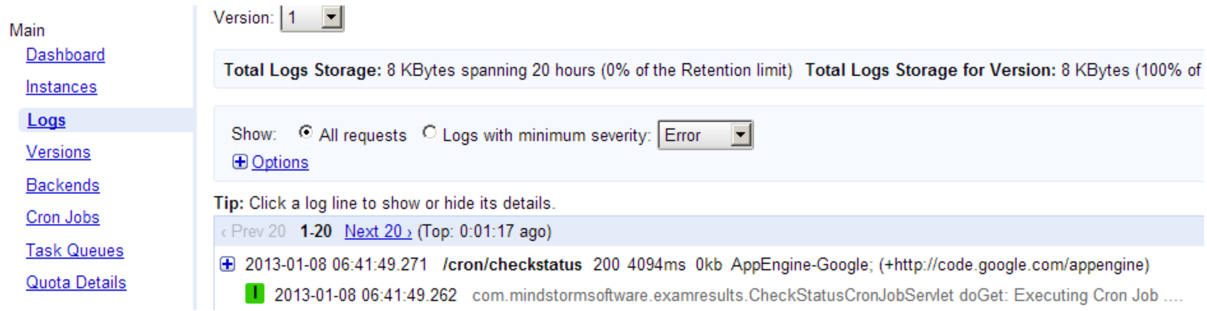

Application: exam-results [High Replication]
Community Support « My Applications
Main
Dashboard
Instances
Logs
Versions
Backends
Cron Jobs
Task Queues
Quota Details

Below are scheduled tasks (cron jobs) for the application. Cron jobs are defined in cron.yaml (Python and Go) or cron.xml (Java). Learn more about cron.

‹ Prev 10 Next 10 › Order by: **Job Description** | Job ETA

Cron Job	Schedule/Last Run/Last Status (All times are UTC)
/cron/checkstatus Check Status every 2 minutes	every 2 minutes (UTC) 2013/01/08 01:11:45 on time Success

You can view the Logs to see the execution messages.



Total Logs Storage: 8 KBytes spanning 20 hours (0% of the Retention limit) **Total Logs Storage for Version:** 8 KBytes (100% of

Show: ● All requests ○ Logs with minimum severity: Error

Tip: Click a log line to show or hide its details.

‹ Prev 20 **1-20** Next 20 › (Top: 0:01:17 ago)

```
2013-01-08 06:41:49.271 /cron/checkstatus 200 4094ms 0kb AppEngine-Google; (+http://code.google.com/appengine)
  2013-01-08 06:41:49.262 com.mindstormsoftware.examresults.CheckStatusCronJobServlet doGet: Executing Cron Job ....
```