Quartz Scheduler Details

Starting point for White Label Quartz scheduler configuration is Spring configuration file: application.xml

```xml
<!-- White Label Quartz scheduling START -->
 <bean name="jobWhiteLabelPoll"
class="org.springframework.scheduling.quartz.JobDetailFactoryBean">
   <property name="jobClass"
value="com.ob.scheduling.quartz.WhiteLabelPoll" />
   <property name="durability" value="true" />
 </bean>

 <bean id="triggerWhiteLabelPoll"
class="org.springframework.scheduling.quartz.SimpleTriggerFactoryBean">
   <property name="jobDetail" ref="jobWhiteLabelPoll" />
   <property name="repeatInterval"
value="${drupal.client.wl.poll_interval}" />
   <property name="startDelay" value="5000" />
 </bean>

 <bean id="whiteLabelScheduler"
class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
         <property name="autoStartup"
value="${drupal.client.wl.quartz_enabled}" />
         <property name="jobDetails">
   <list>
    <ref bean="jobWhiteLabelPoll" />
   </list>
  </property>
         <property name="triggers">
   <list>
    <ref bean="triggerWhiteLabelPoll" />
   </list>
  </property>
 </bean>
```

Bean `whiteLabelScheduler` is configured with `autoStartup` property, which allows to enable/disable scheduler if needed. The reason for this is that there is going to be Quartz schedulers on each platform, `DEV`, `TEST`, `STAGE` and `PROD`. We don't need to them all to be active. It is necessary to have only once instance, `PROD`. All other instances should be disabled once Content Management code is deployed to `PROD`. Value for `drupal.client.wl.quartz_enabled` comes from property file.

Bean `triggerWhiteLabelPoll` is configured with `repeatInterval` property which determines how often Quartz Job is triggered and Drupal is checked if CSS has to be recompiled. Property `drupal.client.wl.poll_interval` is configured in property file.

Bean `jobWhiteLabelPoll` is determining what action is taken when Quartz execution is triggered.

```
@Override
    public void executeInternal(final JobExecutionContext
jobExecutionContext) throws JobExecutionException {
        log.debug( "Calling WhiteLabelPoll" );
        ContentManagementService contentManagementService =
                ( ContentManagementService )context.getBean(
Constants.Service.CONTENT_MANAGEMENT);
        Map<String,String> rulesMap = new HashMap<>();
        rulesMap.put( TAG_NAME, CM_WHITE_LABEL_TAG_NAME );
        contentManagementService.readContentByTag( STAGE , rulesMap );
    }
```

Method `executeInternal` is called and it executes Java code which checks Drupal for White Label content which was been updated since last run and if necessary compiles it and publishes to Object Store.