CronJobs

It is like a timer.

It is a scheduled program.

If you have specific task which needs to be executed at periodic (like Every Sunday @ 8AM IST) → then put that task inside the **Cron job**.

Example: - I want to send promotion for new customer (who are within last 1 month) @ every **Sunday 8 AM IST**.

Example: - I want to email **daily @ 7 AM IST** about **Top 10 order details** (In terms of the price) to product manager.

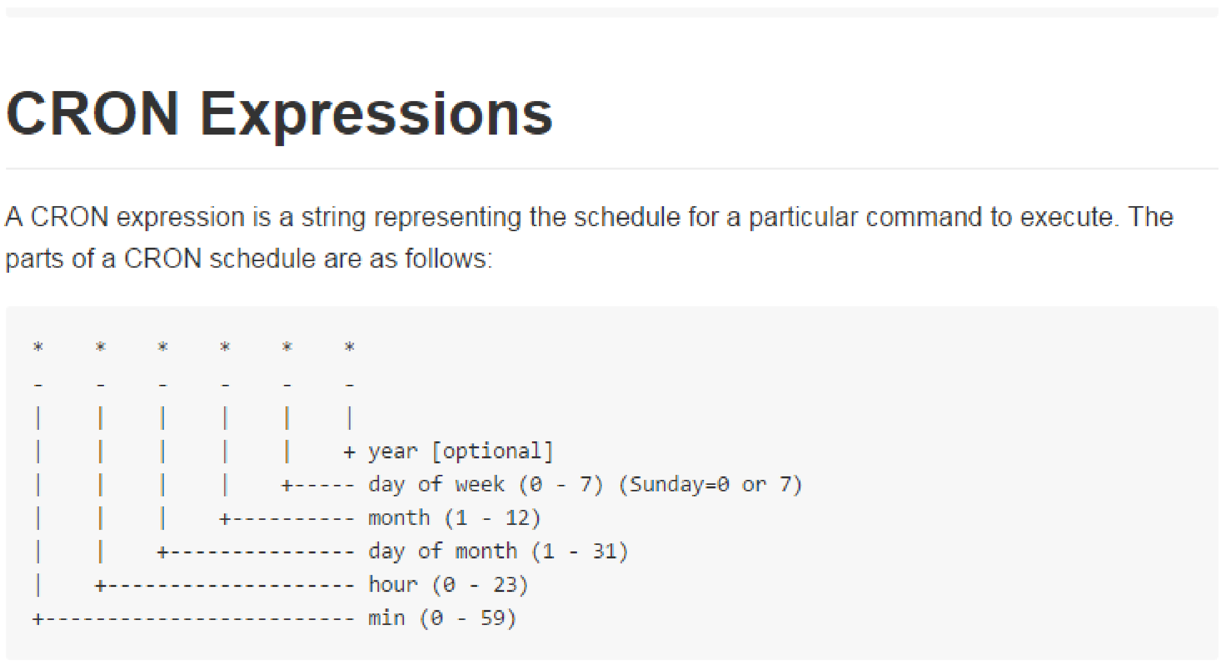
Example: - Load new products into “SAP Comm” every day @ 7 AM IST.

Q: What kind of logic we can write in Cronjob program?

> If we have something which needs to be executed at periodic intervals (or) at scheduled time – Then put that something inside the CronJob.

Q: What Cronjob contains?

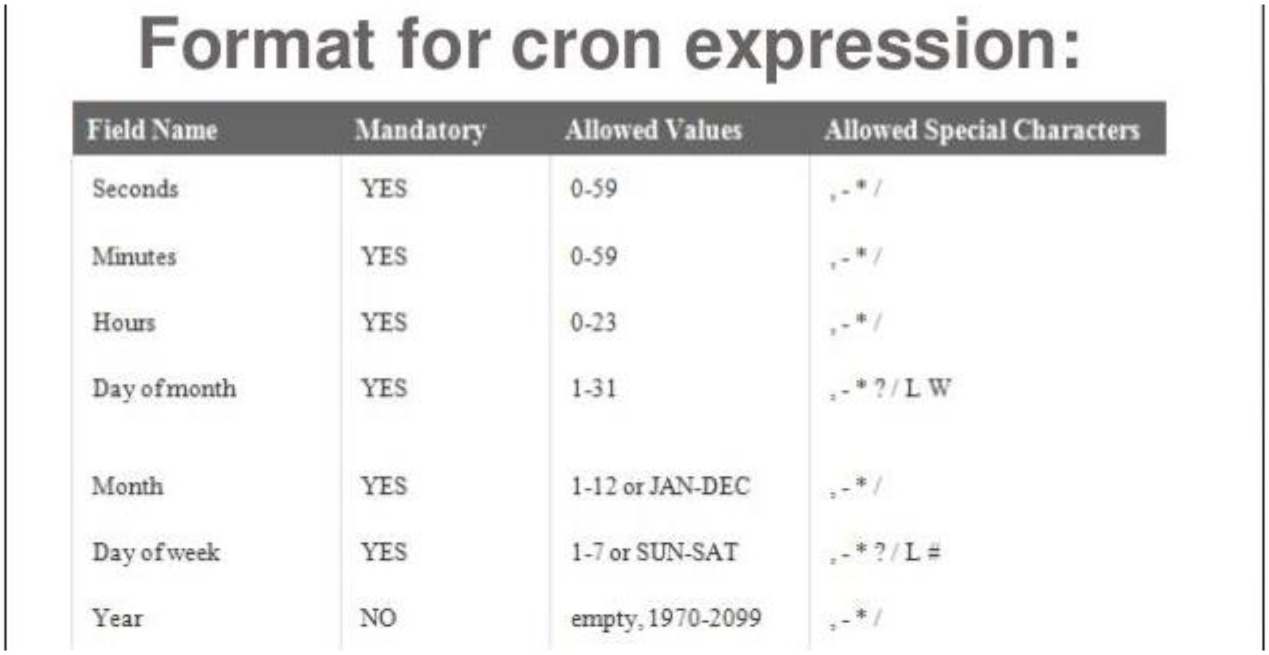
> **1) Trigger =** This is used to schedule when to run the job.
>
> For this – We will use “**Cron Expression**”.



CRON Expressions

A CRON expression is a string representing the schedule for a particular command to execute. The parts of a CRON schedule are as follows:

```
*    *    *    *    *    *
-    -    -    -    -    -
|    |    |    |    |    |
|    |    |    |    |    + year [optional]
|    |    |    |    +----- day of week (0 - 7) (Sunday=0 or 7)
|    |    |    +---------- month (1 - 12)
|    |    +--------------- day of month (1 - 31)
|    +-------------------- hour (0 - 23)
+------------------------- min (0 - 59)
```



Format for cron expression:

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	YES	0-59	, - * /
Minutes	YES	0-59	, - * /
Hours	YES	0-23	, - * /
Day of month	YES	1-31	, - * ? / L W
Month	YES	1-12 or JAN-DEC	, - * /
Day of week	YES	1-7 or SUN-SAT	, - * ? / L #
Year	NO	empty, 1970-2099	, - * /

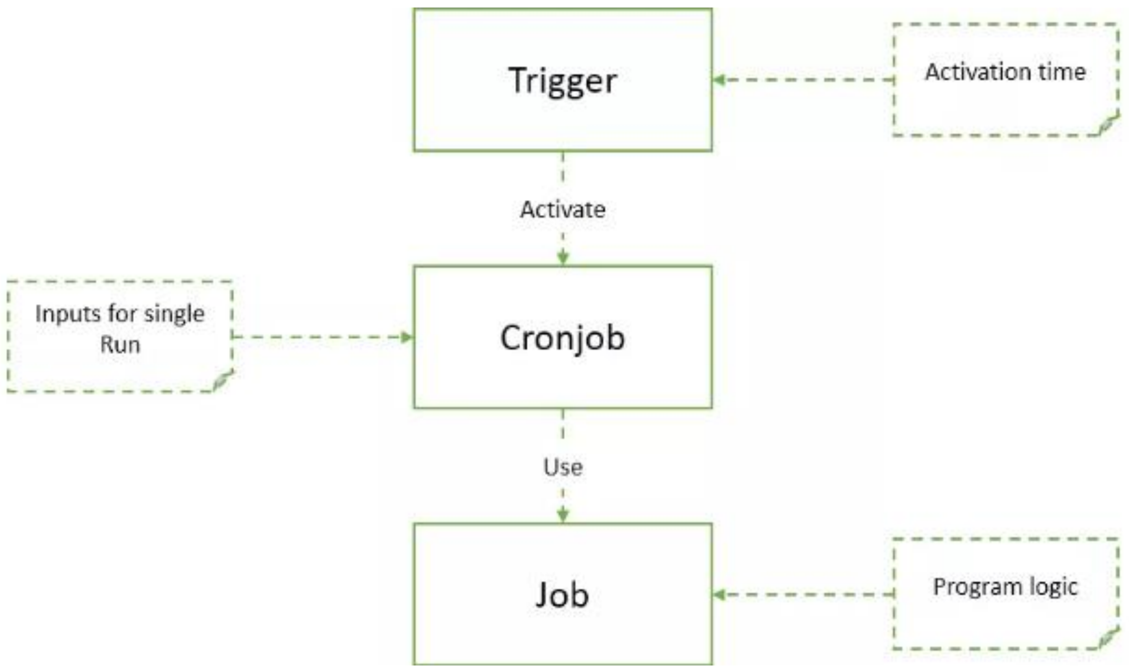
2) Cronjob = This hold the business logic which needs to be executed at specified time.

3) Job = This consists of logic which defines by "**Job Performable**".

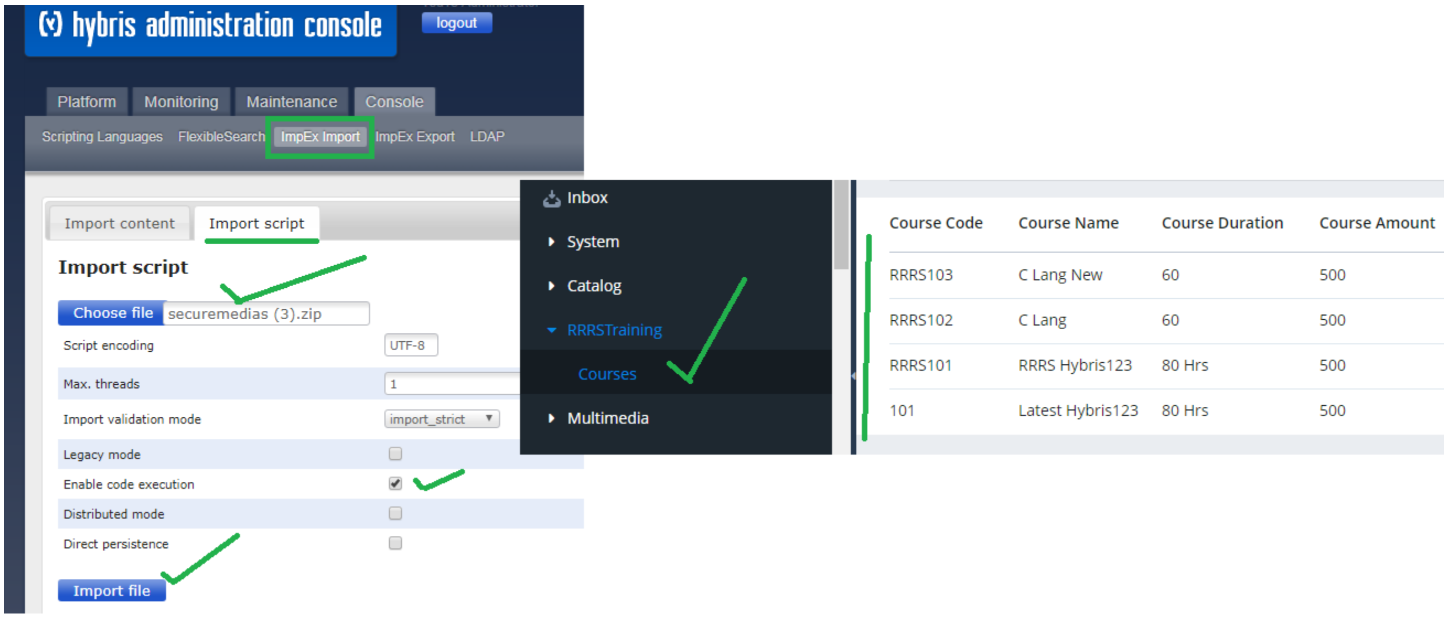
2 Ways: -

(a) Create a class which extends **AbstractJobPerformable**

(b) Create a class which implements **JobPerformable**



Example: - We already created "**Courses**" itemtype (Table name = **TrainingCourses**). It has the records.

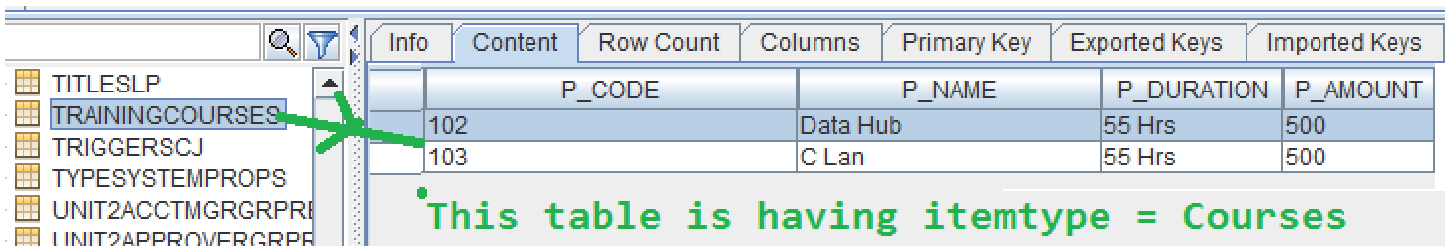


Import script

Choose file securemedias (3).zip

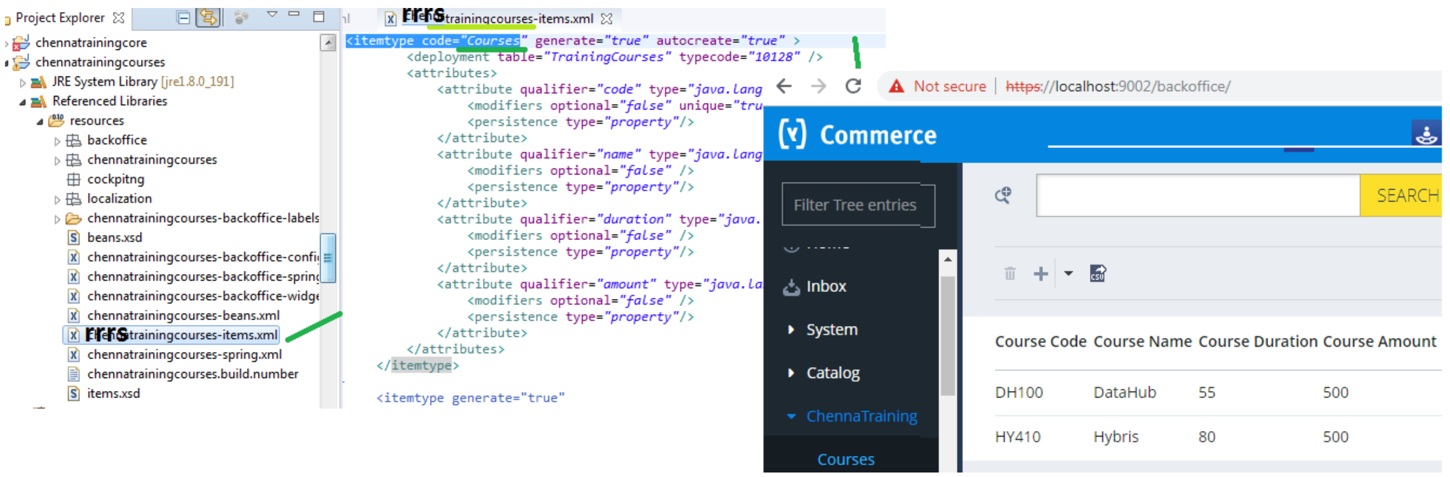
Script encoding	UTF-8
Max. threads	1
Import validation mode	import_strict
Legacy mode	☐
Enable code execution	✓
Distributed mode	☐
Direct persistence	☐

Course Code	Course Name	Course Duration	Course Amount
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500
101	Latest Hybris123	80 Hrs	500



P_CODE	P_NAME	P_DURATION	P_AMOUNT
102	Data Hub	55 Hrs	500
103	C Lan	55 Hrs	500

This table is having itemtype = Courses


Project Explorer
chennatrainingcore
chennatrainingcourses
JRE System Library [jre1.8.0_191]
Referenced Libraries
resources
backoffice
chennatrainingcourses
cockpitng
localization
chennatrainingcourses-backoffice-labels
beans.xsd
chennatrainingcourses-backoffice-confi
chennatrainingcourses-backoffice-spring
chennatrainingcourses-backoffice-widge
chennatrainingcourses-beans.xml
rrrstrainingcourses-items.xml
chennatrainingcourses-spring.xml
chennatrainingcourses.build.number
items.xsd
rrrstrainingcourses-items.xml
Not secure
https://localhost:9002/backoffice/
Commerce
Filter Tree entries
SEARCH
Inbox
System
Catalog
ChennaTraining
Courses

```
<itemtype code="Courses" generate="true" autocreate="true" >
        <deployment table="TrainingCourses" typecode="10128" />
        <attributes>
            <attribute qualifier="code" type="java.lang
                <modifiers optional="false" unique="tru
                <persistence type="property"/>
            </attribute>
            <attribute qualifier="name" type="java.lang
                <modifiers optional="false" />
                <persistence type="property"/>
            </attribute>
            <attribute qualifier="duration" type="java.
                <modifiers optional="false" />
                <persistence type="property"/>
            </attribute>
            <attribute qualifier="amount" type="java.la
                <modifiers optional="false" />
                <persistence type="property"/>
            </attribute>
        </attributes>
    </itemtype>

    <itemtype generate="true"
```

Course Code	Course Name	Course Duration	Course Amount
DH100	DataHub	55	500
HY410	Hybris	80	500

Business Scenario = Read “**TrainingCourses**” table records (Or) **Courses** Itemtype records: -

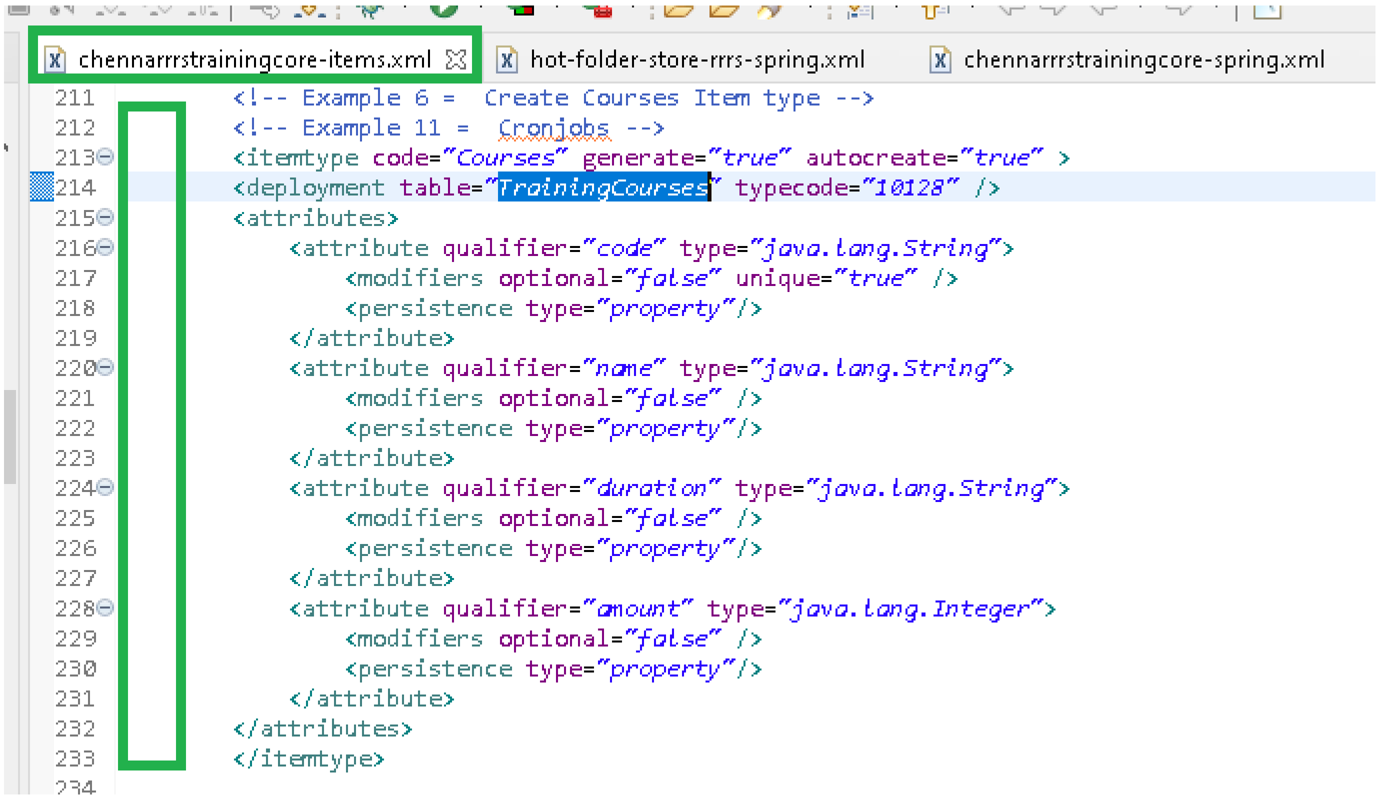
1) Display records in console every 1 min (Display in Logs every 1 min)

2) Email those Records every 1 min

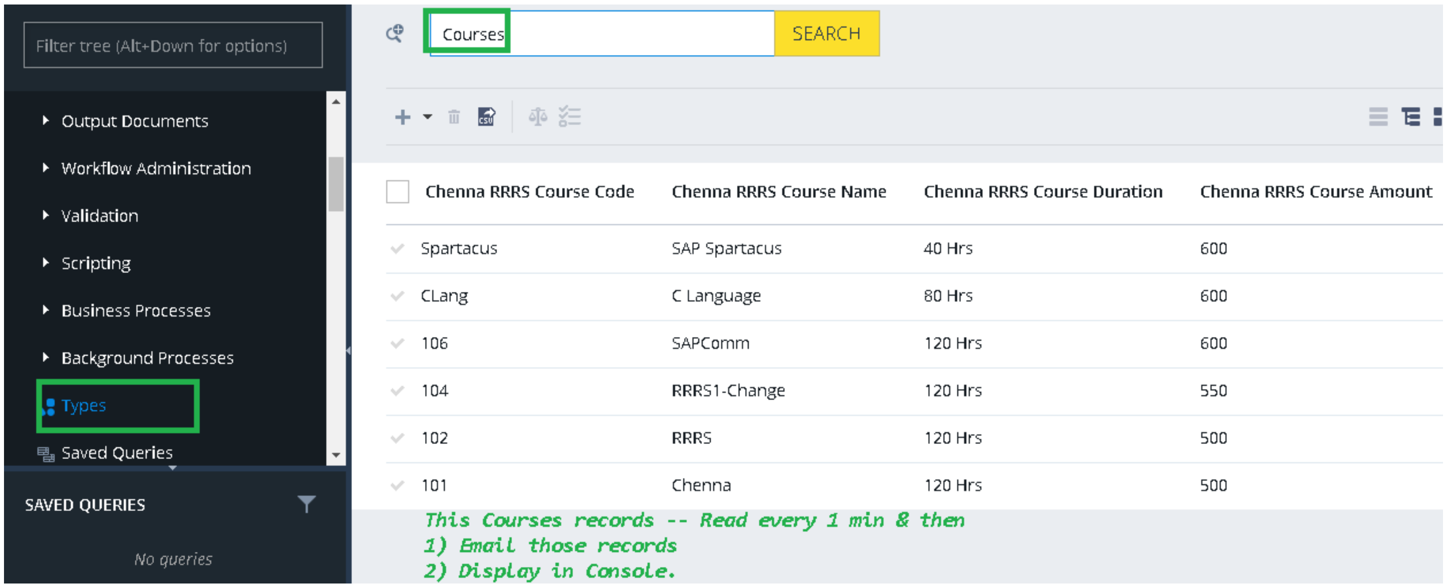
Step 1 = Create **Courses** Itemtype (or) **TrainingCourses** table.

Also – Insert the records.

Example =


chennarrrstrainingcore-items.xml
hot-folder-store-rrrs-spring.xml
chennarrrstrainingcore-spring.xml

```xml
<!-- Example 6 =  Create Courses Item type -->
<!-- Example 11 =  Cronjobs -->
<itemtype code="Courses" generate="true" autocreate="true" >
<deployment table="TrainingCourses" typecode="10128" />
<attributes>
    <attribute qualifier="code" type="java.lang.String">
        <modifiers optional="false" unique="true" />
        <persistence type="property"/>
    </attribute>
    <attribute qualifier="name" type="java.lang.String">
        <modifiers optional="false" />
        <persistence type="property"/>
    </attribute>
    <attribute qualifier="duration" type="java.lang.String">
        <modifiers optional="false" />
        <persistence type="property"/>
    </attribute>
    <attribute qualifier="amount" type="java.lang.Integer">
        <modifiers optional="false" />
        <persistence type="property"/>
    </attribute>
</attributes>
</itemtype>
```

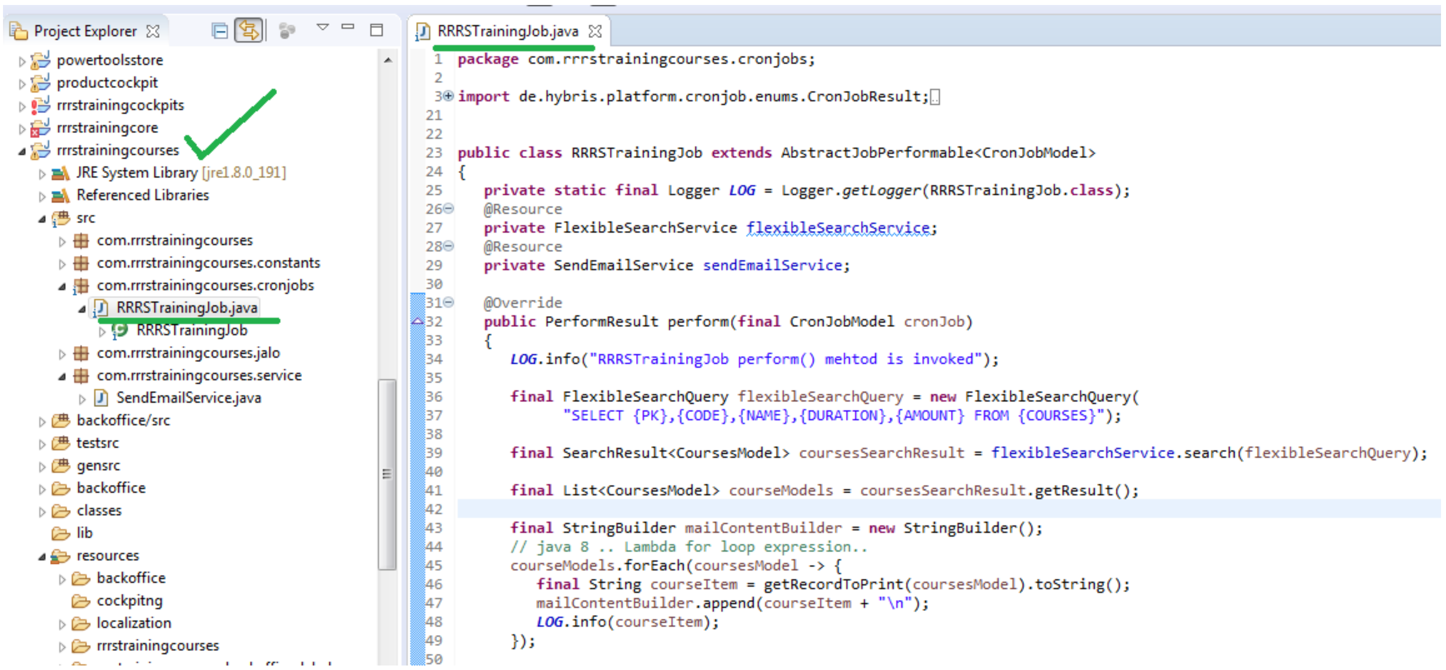


Chenna RRRS Course Code	Chenna RRRS Course Name	Chenna RRRS Course Duration	Chenna RRRS Course Amount
Spartacus	SAP Spartacus	40 Hrs	600
CLang	C Language	80 Hrs	600
106	SAPComm	120 Hrs	600
104	RRRS1-Change	120 Hrs	550
102	RRRS	120 Hrs	500
101	Chenna	120 Hrs	500

This Courses records -- Read every 1 min & then
1) Email those records
2) Display in Console.

Step 2 = Create new class called “**RRRSTrainingJob.java**” by extending “**AbstractJobPerformable**”.

AbstractJobPerformable is having a method called “**perform()**” & we need to override this method our own business logic.

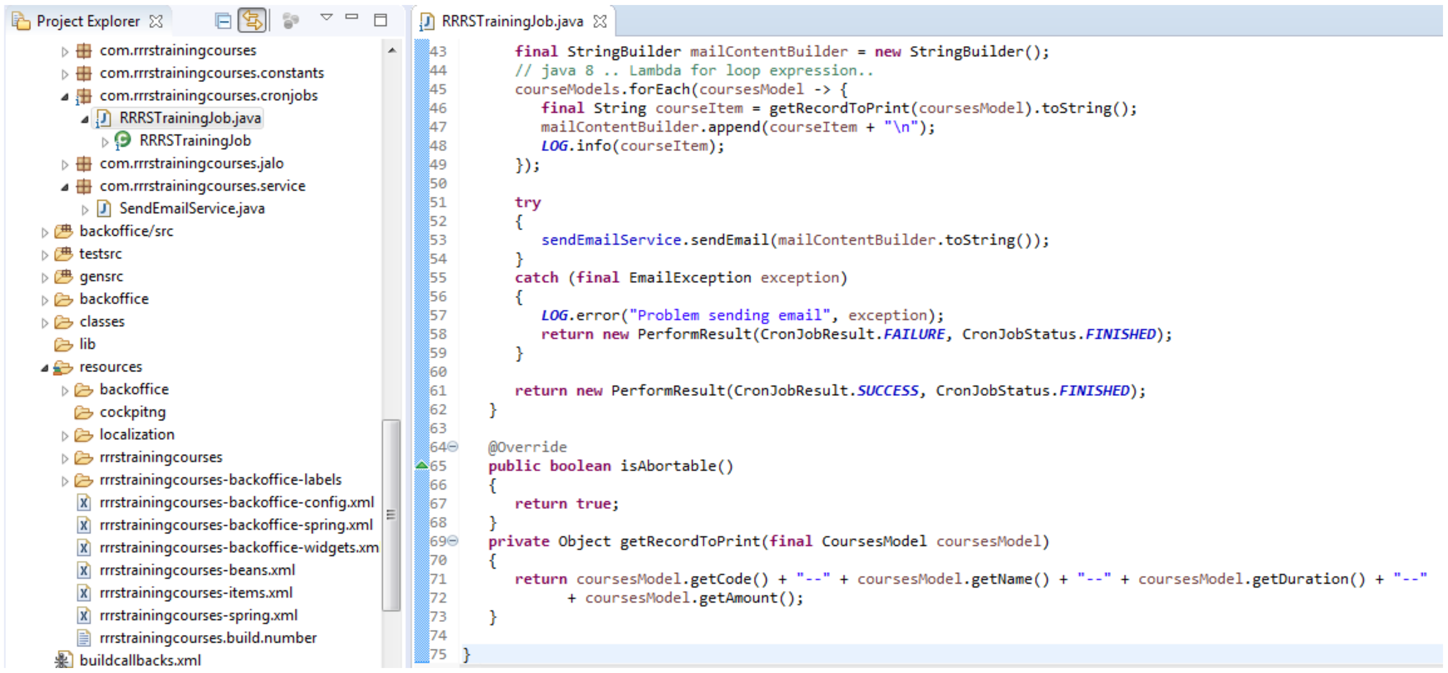


Project Explorer

- powertoolsstore
- productcockpit
- rrrstrainingcockpits
- rrrstrainingcore
- rrrstrainingcourses
  - JRE System Library [jre1.8.0_191]
  - Referenced Libraries
  - src
    - com.rrrstrainingcourses
    - com.rrrstrainingcourses.constants
    - com.rrrstrainingcourses.cronjobs
      - RRRSTrainingJob.java
        - RRRSTrainingJob
    - com.rrrstrainingcourses.jalo
    - com.rrrstrainingcourses.service
      - SendEmailService.java
  - backoffice/src
  - testsrc
  - gensrc
  - backoffice
  - classes
  - lib
  - resources
    - backoffice
    - cockpitng
    - localization
    - rrrstrainingcourses

RRRSTrainingJob.java

```java
package com.rrrstrainingcourses.cronjobs;

import de.hybris.platform.cronjob.enums.CronJobResult;[]


public class RRRSTrainingJob extends AbstractJobPerformable<CronJobModel>
{
   private static final Logger LOG = Logger.getLogger(RRRSTrainingJob.class);
   @Resource
   private FlexibleSearchService flexibleSearchService;
   @Resource
   private SendEmailService sendEmailService;

   @Override
   public PerformResult perform(final CronJobModel cronJob)
   {
      LOG.info("RRRSTrainingJob perform() mehtod is invoked");

      final FlexibleSearchQuery flexibleSearchQuery = new FlexibleSearchQuery(
            "SELECT {PK},{CODE},{NAME},{DURATION},{AMOUNT} FROM {COURSES}");

      final SearchResult<CoursesModel> coursesSearchResult = flexibleSearchService.search(flexibleSearchQuery);

      final List<CoursesModel> courseModels = coursesSearchResult.getResult();

      final StringBuilder mailContentBuilder = new StringBuilder();
      // java 8 .. Lambda for loop expression..
      courseModels.forEach(coursesModel -> {
         final String courseItem = getRecordToPrint(coursesModel).toString();
         mailContentBuilder.append(courseItem + "\n");
         LOG.info(courseItem);
      });

```



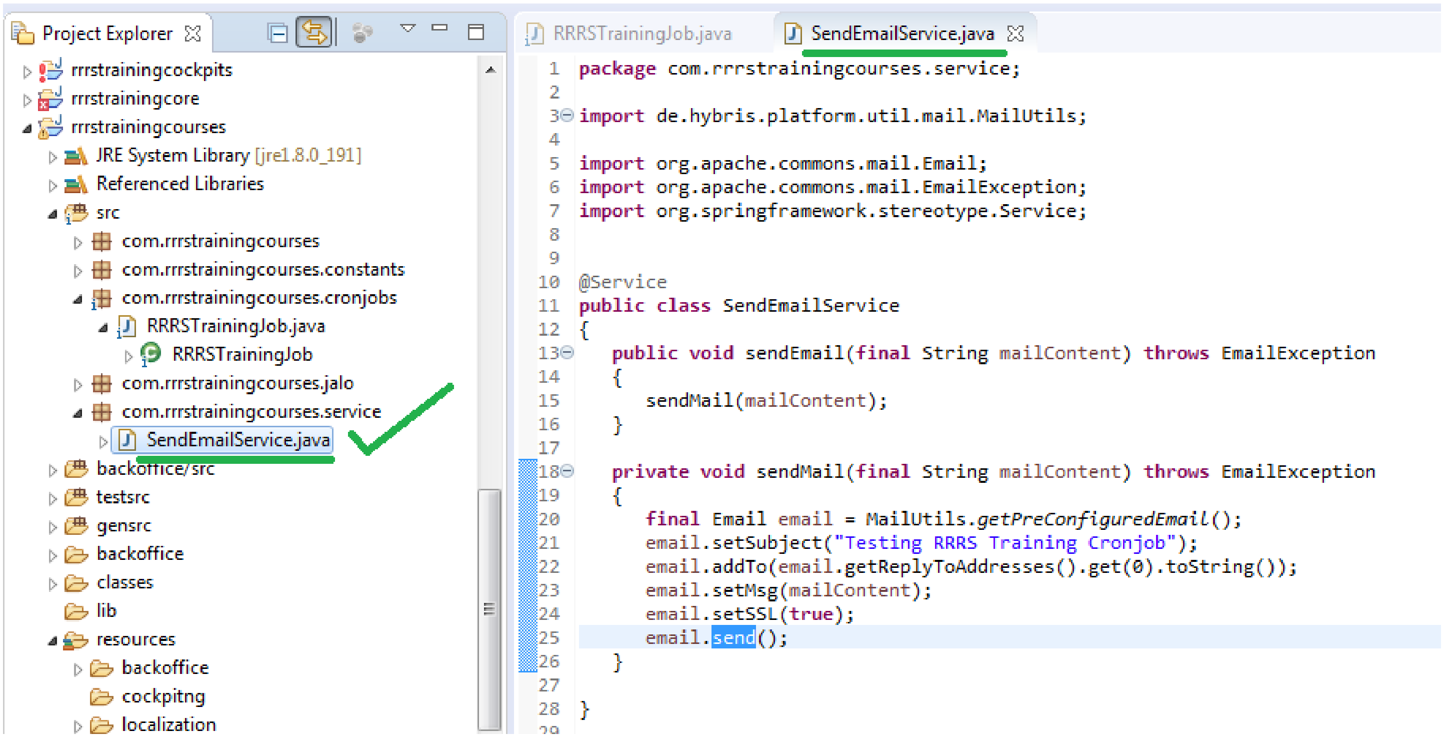
Project Explorer

- com.rrrstrainingcourses
- com.rrrstrainingcourses.constants
- com.rrrstrainingcourses.cronjobs
  - RRRSTrainingJob.java
    - RRRSTrainingJob
- com.rrrstrainingcourses.jalo
- com.rrrstrainingcourses.service
  - SendEmailService.java
- backoffice/src
- testsrc
- gensrc
- backoffice
- classes
- lib
- resources
  - backoffice
  - cockpitng
  - localization
  - rrrstrainingcourses
  - rrrstrainingcourses-backoffice-labels
  - rrrstrainingcourses-backoffice-config.xml
  - rrrstrainingcourses-backoffice-spring.xml
  - rrrstrainingcourses-backoffice-widgets.xm
  - rrrstrainingcourses-beans.xml
  - rrrstrainingcourses-items.xml
  - rrrstrainingcourses-spring.xml
  - rrrstrainingcourses.build.number
- buildcallbacks.xml

RRRSTrainingJob.java

```java
      final StringBuilder mailContentBuilder = new StringBuilder();
      // java 8 .. Lambda for loop expression..
      courseModels.forEach(coursesModel -> {
         final String courseItem = getRecordToPrint(coursesModel).toString();
         mailContentBuilder.append(courseItem + "\n");
         LOG.info(courseItem);
      });

      try
      {
         sendEmailService.sendEmail(mailContentBuilder.toString());
      }
      catch (final EmailException exception)
      {
         LOG.error("Problem sending email", exception);
         return new PerformResult(CronJobResult.FAILURE, CronJobStatus.FINISHED);
      }

      return new PerformResult(CronJobResult.SUCCESS, CronJobStatus.FINISHED);
   }

   @Override
   public boolean isAbortable()
   {
      return true;
   }
   private Object getRecordToPrint(final CoursesModel coursesModel)
   {
      return coursesModel.getCode() + "--" + coursesModel.getName() + "--" + coursesModel.getDuration() + "--"
            + coursesModel.getAmount();
   }

}
```

Step 3 = Create an Email Service called "**SendEmailService.java**"

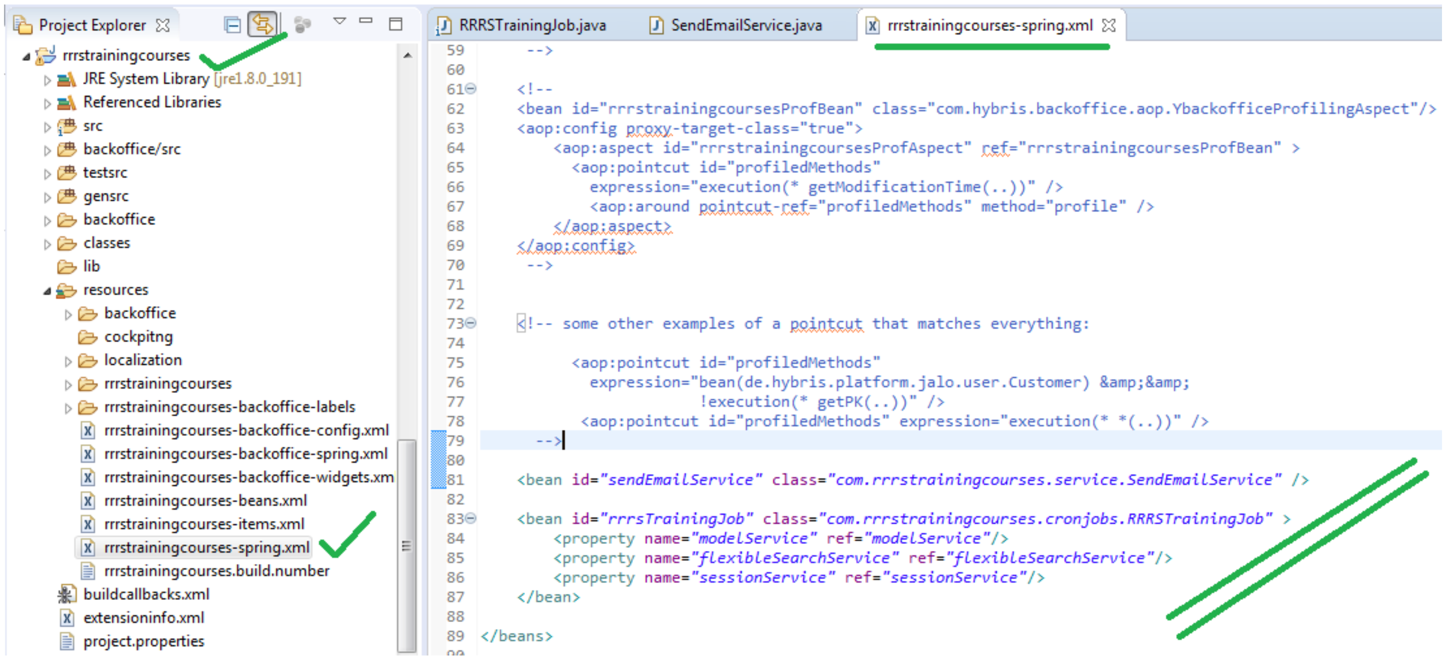


```java
package com.rrrstrainingcourses.service;

import de.hybris.platform.util.mail.MailUtils;

import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.springframework.stereotype.Service;


@Service
public class SendEmailService
{
    public void sendEmail(final String mailContent) throws EmailException
    {
        sendMail(mailContent);
    }

    private void sendMail(final String mailContent) throws EmailException
    {
        final Email email = MailUtils.getPreConfiguredEmail();
        email.setSubject("Testing RRRS Training Cronjob");
        email.addTo(email.getReplyToAddresses().get(0).toString());
        email.setMsg(mailContent);
        email.setSSL(true);
        email.send();
    }

}
```

Step 4 = we have written "**RRRSTrainingJob.java**" file

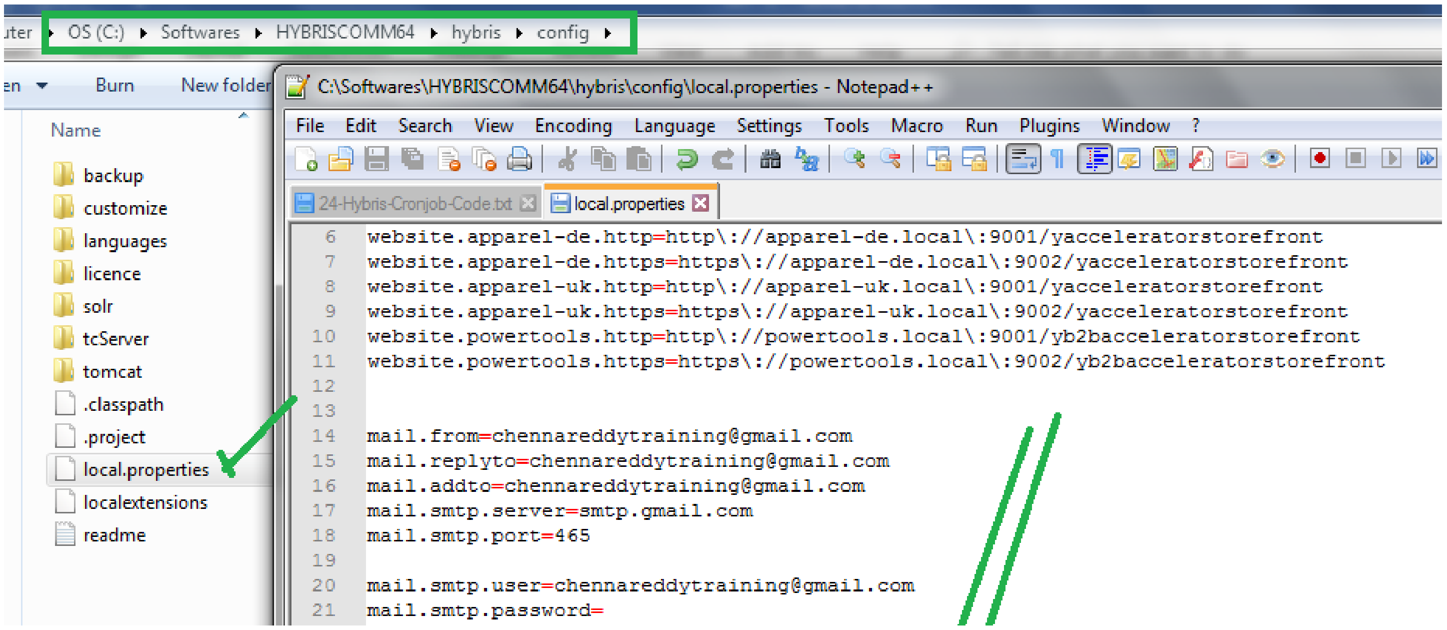
& "**SendEmailService.java**".

=== So, it's time to register those classes (or) **Beans**.


Project Explorer
rrrstrainingcourses
JRE System Library [jre1.8.0_191]
Referenced Libraries
src
backoffice/src
testsrc
gensrc
backoffice
classes
lib
resources
backoffice
cockpitng
localization
rrrstrainingcourses
rrrstrainingcourses-backoffice-labels
rrrstrainingcourses-backoffice-config.xml
rrrstrainingcourses-backoffice-spring.xml
rrrstrainingcourses-backoffice-widgets.xm
rrrstrainingcourses-beans.xml
rrrstrainingcourses-items.xml
rrrstrainingcourses-spring.xml
rrrstrainingcourses.build.number
buildcallbacks.xml
extensioninfo.xml
project.properties
RRRSTrainingJob.java
SendEmailService.java
rrrstrainingcourses-spring.xml

```xml
     -->

    <!--
    <bean id="rrrstrainingcoursesProfBean" class="com.hybris.backoffice.aop.YbackofficeProfilingAspect"/>
    <aop:config proxy-target-class="true">
        <aop:aspect id="rrrstrainingcoursesProfAspect" ref="rrrstrainingcoursesProfBean" >
          <aop:pointcut id="profiledMethods"
            expression="execution(* getModificationTime(..))" />
            <aop:around pointcut-ref="profiledMethods" method="profile" />
        </aop:aspect>
    </aop:config>
     -->


    <!-- some other examples of a pointcut that matches everything:

          <aop:pointcut id="profiledMethods"
            expression="bean(de.hybris.platform.jalo.user.Customer) &amp;&amp;
                        !execution(* getPK(..))" />
           <aop:pointcut id="profiledMethods" expression="execution(* *(..))" />
      -->

    <bean id="sendEmailService" class="com.rrrstrainingcourses.service.SendEmailService" />

    <bean id="rrrsTrainingJob" class="com.rrrstrainingcourses.cronjobs.RRRSTrainingJob" >
        <property name="modelService" ref="modelService"/>
        <property name="flexibleSearchService" ref="flexibleSearchService"/>
        <property name="sessionService" ref="sessionService"/>
    </bean>

</beans>
```

Step 5 = Define email properties in “local.properties” file

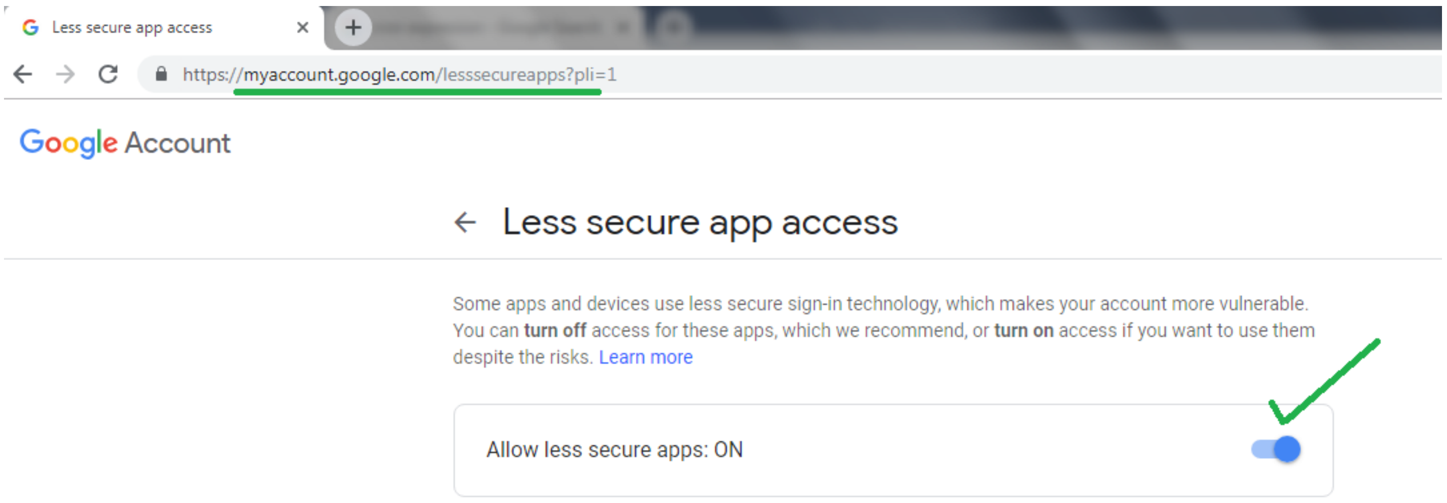


OS (C:) ▸ Softwares ▸ HYBRISCOMM64 ▸ hybris ▸ config
backup
customize
languages
licence
solr
tcServer
tomcat
.classpath
.project
local.properties
localextensions
readme
C:\Softwares\HYBRISCOMM64\hybris\config\local.properties - Notepad++
24-Hybris-Cronjob-Code.txt
local.properties

```
website.apparel-de.http=http\://apparel-de.local\:9001/yacceleratorstorefront
website.apparel-de.https=https\://apparel-de.local\:9002/yacceleratorstorefront
website.apparel-uk.http=http\://apparel-uk.local\:9001/yacceleratorstorefront
website.apparel-uk.https=https\://apparel-uk.local\:9002/yacceleratorstorefront
website.powertools.http=http\://powertools.local\:9001/yb2bacceleratorstorefront
website.powertools.https=https\://powertools.local\:9002/yb2bacceleratorstorefront


mail.from=chennareddytraining@gmail.com
mail.replyto=chennareddytraining@gmail.com
mail.addto=chennareddytraining@gmail.com
mail.smtp.server=smtp.gmail.com
mail.smtp.port=465

mail.smtp.user=chennareddytraining@gmail.com
mail.smtp.password=
```

Step 6 = Port **465** works only if we make allow less secure aps “**ON**” in your Gmail account.

URL = https://myaccount.google.com/lesssecureapps?pli=1



Less secure app access

Some apps and devices use less secure sign-in technology, which makes your account more vulnerable. You can **turn off** access for these apps, which we recommend, or **turn on** access if you want to use them despite the risks. Learn more

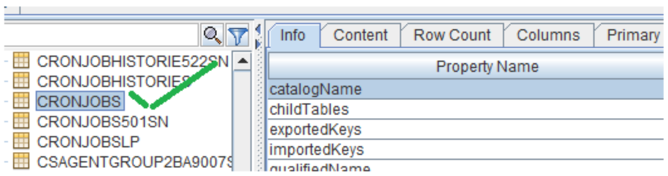
Allow less secure apps: ON

Step 7 = Do the build (**ant clean all**)

Step 9 = Start the Server (**hybrisserver.bat**)

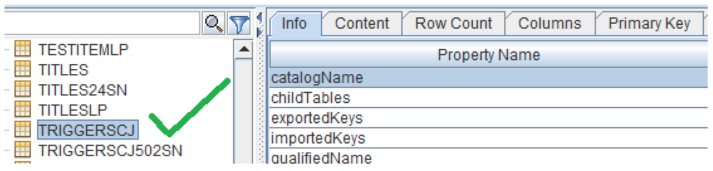
Step 9 = Perform the Platform Update (**hAC – Update**)

Step 10 = Create **Cronjob** & Register Cronjob using **Trigger**.



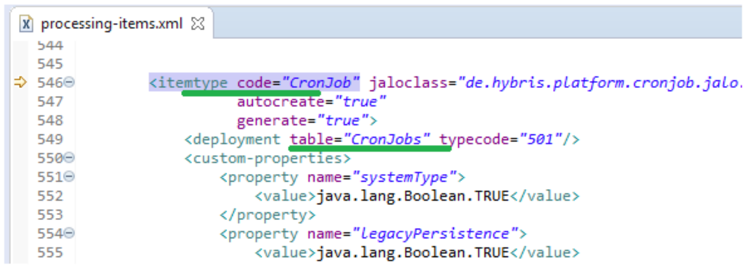
CRONJOBHISTORIE522SN
CRONJOBHISTORIES
CRONJOBS
CRONJOBS501SN
CRONJOBSLP
CSAGENTGROUP2BA9007S

Property Name
catalogName
childTables
exportedKeys
importedKeys
qualifiedName



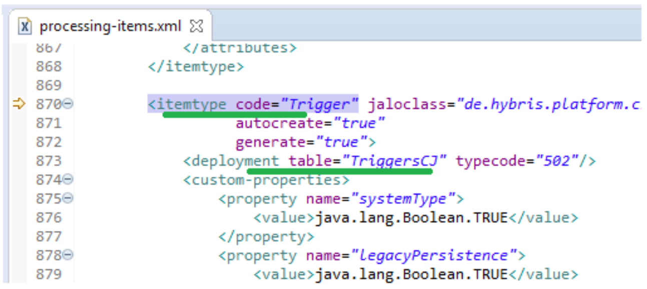
TESTITEMLP
TITLES
TITLES24SN
TITLESLP
TRIGGERSCJ
TRIGGERSCJ502SN

Property Name
catalogName
childTables
exportedKeys
importedKeys
qualifiedName



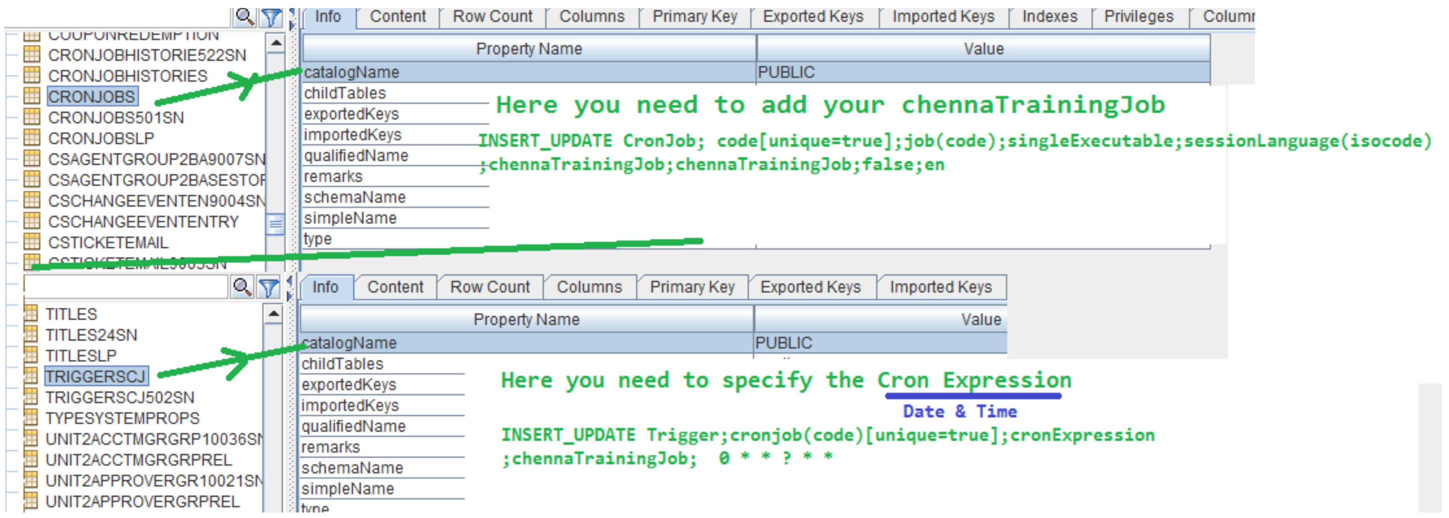
processing-items.xml

```xml

        <itemtype code="CronJob" jaloclass="de.hybris.platform.cronjob.jalo.
                autocreate="true"
                generate="true">
            <deployment table="CronJobs" typecode="501"/>
            <custom-properties>
                <property name="systemType">
                    <value>java.lang.Boolean.TRUE</value>
                </property>
                <property name="legacyPersistence">
                    <value>java.lang.Boolean.TRUE</value>
```



processing-items.xml

```xml
            </attributes>
        </itemtype>

        <itemtype code="Trigger" jaloclass="de.hybris.platform.c
                autocreate="true"
                generate="true">
            <deployment table="TriggersCJ" typecode="502"/>
            <custom-properties>
                <property name="systemType">
                    <value>java.lang.Boolean.TRUE</value>
                </property>
                <property name="legacyPersistence">
                    <value>java.lang.Boolean.TRUE</value>
```



COUPONREDEMPTION
CRONJOBHISTORIE522SN
CRONJOBHISTORIES
CRONJOBS
CRONJOBS501SN
CRONJOBSLP
CSAGENTGROUP2BA9007SN
CSAGENTGROUP2BASESTOR
CSCHANGEEVENTEN9004SN
CSCHANGEEVENTENTRY
CSTICKETEMAIL
CSTICKETEMAIL5005SN

Property Name	Value
catalogName	PUBLIC
childTables	
exportedKeys	
importedKeys	
qualifiedName	
remarks	
schemaName	
simpleName	
type	

Here you need to add your chennaTrainingJob

```
INSERT_UPDATE CronJob; code[unique=true];job(code);singleExecutable;sessionLanguage(isocode)
;chennaTrainingJob;chennaTrainingJob;false;en
```

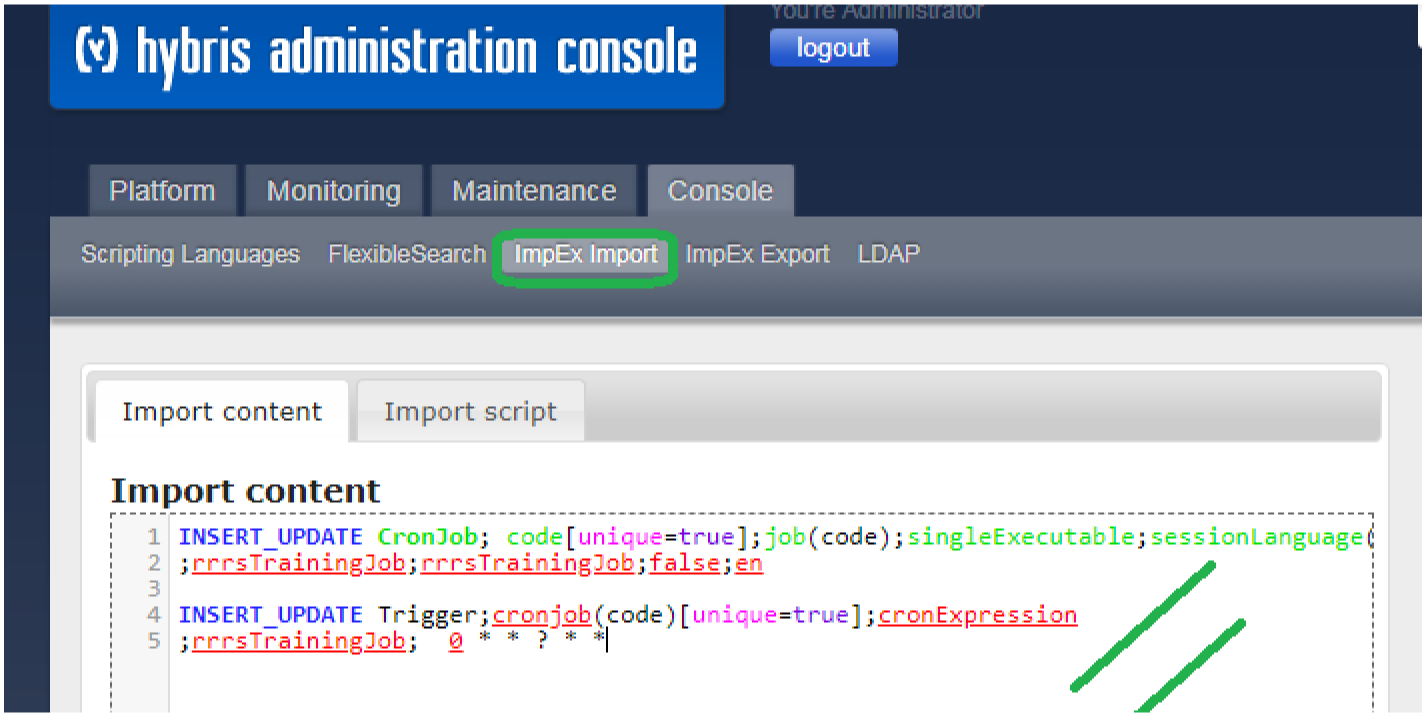
TITLES
TITLES24SN
TITLESLP
TRIGGERSCJ
TRIGGERSCJ502SN
TYPESYSTEMPROPS
UNIT2ACCTMGRGRP10036SN
UNIT2ACCTMGRGRPREL
UNIT2APPROVERGR10021SN
UNIT2APPROVERGRPREL

Property Name	Value
catalogName	PUBLIC
childTables	
exportedKeys	
importedKeys	
qualifiedName	
remarks	
schemaName	
simpleName	
type	

Here you need to specify the Cron Expression

Date & Time

```
INSERT_UPDATE Trigger;cronjob(code)[unique=true];cronExpression
;chennaTrainingJob;  0 * * ? * *
```

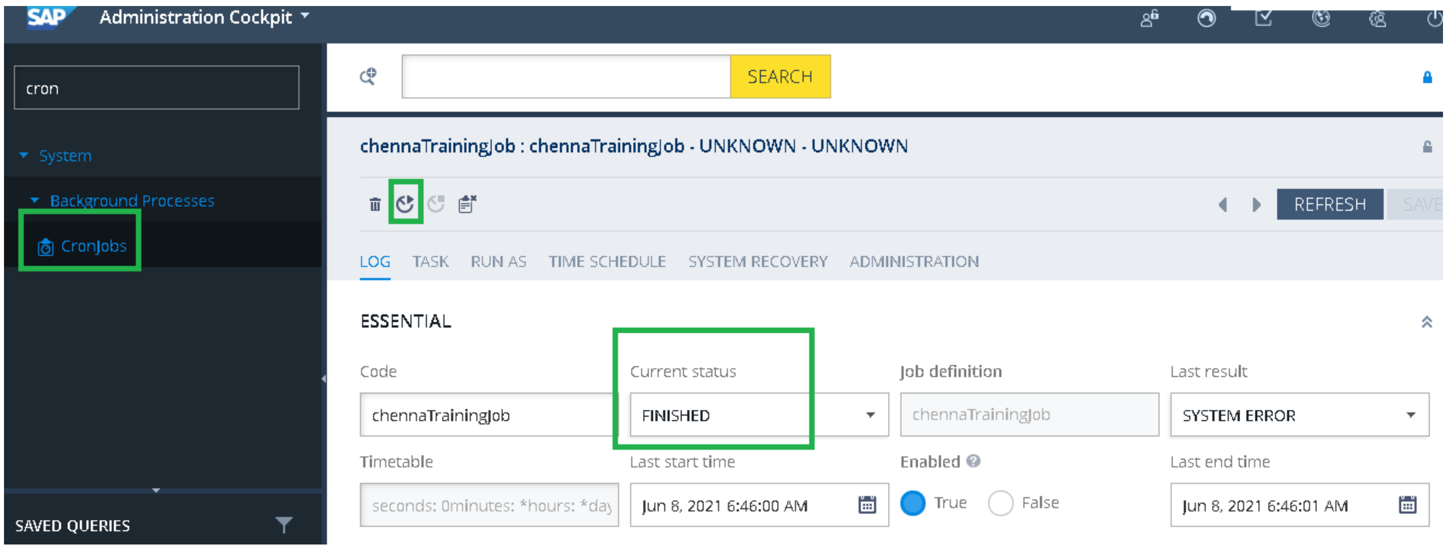


Import content

```
INSERT_UPDATE CronJob; code[unique=true];job(code);singleExecutable;sessionLanguage(
;rrrsTrainingJob;rrrsTrainingJob;false;en

INSERT_UPDATE Trigger;cronjob(code)[unique=true];cronExpression
;rrrsTrainingJob;  0 * * ? * *
```

```
INSERT_UPDATE CronJob; code[unique=true];job(code);singleExecutable;sessionLanguage(isocode)
;rrrsTrainingJob;rrrsTrainingJob;false;en

INSERT_UPDATE Trigger;cronjob(code)[unique=true];cronExpression
;rrrsTrainingJob;  0 * * ? * *
```

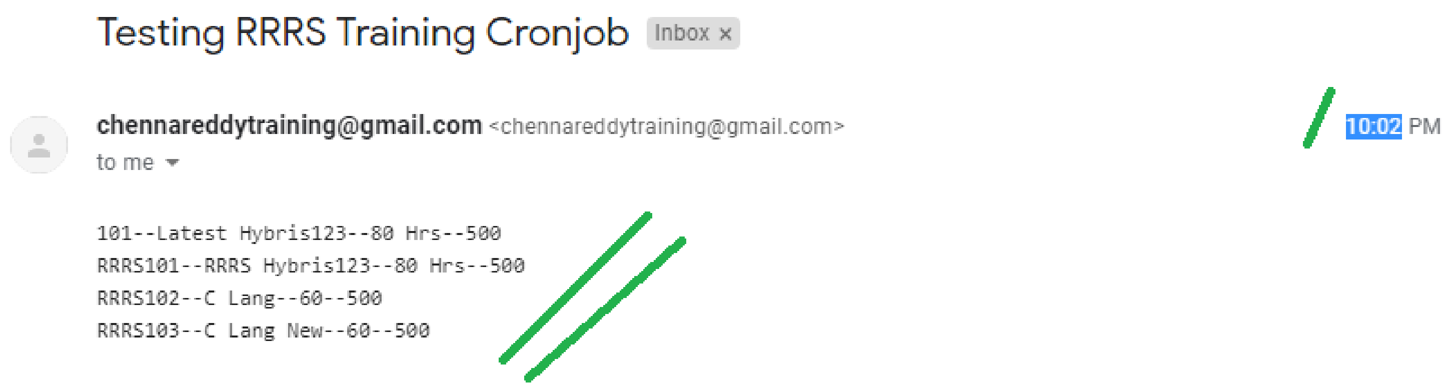
Step 11 = Test the Results: -



chennaTrainingJob : chennaTrainingJob - UNKNOWN - UNKNOWN

ESSENTIAL

Code	Current status	Job definition	Last result
chennaTrainingJob	FINISHED	chennaTrainingJob	SYSTEM ERROR
Timetable	**Last start time**	**Enabled**	**Last end time**
seconds: 0minutes: *hours: *day	Jun 8, 2021 6:46:00 AM	[✓ True] False	Jun 8, 2021 6:46:01 AM

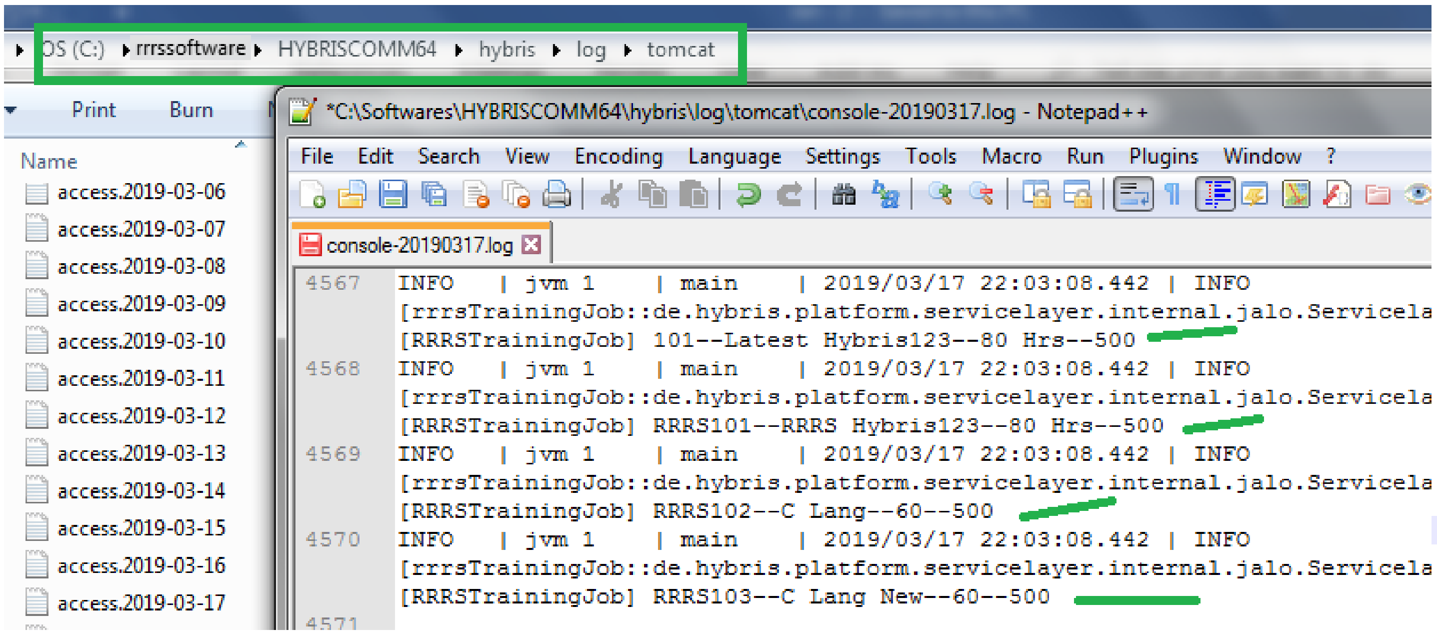


Testing RRRS Training Cronjob Inbox ×

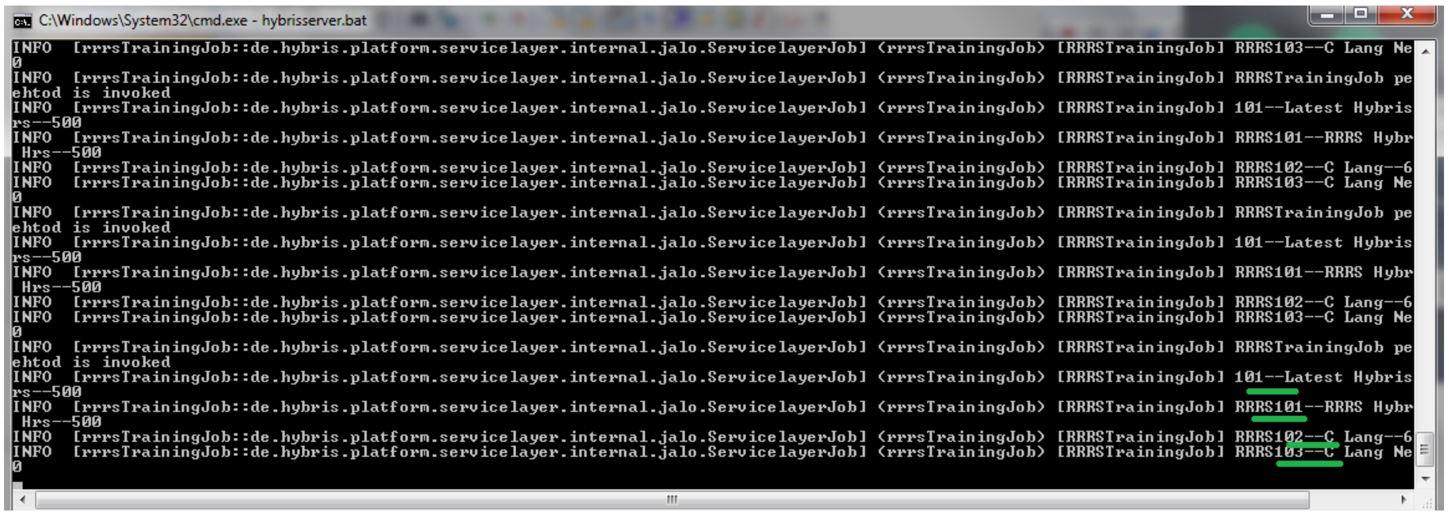
chennareddytraining@gmail.com <chennareddytraining@gmail.com> 10:02 PM

to me

```
101--Latest Hybris123--80 Hrs--500
RRRS101--RRRS Hybris123--80 Hrs--500
RRRS102--C Lang--60--500
RRRS103--C Lang New--60--500
```



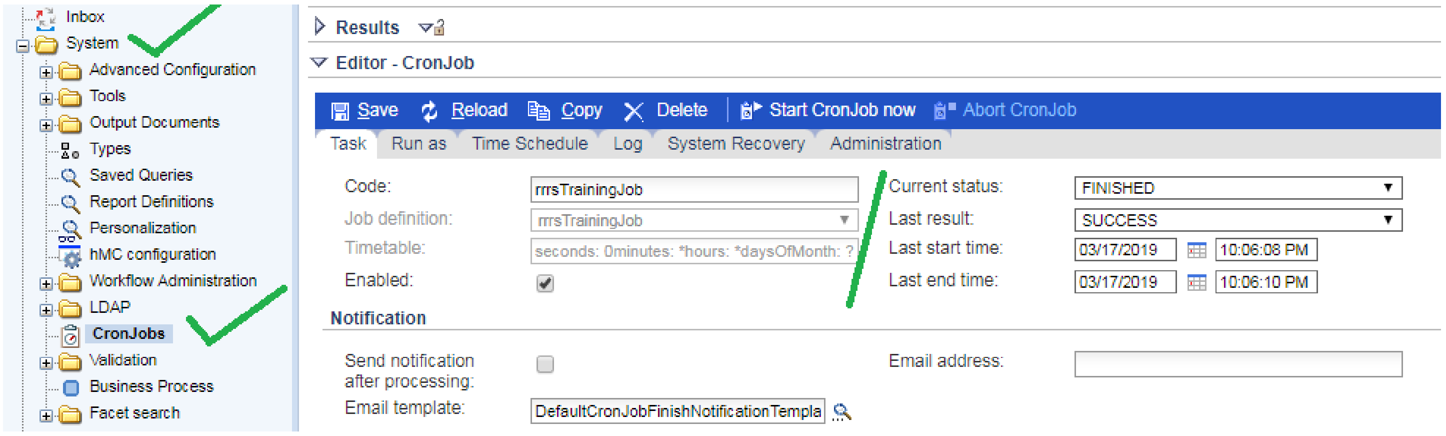
```
INFO   | jvm 1    | main    | 2019/03/17 22:03:08.442 | INFO
[rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.Servicela
[RRRSTrainingJob] 101--Latest Hybris123--80 Hrs--500
INFO   | jvm 1    | main    | 2019/03/17 22:03:08.442 | INFO
[rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.Servicela
[RRRSTrainingJob] RRRS101--RRRS Hybris123--80 Hrs--500
INFO   | jvm 1    | main    | 2019/03/17 22:03:08.442 | INFO
[rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.Servicela
[RRRSTrainingJob] RRRS102--C Lang--60--500
INFO   | jvm 1    | main    | 2019/03/17 22:03:08.442 | INFO
[rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.Servicela
[RRRSTrainingJob] RRRS103--C Lang New--60--500
```



C:\Windows\System32\cmd.exe - hybrisserver.bat

```
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS103--C Lang Ne
0
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRSTrainingJob pe
ehtod is invoked
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] 101--Latest Hybris
rs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS101--RRRS Hybr
 Hrs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS102--C Lang--6
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS103--C Lang Ne
0
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRSTrainingJob pe
ehtod is invoked
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] 101--Latest Hybris
rs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS101--RRRS Hybr
 Hrs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS102--C Lang--6
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS103--C Lang Ne
0
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRSTrainingJob pe
ehtod is invoked
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] 101--Latest Hybris
rs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS101--RRRS Hybr
 Hrs--500
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS102--C Lang--6
INFO  [rrrsTrainingJob::de.hybris.platform.servicelayer.internal.jalo.ServicelayerJob] (rrrsTrainingJob) [RRRSTrainingJob] RRRS103--C Lang Ne
0
```

Note: - We can also run the Cronjobs manually.

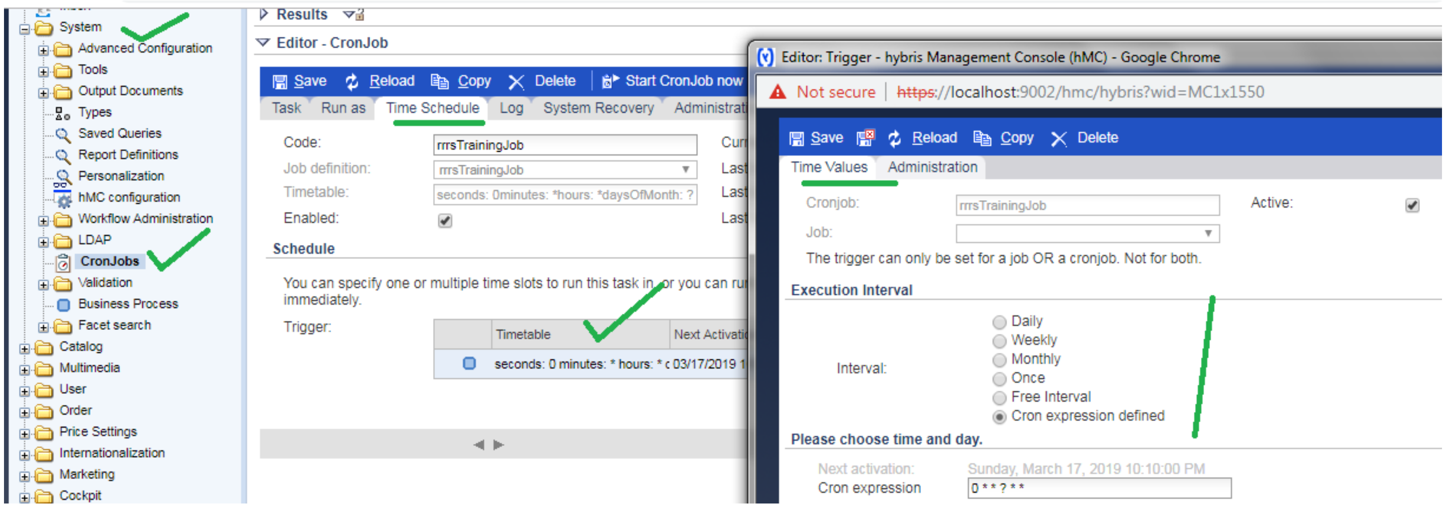


Code: rrrsTrainingJob
Job definition: rrrsTrainingJob
Timetable: seconds: 0minutes: *hours: *daysOfMonth: ?
Enabled: ✓

Current status: FINISHED
Last result: SUCCESS
Last start time: 03/17/2019 10:06:08 PM
Last end time: 03/17/2019 10:06:10 PM

Notification

Send notification after processing: ☐
Email address:
Email template: DefaultCronJobFinishNotificationTempla



Code: rrrsTrainingJob
Job definition: rrrsTrainingJob
Timetable: seconds: 0minutes: *hours: *daysOfMonth: ?
Enabled: ✓

Schedule

You can specify one or multiple time slots to run this task in, or you can ru… immediately.

Trigger:

	Timetable	Next Activati…
▢	seconds: 0 minutes: * hours: * c	03/17/2019 1…

Editor: Trigger - hybris Management Console (hMC) - Google Chrome
Not secure | ~~https~~://localhost:9002/hmc/hybris?wid=MC1x1550

Cronjob: rrrsTrainingJob
Active: ✓
Job:
The trigger can only be set for a job OR a cronjob. Not for both.

Execution Interval

Interval: ○ Daily ○ Weekly ○ Monthly ○ Once ○ Free Interval [✓ Cron expression defined]

Please choose time and day.

Next activation: Sunday, March 17, 2019 10:10:00 PM
Cron expression: 0 * * ? * *

Composite Cron Jobs: -

https://www.stackextend.com/hybris/use-composite-cronjob-in-hybris/

Q: How to Start a CronJob? = There are different ways to start a cronjob which are given below :

- Manually start using HMC = → hmc → system → cronjobs → select CronJob → "StartCronJobNow".
- Automatically running the CronJob Through Impex file
- Using the ant command → ant runcronjob -d cronjob="CronJobName "
- Using the javacode using the CronJob services we can run the cronjob.

Q = What are the major different types of the preconfigured cronjobs in "SAP Comm"?

- SOLR and Lucene related: indexing, updating, removing data
- Clean up unnecessary data from the database or file system
- Product Catalog synchronization
- Regular data export (Product, Price, Inventory, Order Status, and …. Import / Export).
- Workflow
- Impex import.

Q = How to stop a cronjob? = We can stop the cronjob by following ways:

- Using the abort method in the java code. It is done automatically after performing the CronJob.
- Manually from hmc we can stop the CronJob.

Q = Where to see the created CronJob? = hmc → System → CronJobs

Q = How to see the Job Details? =

```
select * from {servicelayerjob} where {code} = 'RRRSTraininglJob'
```

Q = How to Run Cron Job through Ant? =

```
ant runcronjob -Dcronjob=rrrsCronJob -Dtenant=master
```

Q = What are setting session related attributes to the cron job?

Some time we write a Cron job whose logic requires some session attributes like user, sessionLanguage and sessionCurrency etc.

So how do we set these attributes to the cron job so that we can access them while writing the logic of the cron job ?

It can be done in any one of the 2 ways listed below

1) Set the session attributes through impex
2) Set the session attributes through code

Q = The "SAP Comm" commercefacades extension facades mostly return?

a) Data objects b) Model objects c) Data model d) none

Q = "SAP Comm" ServiceLayer Models should not be used as part of a facade interface, maintaining a clean abstraction of the?

a) Business layer and the persistence layer

b) Business layer and the presentation layer

c) Persistence layer and the presentation layer

d) All

Q = All converters should be Spring configured only and should use the ------------------ base class

a) AbstractConverter b) DefaultConverter

c) a & b d) none of above

Q = -------Is an implementation of a Populator pipeline where each population step is evaluated against a Set of Enum values passed by the caller.

a) DefaultConfigurablePopulator

b) AbstractPopulatingConverter

c) AbstractConverter

Q = Product Populator are a little unique in that they typically extend a ------------------------------- class that is variant aware and supports the ability of falling back to a variants parent product for attribute values in the event of the source product value being null

a) DefaultProductPopulator

b) AbstractProductPopulatingConverter

c) AbstractProductPopulator d) None

Q = …………………extension is the template shipped with the "SAP Comm" Accelerator that you can use as a starting point for your own extension.

a) yacceleratorfacades b) commercefacades

c) both d) none

Q = Data Transfer Objects (DTOs) are objects created to contain….

a) Values b) Business logic

c) Both d) none of them

Q = ………..template method that allows a concrete sub-class to pick the appropriate target data object implementation.

a) createListTarget **b) createTarget**

c) createSourceToTarget d)None of the above

Q = Facades are which scoped Spring managed beans

a) yrequest **b) singleton** c) tenant d) request

Q = ………. Provides access to various internationalization switches that the user can make when they visit a specific storefront

a) Store Session façade b) Store Locator façade

c) User façade d) User Locale façade

Q = Explain different types of interceptors?

Model Interceptors = Intercept the behavior of the lifecycle of Models. Model lifecycle consists of loading from Database, saving to Database and deleting or removing from the Database.

In lifecycle of a model, interceptors can intercept & modify model data. It includes auditing, validating & even restricting the data from being removed/deleted from database if certain conditions are not met.

There are 5 types of Interceptors: -

a) LoadInterceptor = Invoked whenever a model is loaded from the database

b) InitDefaultsInterceptor = Invoked during modelService.create() & modelService.initDefaults()

c) PrepareInterceptor = Invoked before model is saved to database & before ValidateInterceptor

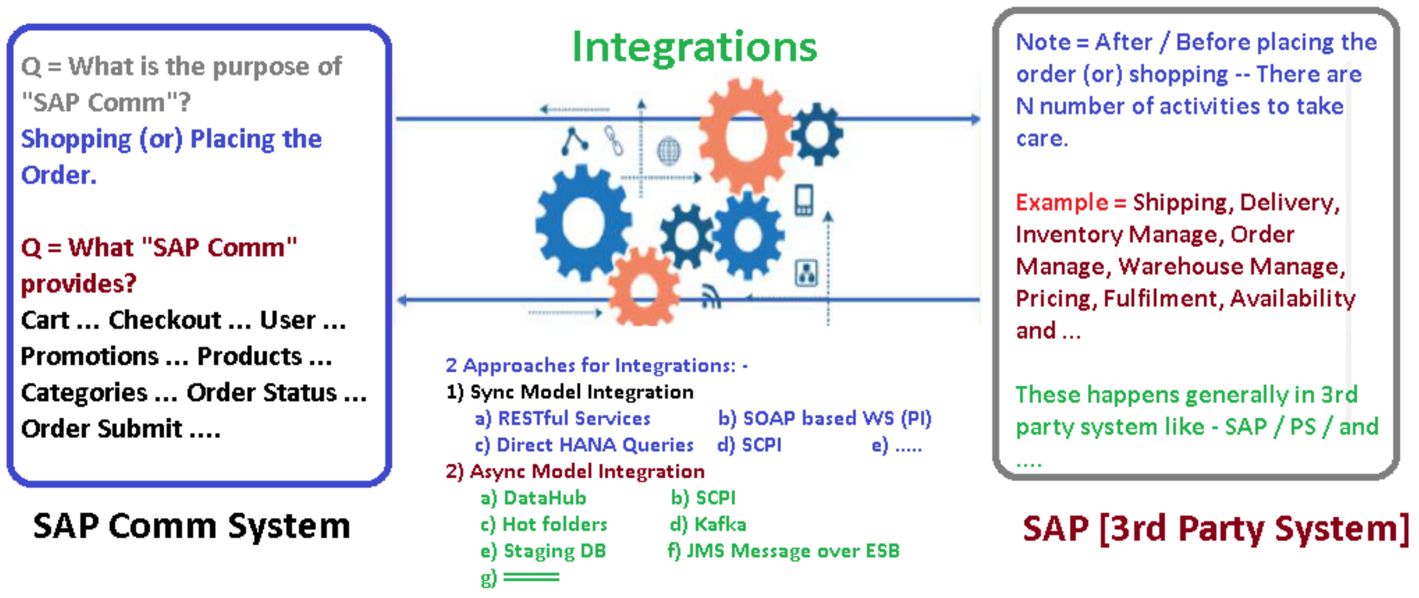
d) ValidateInterceptor = Invoked before a model is saved to database & after PrepareInterceptor

e) RemoveInterceptor = Invoked before a model is removed from database.

Note = If any Error like → javax.servlet.ServletException: File &quot;/WEB-INF/views/responsive/cms/assistedservicecomponent.jsp&quot; not found

Solution = ant addoninstall -Daddonnames="assistedservicestorefront" -DaddonStorefront.yacceleratorstorefront="**rrrstrainingstorefront**"

Integrations: -


Q = What is the purpose of "SAP Comm"?
Shopping (or) Placing the Order.

Q = What "SAP Comm" provides?
Cart ... Checkout ... User ... Promotions ... Products ... Categories ... Order Status ... Order Submit ....

SAP Comm System

Integrations

2 Approaches for Integrations: -
1) Sync Model Integration
   a) RESTful Services   b) SOAP based WS (PI)
   c) Direct HANA Queries   d) SCPI   e) .....
2) Async Model Integration
   a) DataHub   b) SCPI
   c) Hot folders   d) Kafka
   e) Staging DB   f) JMS Message over ESB
   g) ════

Note = After / Before placing the order (or) shopping -- There are N number of activities to take care.

Example = Shipping, Delivery, Inventory Manage, Order Manage, Warehouse Manage, Pricing, Fulfilment, Availability and ...

These happens generally in 3rd party system like - SAP / PS / and ....

SAP [3rd Party System]

Note = **Sync** Integration – We can do in N number of ways.

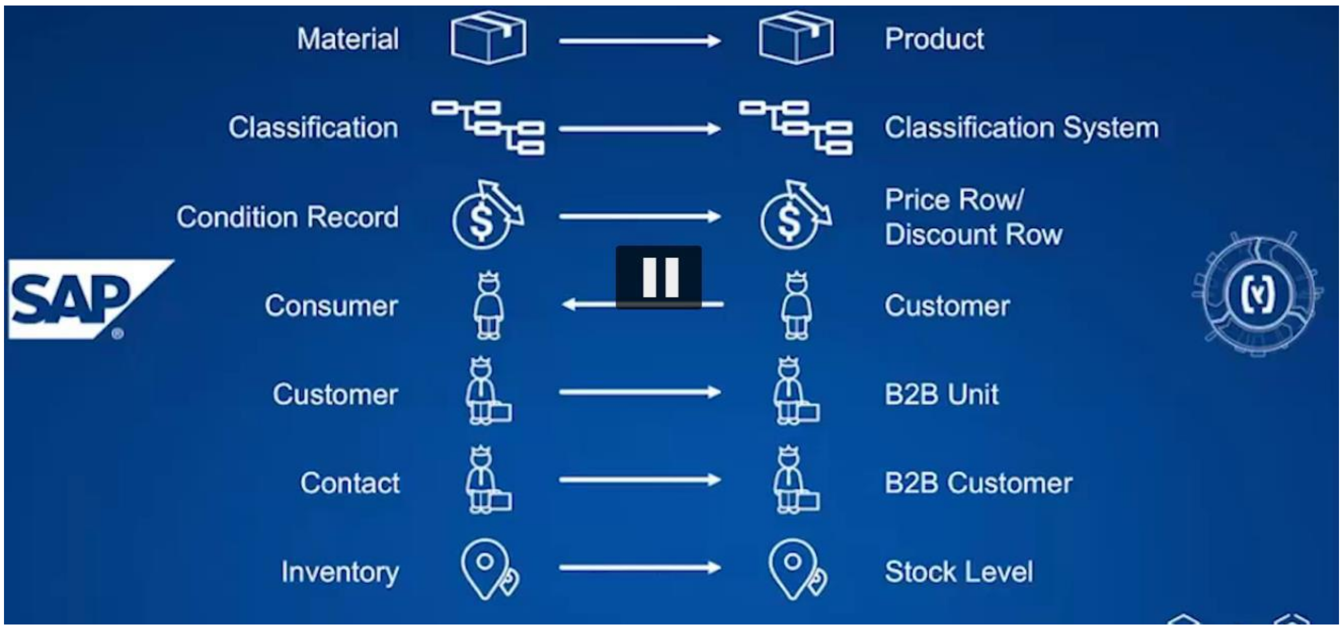
Also, **Async** integration – We can do in N number of ways.

Generally companies will do the **Hybrid** approach.

Hybrid approach =

Sync Integration for some scenario.

Async Integration for some scenario.



Q = Explain typical commerce application architecture Layers?

1. Front-end (i.e. the customer view or presentation layer)

2. Commerce API layer **3.** ERP

4. External single-purpose applications

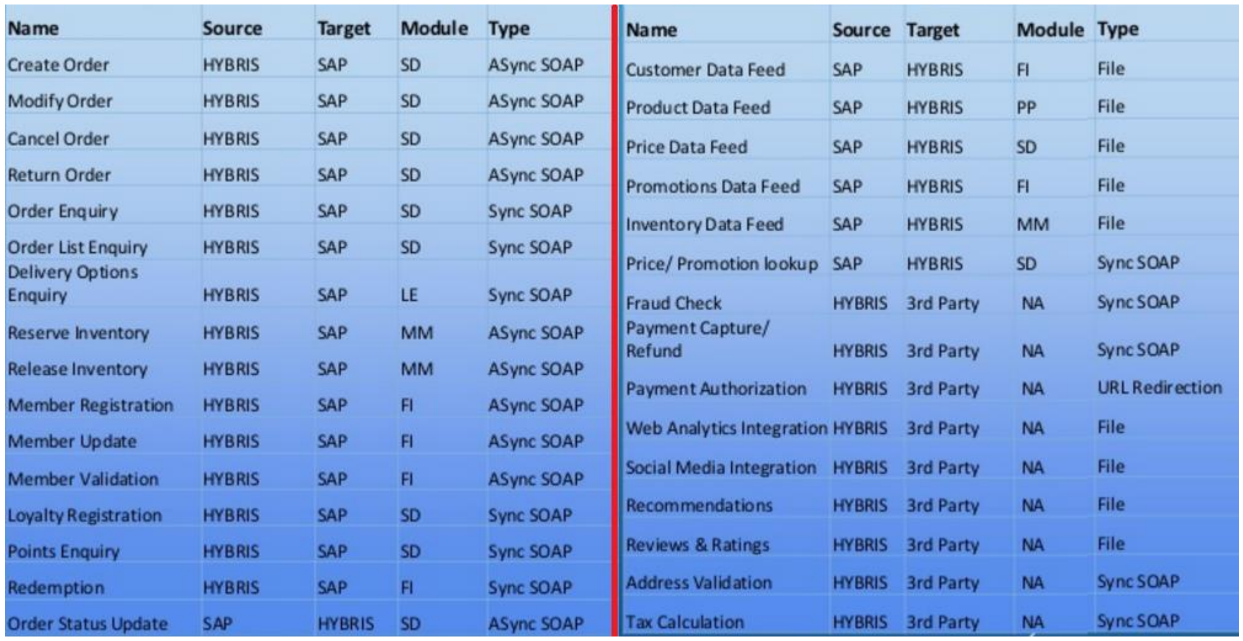
Note = Integration between the commerce layer and external, single-purpose applications allows commerce layer to easily display product, inventory, pricing, customer, and order data.

There are **2 approaches** for this (asynchronous and synchronous integration).

Q = How to Determine which method to use for specific data points is an important decision.

Data Point	Synchronous	Asynchronous
Product Data		Yes
Customer Data		Yes
Pricing Data	Yes	Yes
Inventory (Stock)	Yes	Yes
Order Status Data	Yes	Yes
Order Submission	Yes	Yes

Note: - So decision must of strategic & agreed to across all affected parts of the organization.



Name	Source	Target	Module	Type
Create Order	HYBRIS	SAP	SD	ASync SOAP
Modify Order	HYBRIS	SAP	SD	ASync SOAP
Cancel Order	HYBRIS	SAP	SD	ASync SOAP
Return Order	HYBRIS	SAP	SD	ASync SOAP
Order Enquiry	HYBRIS	SAP	SD	Sync SOAP
Order List Enquiry	HYBRIS	SAP	SD	Sync SOAP
Delivery Options Enquiry	HYBRIS	SAP	LE	Sync SOAP
Reserve Inventory	HYBRIS	SAP	MM	ASync SOAP
Release Inventory	HYBRIS	SAP	MM	ASync SOAP
Member Registration	HYBRIS	SAP	FI	ASync SOAP
Member Update	HYBRIS	SAP	FI	ASync SOAP
Member Validation	HYBRIS	SAP	FI	ASync SOAP
Loyalty Registration	HYBRIS	SAP	SD	Sync SOAP
Points Enquiry	HYBRIS	SAP	SD	Sync SOAP
Redemption	HYBRIS	SAP	FI	Sync SOAP
Order Status Update	SAP	HYBRIS	SD	ASync SOAP

Name	Source	Target	Module	Type
Customer Data Feed	SAP	HYBRIS	FI	File
Product Data Feed	SAP	HYBRIS	PP	File
Price Data Feed	SAP	HYBRIS	SD	File
Promotions Data Feed	SAP	HYBRIS	FI	File
Inventory Data Feed	SAP	HYBRIS	MM	File
Price/ Promotion lookup	SAP	HYBRIS	SD	Sync SOAP
Fraud Check	HYBRIS	3rd Party	NA	Sync SOAP
Payment Capture/ Refund	HYBRIS	3rd Party	NA	Sync SOAP
Payment Authorization	HYBRIS	3rd Party	NA	URL Redirection
Web Analytics Integration	HYBRIS	3rd Party	NA	File
Social Media Integration	HYBRIS	3rd Party	NA	File
Recommendations	HYBRIS	3rd Party	NA	File
Reviews & Ratings	HYBRIS	3rd Party	NA	File
Address Validation	HYBRIS	3rd Party	NA	Sync SOAP
Tax Calculation	HYBRIS	3rd Party	NA	Sync SOAP

Note = To simplify integration between SAP & “SAP Comm”, “SAP Comm” has built connectors, allows data to flow from SAP to “SAP Comm” through Data Hub. Data Hub leverages SAP standards IDOC format to transfer data from SAP to “SAP Comm”.

SAP IDOC	SAP COmm Item Type	Description
MATMAS	Product	Product Data
LOISTD	Stock Level	Stock / Inventory Information
CLSMAS	Category & Product	Classification Hierarchy
CLFMAS	Feature, Feature Value, Feature Assignment, Category	Classification Data
DEBMAS	Customer & B2BUnit	Customer Data
ADRMAS	B2BUnit & Address	Customer Address Data
ADR3MAS	B2BUnit & Address	Customer Address Data
COND_A04	Price Row & Discount Row	Customer-Specific Pricing (Price Condition)

Q: Explain Pricing Data? =

1 major difference between B2B & B2C is Price Data.

✓ In **B2C world**, base price is same for all customers. But in B2B world, customers have negotiated price data based on different business factors. Price conditions are typically written & developed in ERP & need to present to customer through online / offline. Hence,

✓ In B2C, it’s efficient to load prices into “SAP Comm” commerce platform (Bcoz Promotion management is stronger & flexible than ERP). In B2B, lot of time, money already spent for customizing ERP to make prices handled (custom pricing conditions).

✓ Now determining whether to use **Sync (or) Async** is based on: -

✓ **How often price change?** → If Org prices does not change often then go with Async Integration. It reduces load on ERP. Increase page load speed. Impacts customer experience.

✓ **How difficult to expert price from SAP to SAP Comm?** → If Org has complex pricing condition & can't easily exported from SAP then go with Sync Integration.

Q = Explain Order Status Updates?

✓ After order is placed, getting details of order is most common customer need.

✓ In B2C, customers are often obtaining tracking (Delivery Date).

✓ In B2B, customers also want additional information like Product allocated for shipment / Hold / …

✓ Product move through various Plants & Warehouses, they all have internal statuses saved in ERP / WMS.

✓ It's technology decision to determine best way to deliver status information from BackEnd to FrontEnd.

There are 2 options: -

- Real – Time Web Service Call from "SAP Comm" to SAP
- Batch data – load with replication of data from SAP - "SAP Comm".

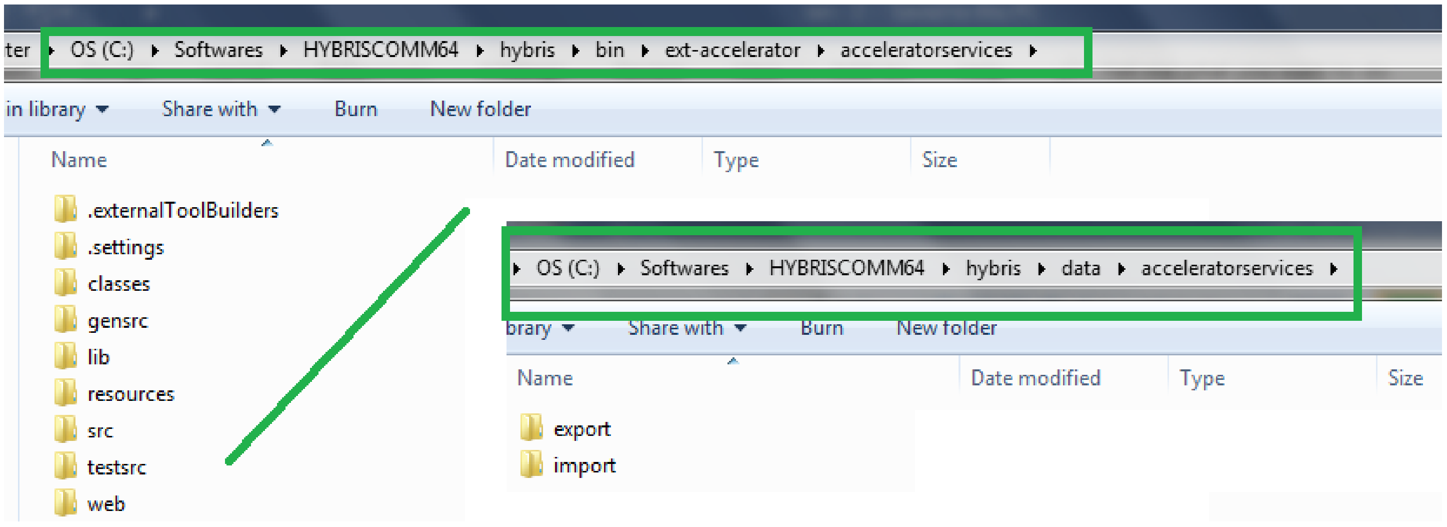
✓ The decision point (order status change) need to be notified (Push either email / mobile) to customer. If push notification is required then it is best to replicate data.

If push notification is not required then real – time web service should be used.

Explain Hot Folder

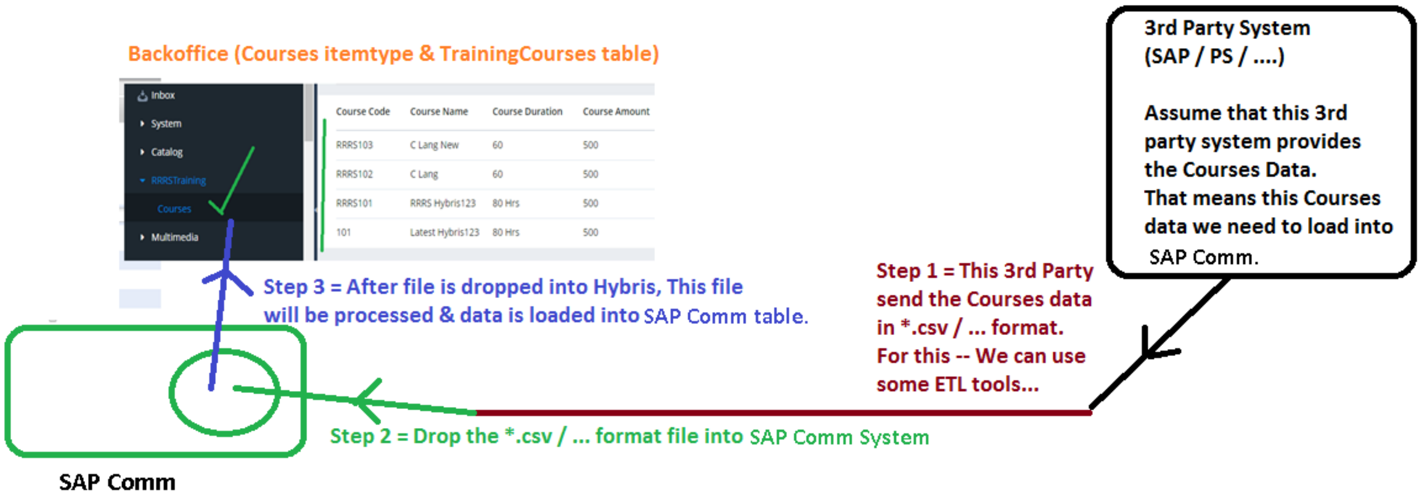
You have already seen how you can use ImpEx files to **import data** into the system. "SAP Comm" supports Hot Folders, which are folders from which data can be automatically imported into the platform by simply placing the data inside of the folder.

"SAP Comm" **acceleratorservices** extension template comes with a batch package that enables automated importing of data from hot folders.

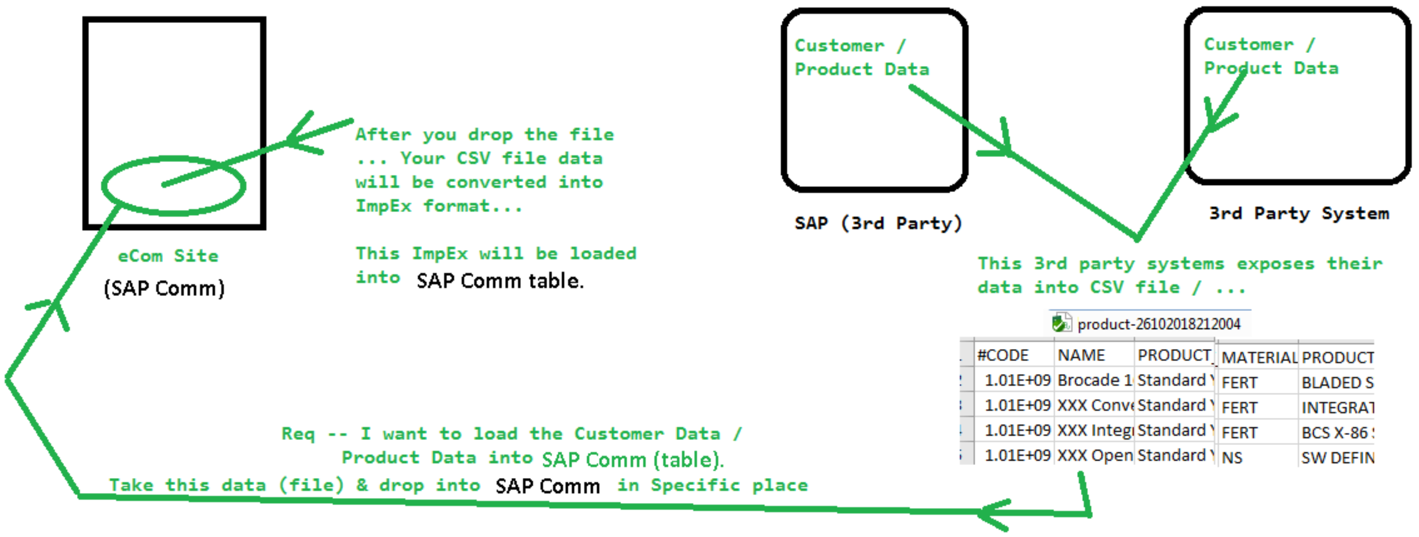

ter ▸ OS (C:) ▸ Softwares ▸ HYBRISCOMM64 ▸ hybris ▸ bin ▸ ext-accelerator ▸ acceleratorservices ▸
in library ▾ Share with ▾ Burn New folder
Name
Date modified
Type
Size
.externalToolBuilders
.settings
classes
gensrc
lib
resources
src
testsrc
web
▸ OS (C:) ▸ Softwares ▸ HYBRISCOMM64 ▸ hybris ▸ data ▸ acceleratorservices ▸
brary ▾ Share with ▾ Burn New folder
Name
Date modified
Type
Size
export
import

The infrastructure enables using simple CSV files (internally translated into "SAP Comm" ImpEx scripts) to import content directly into the product catalogs.

This infrastructure uses **Spring Integration** FWK. Spring Integration provides a pluggable, highly configurable service-based design that can be extended as required.

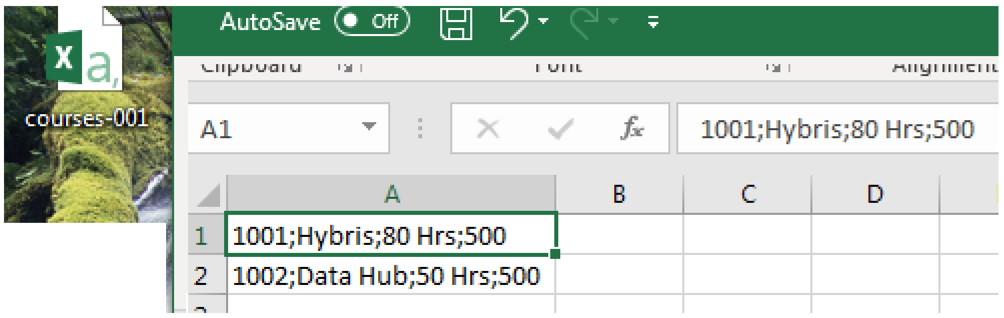

Backoffice (Courses itemtype & TrainingCourses table)
Inbox
System
Catalog
RRRSTraining
Courses
Multimedia
3rd Party System (SAP / PS / ....)
Assume that this 3rd party system provides the Courses Data. That means this Courses data we need to load into SAP Comm.
Step 1 = This 3rd Party send the Courses data in *.csv / ... format. For this -- We can use some ETL tools...
Step 2 = Drop the *.csv / ... format file into SAP Comm System
Step 3 = After file is dropped into Hybris, This file will be processed & data is loaded into SAP Comm table.
SAP Comm

Course Code	Course Name	Course Duration	Course Amount
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500
101	Latest Hybris123	80 Hrs	500

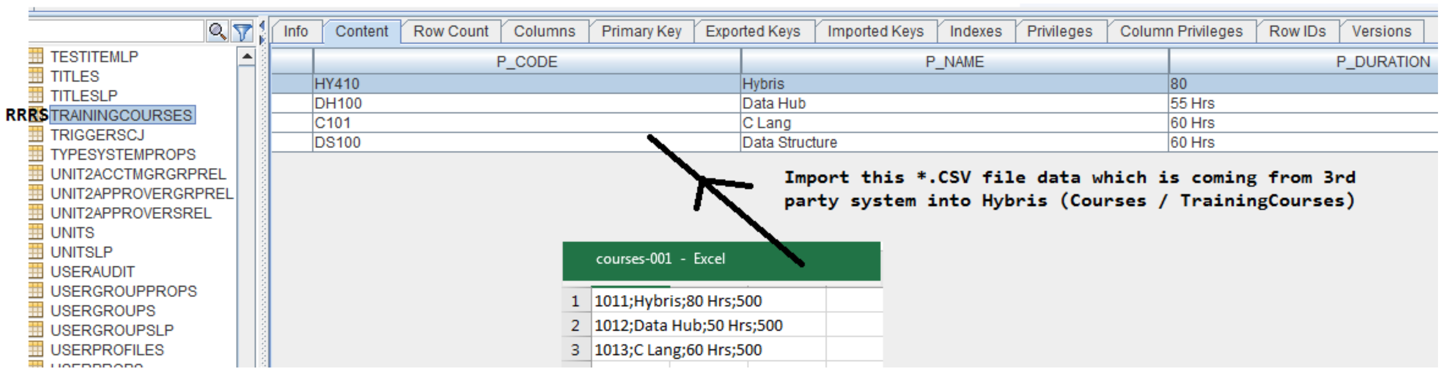

Customer / Product Data
SAP (3rd Party)
Customer / Product Data
3rd Party System
This 3rd party systems exposes their data into CSV file / ...
product-26102018212004
After you drop the file ... Your CSV file data will be converted into ImpEx format...
This ImpEx will be loaded into SAP Comm table.
eCom Site
(SAP Comm)
Req -- I want to load the Customer Data / Product Data into SAP Comm (table).
Take this data (file) & drop into SAP Comm in Specific place

#CODE	NAME	PRODUCT_	MATERIAL	PRODUCT
1.01E+09	Brocade 1	Standard Y	FERT	BLADED S
1.01E+09	XXX Conve	Standard Y	FERT	INTEGRAT
1.01E+09	XXX Integi	Standard Y	FERT	BCS X-86 S
1.01E+09	XXX Open	Standard Y	NS	SW DEFIN

Note: - Assume that, your 3rd party system giving below Courses.csv file.



	A
1	1001;Hybris;80 Hrs;500
2	1002;Data Hub;50 Hrs;500

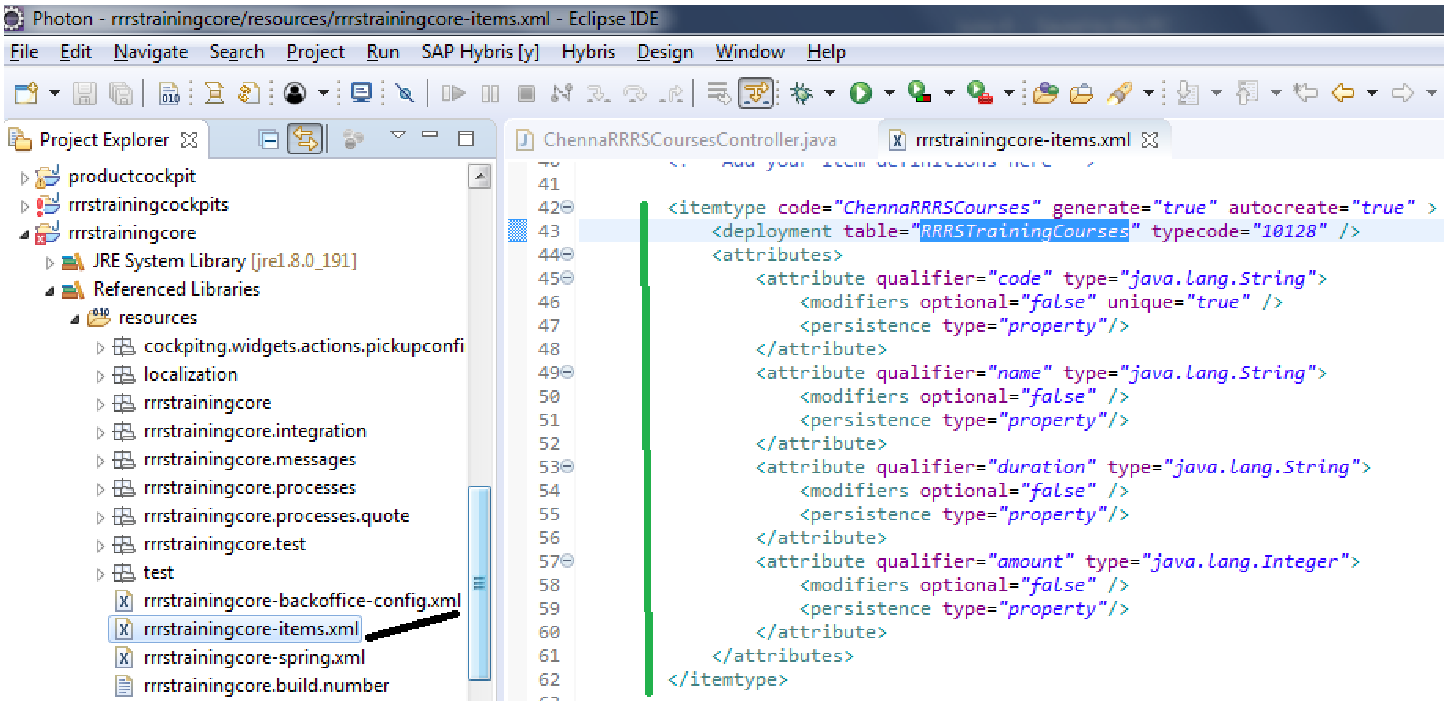


P_CODE	P_NAME	P_DURATION
HY410	Hybris	80
DH100	Data Hub	55 Hrs
C101	C Lang	60 Hrs
DS100	Data Structure	60 Hrs

Import this *.CSV file data which is coming from 3rd party system into Hybris (Courses / TrainingCourses)

```
1 1011;Hybris;80 Hrs;500
2 1012;Data Hub;50 Hrs;500
3 1013;C Lang;60 Hrs;500
```

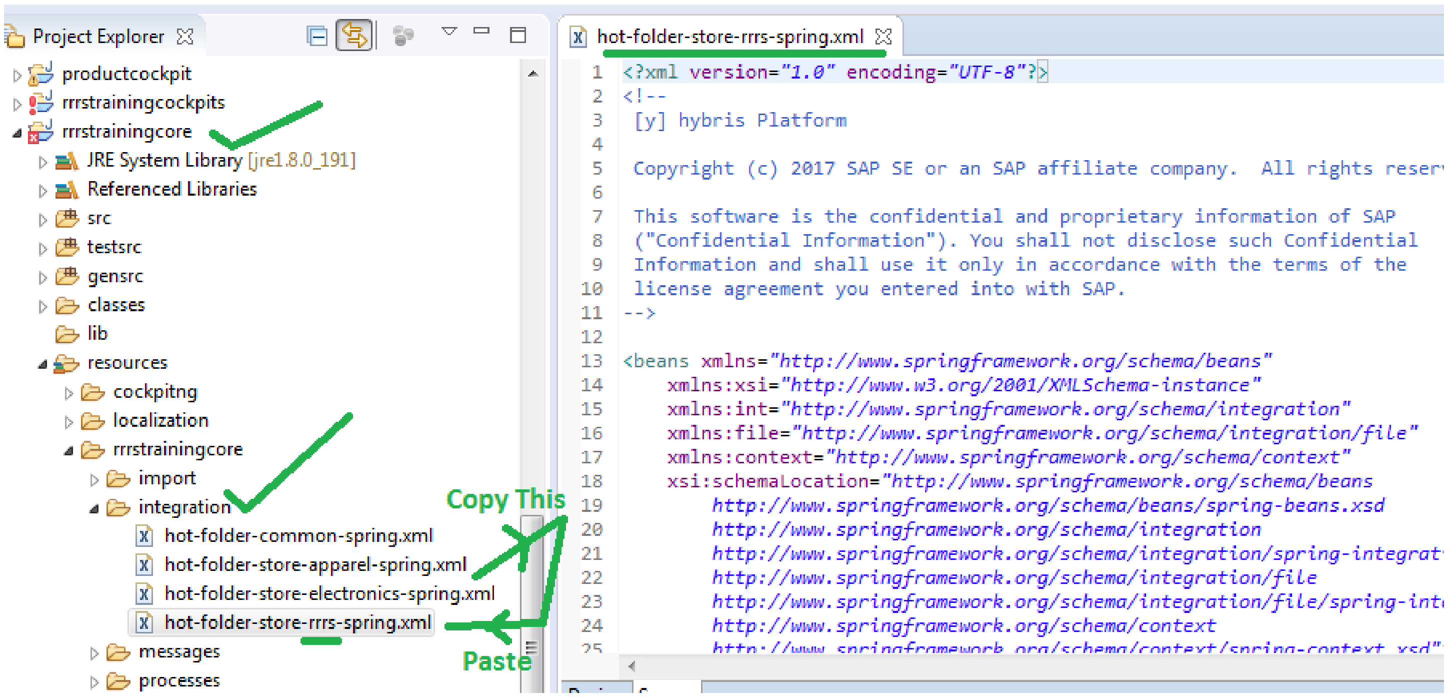
Step 1 = Create New Item Type called "**ChennaRRRSCourses**" & Table Name = "**RRRSTrainingCourses**"


Photon - rrrstrainingcore/resources/rrrstrainingcore-items.xml - Eclipse IDE
File Edit Navigate Search Project Run SAP Hybris [y] Hybris Design Window Help
Project Explorer
productcockpit
rrrstrainingcockpits
rrrstrainingcore
JRE System Library [jre1.8.0_191]
Referenced Libraries
resources
cockpitng.widgets.actions.pickupconfi
localization
rrrstrainingcore
rrrstrainingcore.integration
rrrstrainingcore.messages
rrrstrainingcore.processes
rrrstrainingcore.processes.quote
rrrstrainingcore.test
test
rrrstrainingcore-backoffice-config.xml
rrrstrainingcore-items.xml
rrrstrainingcore-spring.xml
rrrstrainingcore.build.number
ChennaRRRSCoursesController.java
rrrstrainingcore-items.xml

```xml
<itemtype code="ChennaRRRSCourses" generate="true" autocreate="true" >
    <deployment table="RRRSTrainingCourses" typecode="10128" />
    <attributes>
        <attribute qualifier="code" type="java.lang.String">
            <modifiers optional="false" unique="true" />
            <persistence type="property"/>
        </attribute>
        <attribute qualifier="name" type="java.lang.String">
            <modifiers optional="false" />
            <persistence type="property"/>
        </attribute>
        <attribute qualifier="duration" type="java.lang.String">
            <modifiers optional="false" />
            <persistence type="property"/>
        </attribute>
        <attribute qualifier="amount" type="java.lang.Integer">
            <modifiers optional="false" />
            <persistence type="property"/>
        </attribute>
    </attributes>
</itemtype>
```

Step 2 = Enable Standard hot folder for "**rrrstrainingstorefront**" & load the data into "**RRRSTrainingCourses**" table.

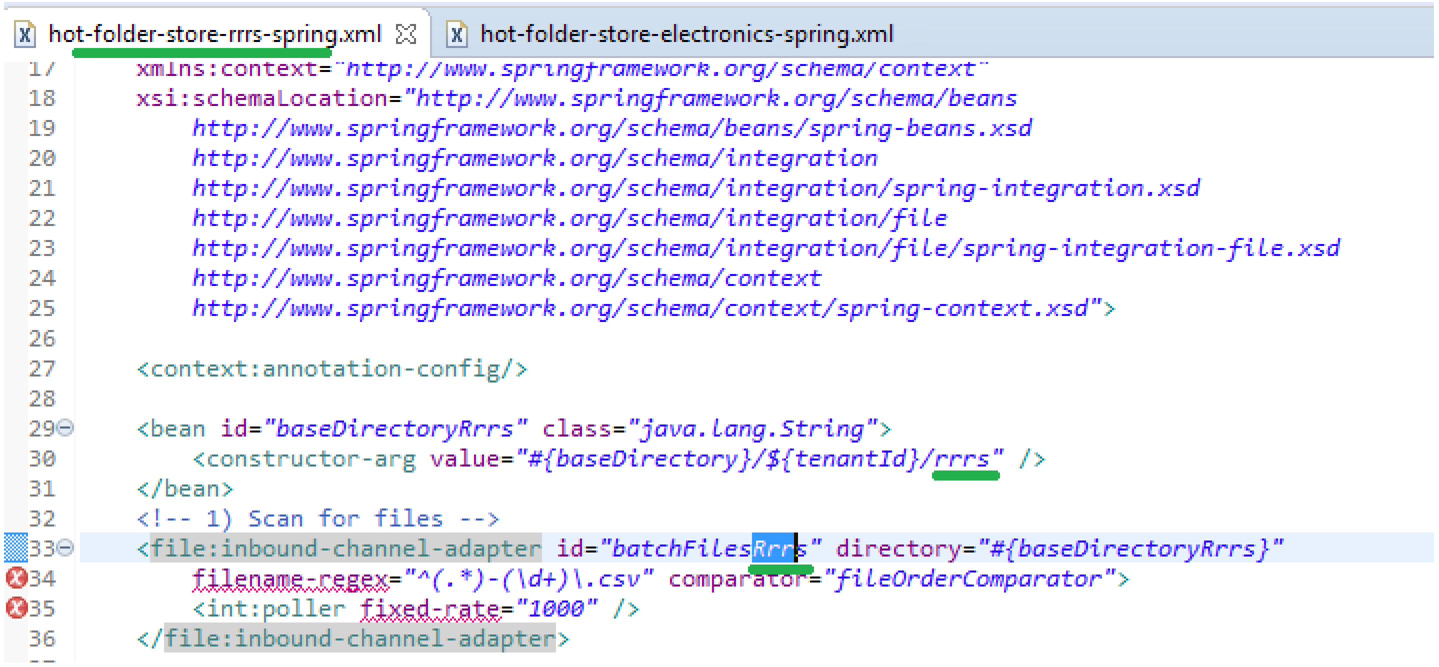
Copy "**hot-folder-store-electronics-spring.xml**" & Paste with different name "**hot-folder-store-rrrs-spring.xml**"



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
 [y] hybris Platform

 Copyright (c) 2017 SAP SE or an SAP affiliate company.  All rights reser

 This software is the confidential and proprietary information of SAP
 ("Confidential Information"). You shall not disclose such Confidential
 Information and shall use it only in accordance with the terms of the
 license agreement you entered into with SAP.
-->

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:int="http://www.springframework.org/schema/integration"
    xmlns:file="http://www.springframework.org/schema/integration/file"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/integration
        http://www.springframework.org/schema/integration/spring-integrat
        http://www.springframework.org/schema/integration/file
        http://www.springframework.org/schema/integration/file/spring-int
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd"
```

Step 3 = Open the file “hot-folder-store-rrrs-spring.xml” file & do below: -

1) Replace all occurrences of **electronics** with **rrrs**
2) Replace all occurrences of **Electronics** with **Rrrs**


hot-folder-store-rrrs-spring.xml
hot-folder-store-electronics-spring.xml

```
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/integration
        http://www.springframework.org/schema/integration/spring-integration.xsd
        http://www.springframework.org/schema/integration/file
        http://www.springframework.org/schema/integration/file/spring-integration-file.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd">

    <context:annotation-config/>

    <bean id="baseDirectoryRrrs" class="java.lang.String">
        <constructor-arg value="#{baseDirectory}/${tenantId}/rrrs" />
    </bean>
    <!-- 1) Scan for files -->
    <file:inbound-channel-adapter id="batchFilesRrrs" directory="#{baseDirectoryRrrs}"
        filename-regex="^(.*)-(\d+)\.csv" comparator="fileOrderComparator">
        <int:poller fixed-rate="1000" />
    </file:inbound-channel-adapter>
```

Q: What is there in “**hot-folder-store-rrrs-spring.xml**”?

1) File Dropping Loc

```
<bean id="baseDirectoryRrrs" class="java.lang.String">
    <constructor-arg value="#{baseDirectory}/${tenantId}/rrrs" />
</bean>
```

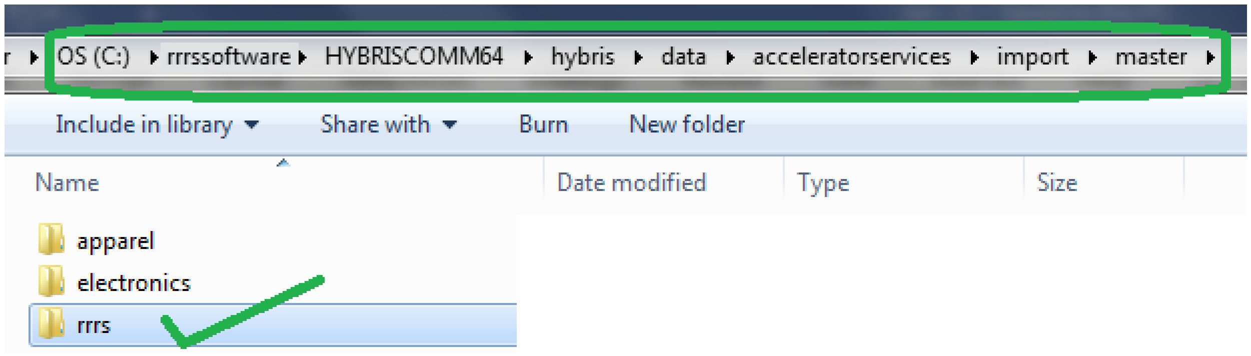
2) Inbound adapter = This scan for the matching files

```
<file:inbound-channel-adapter id="batchFilesRrrs" directory="#{baseDirectoryRrrs}"
    filename-regex="^(.*)-(\d+)\.csv" comparator="fileOrderComparator">
    <int:poller fixed-rate="1000" />
</file:inbound-channel-adapter>
```

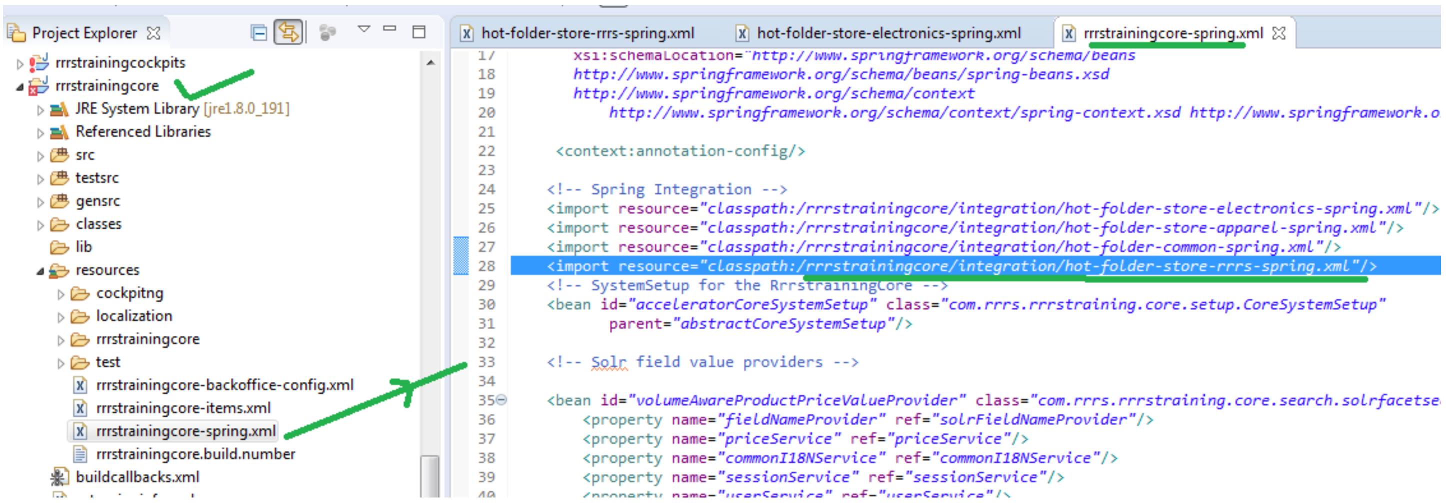
3) Outbound adapter = This takes the file & move for processing.

```
<file:outbound-gateway request-channel="batchFilesRrrs" reply-channel="batchFilesRrrsProc"
    directory="#{baseDirectoryRrrs}/processing" delete-source-files="true" />
```

Step 4 = Create “**rrrs**” folder.

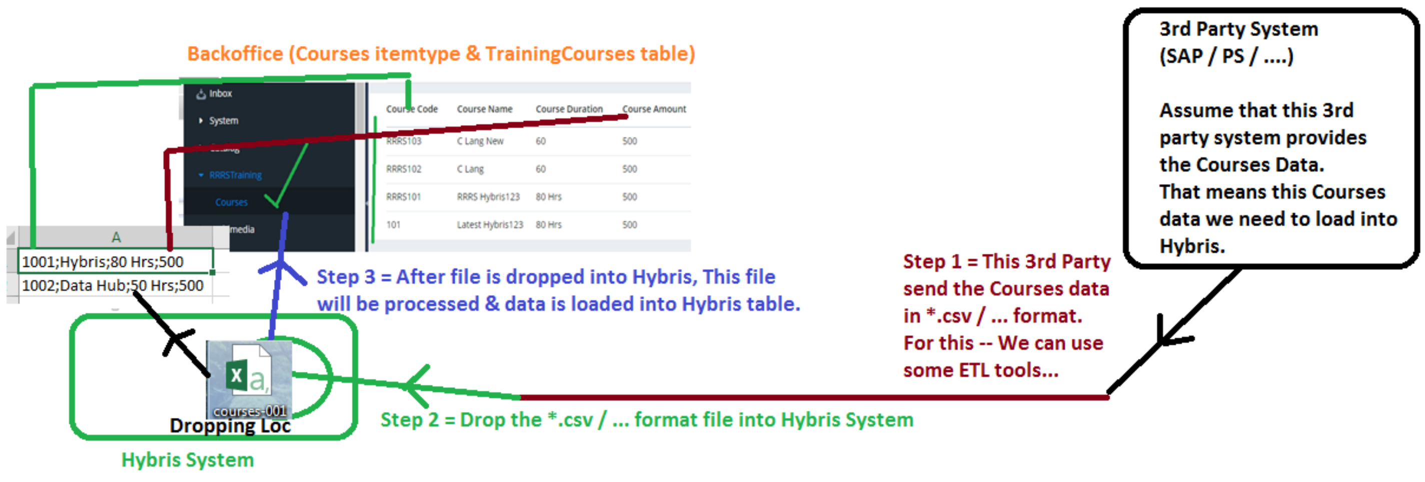

OS (C:) ▸ rrrssoftware ▸ HYBRISCOMM64 ▸ hybris ▸ data ▸ acceleratorservices ▸ import ▸ master ▸
Include in library
Share with
Burn
New folder
Name
Date modified
Type
Size
apparel
electronics
rrrs

Step 5 = Enable Spring Integration by adding "**hot-folder-store-rrrs-spring.xml**" file in "**rrrstrainingcore-spring.xml**"


Project Explorer
rrrstrainingcockpits
rrrstrainingcore
JRE System Library [jre1.8.0_191]
Referenced Libraries
src
testsrc
gensrc
classes
lib
resources
cockpitng
localization
rrrstrainingcore
test
rrrstrainingcore-backoffice-config.xml
rrrstrainingcore-items.xml
rrrstrainingcore-spring.xml
rrrstrainingcore.build.number
buildcallbacks.xml
hot-folder-store-rrrs-spring.xml
hot-folder-store-electronics-spring.xml
rrrstrainingcore-spring.xml

```
       xsi:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans.xsd
       http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.o

     <context:annotation-config/>

    <!-- Spring Integration -->
    <import resource="classpath:/rrrstrainingcore/integration/hot-folder-store-electronics-spring.xml"/>
    <import resource="classpath:/rrrstrainingcore/integration/hot-folder-store-apparel-spring.xml"/>
    <import resource="classpath:/rrrstrainingcore/integration/hot-folder-common-spring.xml"/>
    <import resource="classpath:/rrrstrainingcore/integration/hot-folder-store-rrrs-spring.xml"/>
    <!-- SystemSetup for the RrrstrainingCore -->
    <bean id="acceleratorCoreSystemSetup" class="com.rrrs.rrrstraining.core.setup.CoreSystemSetup"
           parent="abstractCoreSystemSetup"/>

    <!-- Solr field value providers -->

    <bean id="volumeAwareProductPriceValueProvider" class="com.rrrs.rrrstraining.core.search.solrfacetse
        <property name="fieldNameProvider" ref="solrFieldNameProvider"/>
        <property name="priceService" ref="priceService"/>
        <property name="commonI18NService" ref="commonI18NService"/>
        <property name="sessionService" ref="sessionService"/>
```

Step 6 = Create the mapping definition for **ChennaRRRSCourses** itemtype.



Course Code	Course Name	Course Duration	Course Amount
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500
101	Latest Hybris123	80 Hrs	500

```
1001;Hybris;80 Hrs;500
1002;Data Hub;50 Hrs;500
```

3rd Party System (SAP / PS / ….)

Assume that this 3rd party system provides the Courses Data. That means this Courses data we need to load into Hybris.

Step 1 = This 3rd Party send the Courses data in *.csv / … format. For this -- We can use some ETL tools…

Step 2 = Drop the *.csv / … format file into Hybris System

Step 3 = After file is dropped into Hybris, This file will be processed & data is loaded into Hybris table.

You dropped ***.csv** file (That means, Data is in CSV format).

"SAP Comm" knows only ImpEx.

So – It's time to covert ***.csv** data into equivalent **ImpEx**.

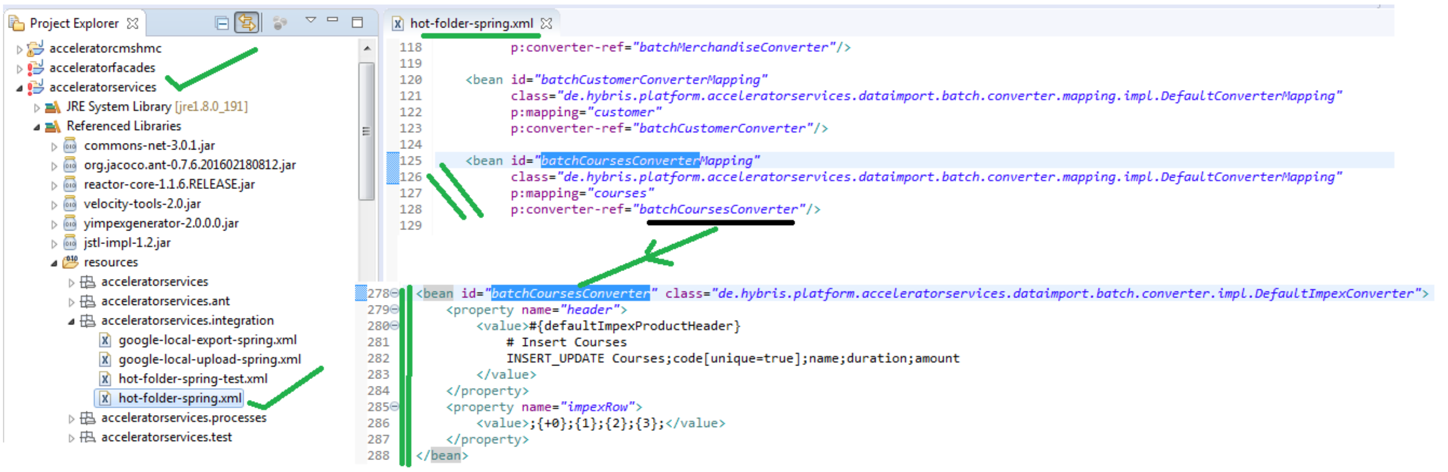
Do the Mapping like: -

Col A data should go to “Course **Code**”

Col B data should go to “Course **Name**”

Col C data should go to “Course **Duration**”

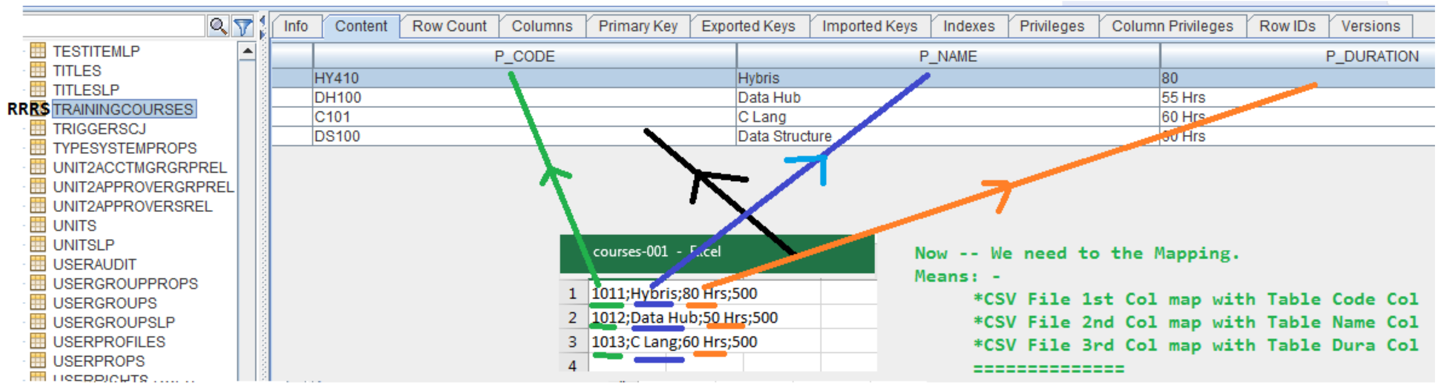
Col D data should go to “Course **Amount**”.


Project Explorer
acceleratorcmshmc
acceleratorfacades
acceleratorservices
JRE System Library [jre1.8.0_191]
Referenced Libraries
commons-net-3.0.1.jar
org.jacoco.ant-0.7.6.201602180812.jar
reactor-core-1.1.6.RELEASE.jar
velocity-tools-2.0.jar
yimpexgenerator-2.0.0.0.jar
jstl-impl-1.2.jar
resources
acceleratorservices
acceleratorservices.ant
acceleratorservices.integration
google-local-export-spring.xml
google-local-upload-spring.xml
hot-folder-spring-test.xml
hot-folder-spring.xml
acceleratorservices.processes
acceleratorservices.test
hot-folder-spring.xml

```xml
            p:converter-ref="batchMerchandiseConverter"/>

    <bean id="batchCustomerConverterMapping"
            class="de.hybris.platform.acceleratorservices.dataimport.batch.converter.mapping.impl.DefaultConverterMapping"
            p:mapping="customer"
            p:converter-ref="batchCustomerConverter"/>

    <bean id="batchCoursesConverterMapping"
            class="de.hybris.platform.acceleratorservices.dataimport.batch.converter.mapping.impl.DefaultConverterMapping"
            p:mapping="courses"
            p:converter-ref="batchCoursesConverter"/>
```

```xml
<bean id="batchCoursesConverter" class="de.hybris.platform.acceleratorservices.dataimport.batch.converter.impl.DefaultImpexConverter">
    <property name="header">
        <value>#{defaultImpexProductHeader}
            # Insert Courses
            INSERT_UPDATE Courses;code[unique=true];name;duration;amount
        </value>
    </property>
    <property name="impexRow">
        <value>;{+0};{1};{2};{3};</value>
    </property>
</bean>
```

Note: - {+0} = Here + represents required field.



P_CODE	P_NAME	P_DURATION
HY410	Hybris	80
DH100	Data Hub	55 Hrs
C101	C Lang	60 Hrs
DS100	Data Structure	[illegible]0 Hrs

1	1011;Hybris;80 Hrs;500
2	1012;Data Hub;50 Hrs;500
3	1013;C Lang;60 Hrs;500
4	

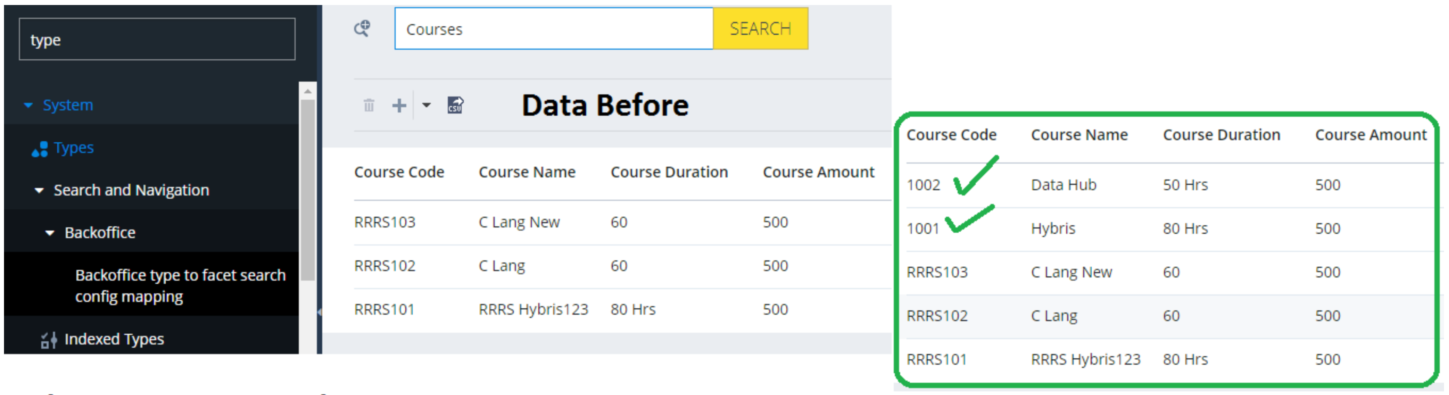
```
Now -- We need to the Mapping.
Means: -
        *CSV File 1st Col map with Table Code Col
        *CSV File 2nd Col map with Table Name Col
        *CSV File 3rd Col map with Table Dura Col
        ==============
```

Contact Us = ChennaReddyTraining@RRRS.CO.IN

Step 6 = Do the build (**ant clean all**)

Step 7 = Start the Server (**hybrisserver.bat**)

Step 8 = Test the Results



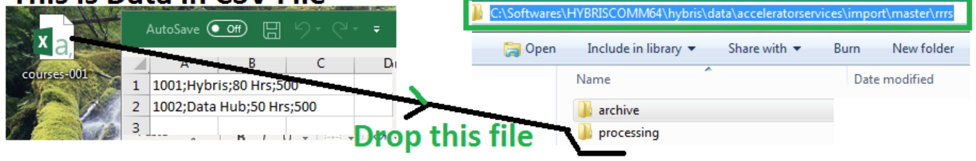
Data Before

Course Code	Course Name	Course Duration	Course Amount
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500

Course Code	Course Name	Course Duration	Course Amount
1002 ✓	Data Hub	50 Hrs	500
1001 ✓	Hybris	80 Hrs	500
RRRS103	C Lang New	60	500
RRRS102	C Lang	60	500
RRRS101	RRRS Hybris123	80 Hrs	500

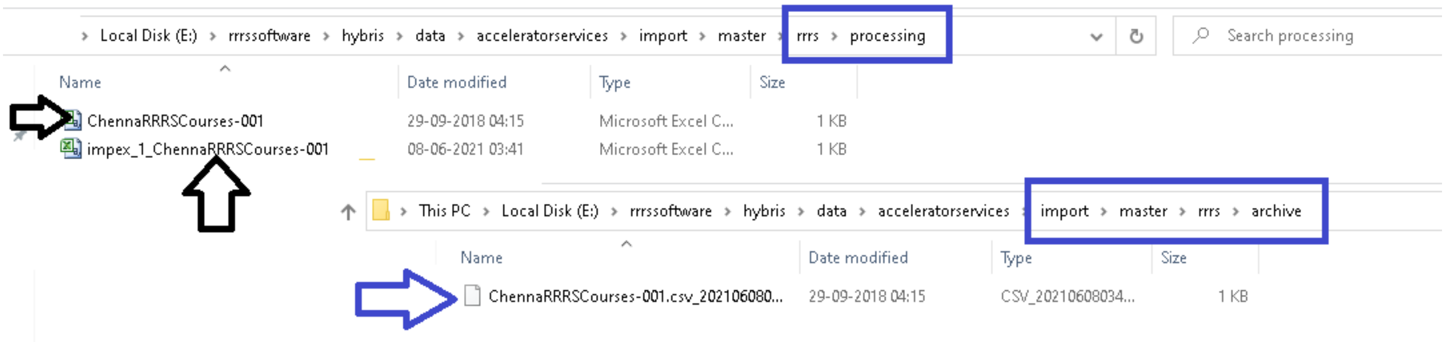
Results (Output)

This is Data in CSV File



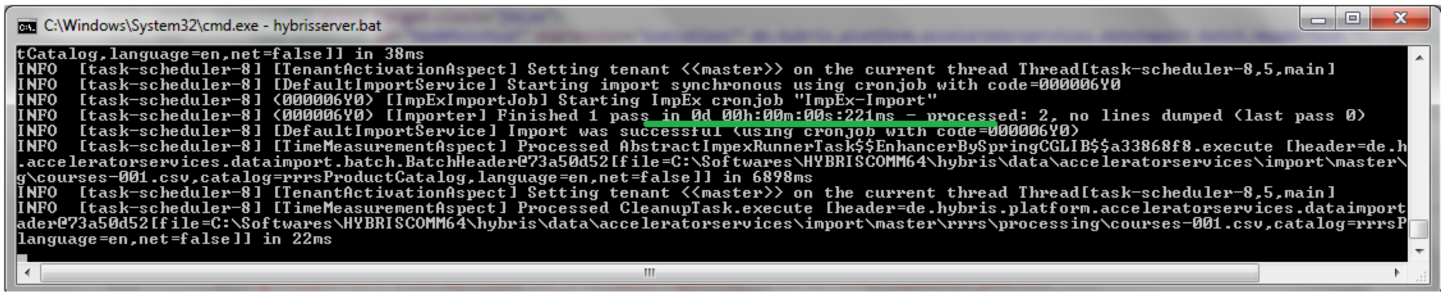
	A	B	C	D
1	1001;Hybris;80 Hrs;500			
2	1002;Data Hub;50 Hrs;500			
3				

Drop this file


Local Disk (E:) › rrrssoftware › hybris › data › acceleratorservices › import › master › rrrs › processing
Search processing
This PC › Local Disk (E:) › rrrssoftware › hybris › data › acceleratorservices › import › master › rrrs › archive

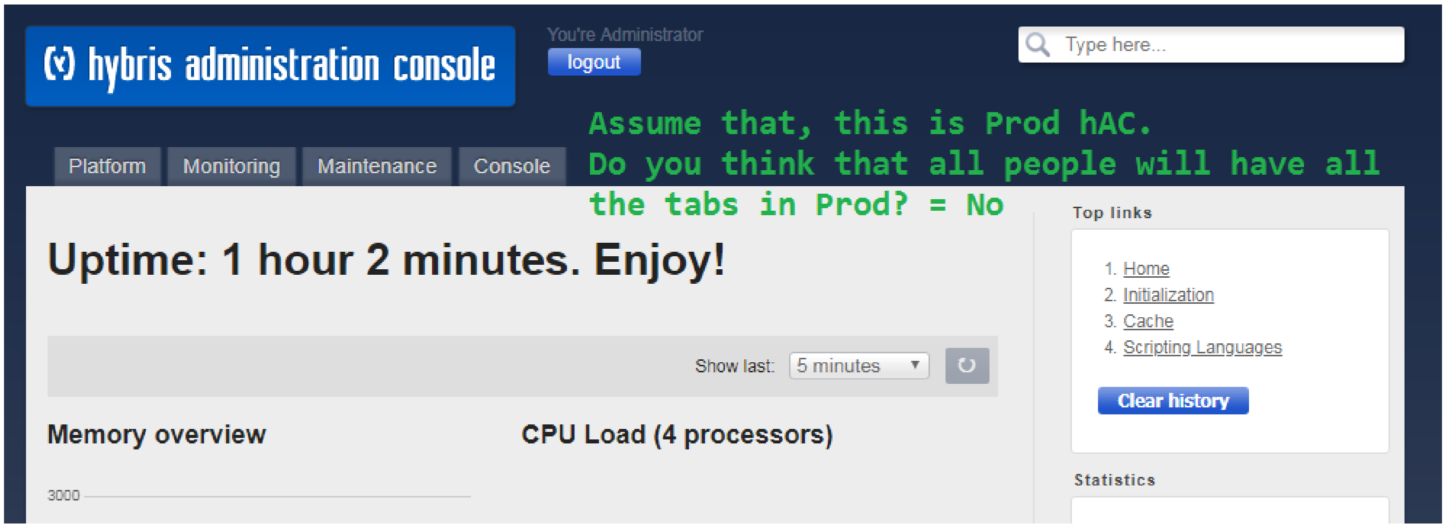
Name	Date modified	Type	Size
ChennaRRRSCourses-001	29-09-2018 04:15	Microsoft Excel C...	1 KB
impex_1_ChennaRRRSCourses-001	08-06-2021 03:41	Microsoft Excel C...	1 KB

Name	Date modified	Type	Size
ChennaRRRSCourses-001.csv_202106080...	29-09-2018 04:15	CSV_20210608034...	1 KB


C:\Windows\System32\cmd.exe - hybrisserver.bat

```
tCatalog,language=en,net=false]] in 38ms
INFO  [task-scheduler-8] [TenantActivationAspect] Setting tenant <<master>> on the current thread Thread[task-scheduler-8,5,main]
INFO  [task-scheduler-8] [DefaultImportService] Starting import synchronous using cronjob with code=000006Y0
INFO  [task-scheduler-8] (000006Y0) [ImpExImportJob] Starting ImpEx cronjob "ImpEx-Import"
INFO  [task-scheduler-8] (000006Y0) [Importer] Finished 1 pass in 0d 00h:00m:00s:221ms - processed: 2, no lines dumped (last pass 0)
INFO  [task-scheduler-8] [DefaultImportService] Import was successful (using cronjob with code=000006Y0)
INFO  [task-scheduler-8] [TimeMeasurementAspect] Processed AbstractImpexRunnerTask$$EnhancerBySpringCGLIB$$a33868f8.execute [header=de.h
.acceleratorservices.dataimport.batch.BatchHeader@73a50d52[file=C:\Softwares\HYBRISCOMM64\hybris\data\acceleratorservices\import\master\
g\courses-001.csv,catalog=rrrsProductCatalog,language=en,net=false]] in 6898ms
INFO  [task-scheduler-8] [TenantActivationAspect] Setting tenant <<master>> on the current thread Thread[task-scheduler-8,5,main]
INFO  [task-scheduler-8] [TimeMeasurementAspect] Processed CleanupTask.execute [header=de.hybris.platform.acceleratorservices.dataimport
ader@73a50d52[file=C:\Softwares\HYBRISCOMM64\hybris\data\acceleratorservices\import\master\rrrs\processing\courses-001.csv,catalog=rrrsP
language=en,net=false]] in 22ms
```

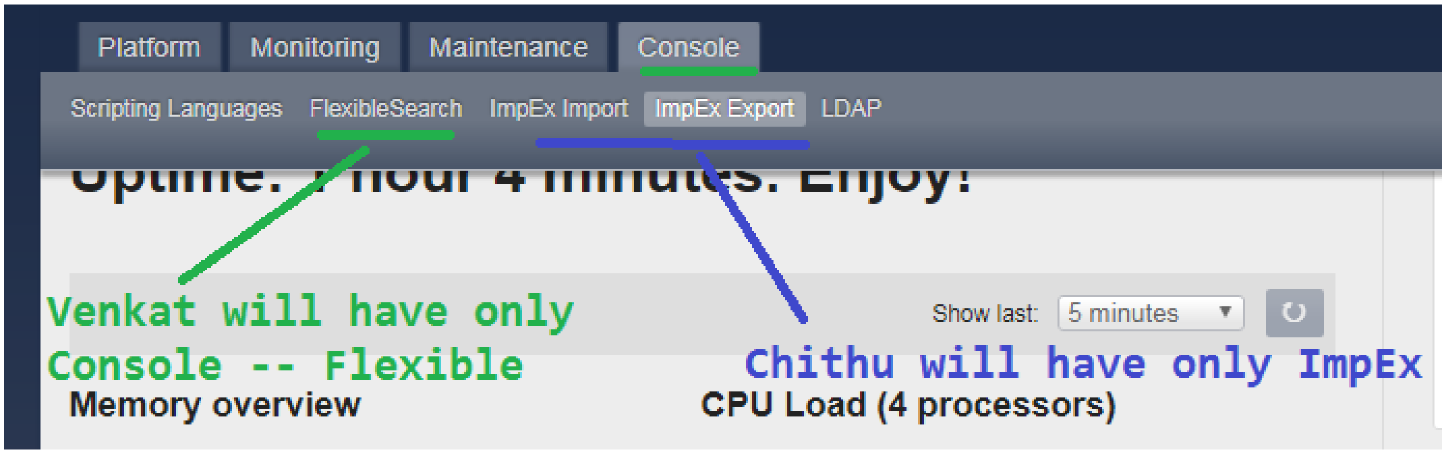
"SAP Comm" / Commerce Security & Roles Configuration



Uptime: 1 hour 2 minutes. Enjoy!

Assume that, this is Prod hAC.
Do you think that all people will have all the tabs in Prod? = No

So – It's time to control the Tabs & Also Options within the tab.



Venkat will have only Console -- Flexible

Chithu will have only ImpEx

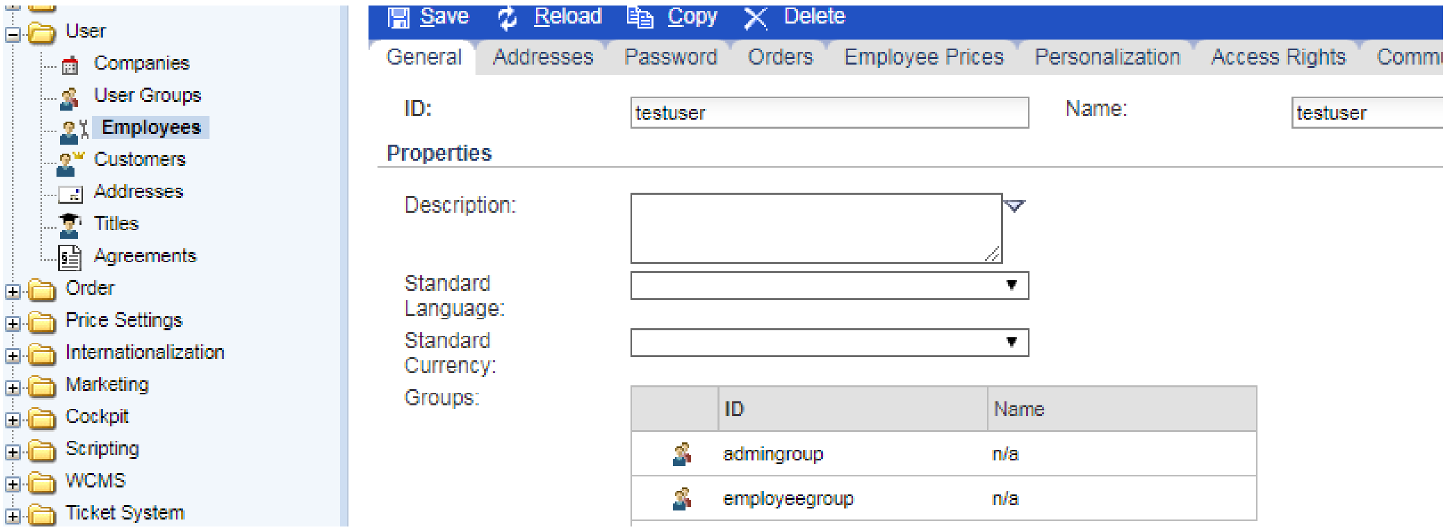
"SAP Comm" Security → based on Spring Security.

Example: - Create **1 User** & that user should have only "Console tab" & 2 Options (Scripting Language & Flexible Search).

Backoffice / hMC – Users – Employee –

ID = testuser (or) chenna

Step 1 = Let’s create user with admin group.



ID: testuser
Name: testuser

Properties

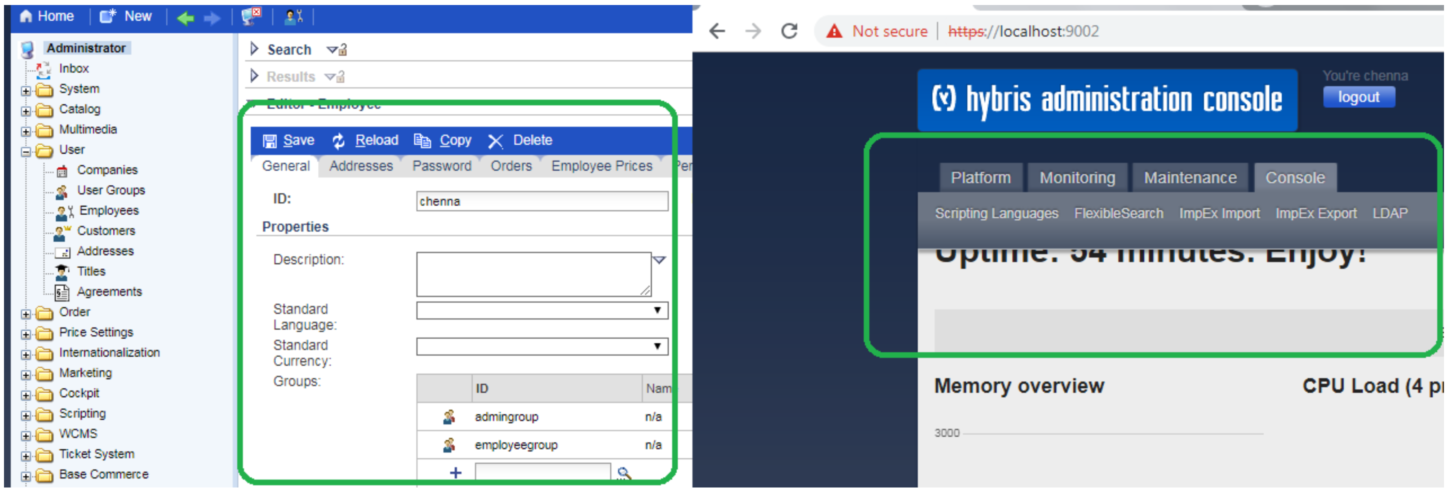
Description:

Standard Language:

Standard Currency:

Groups:

ID	Name
admingroup	n/a
employeegroup	n/a



ID: chenna

Properties

Description:

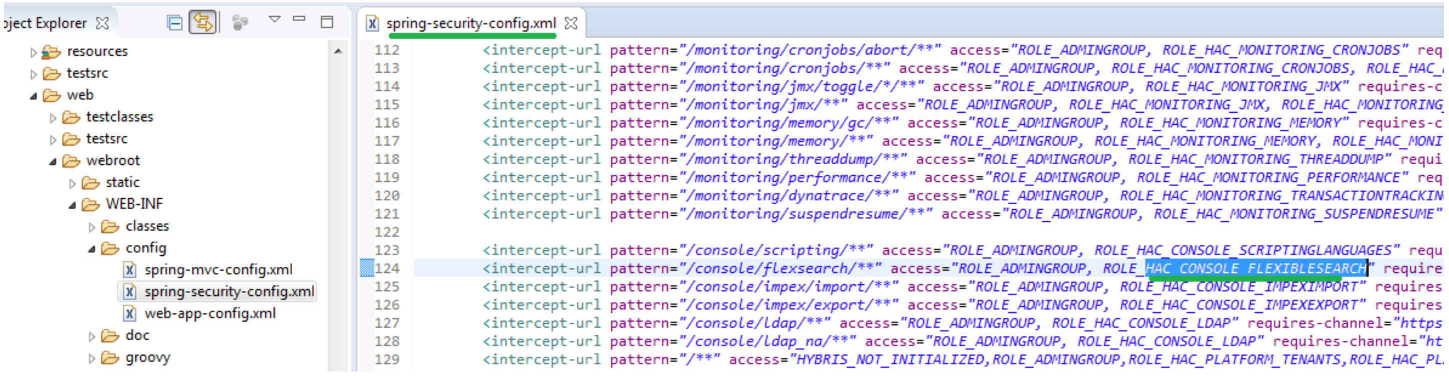
Standard Language:

Standard Currency:

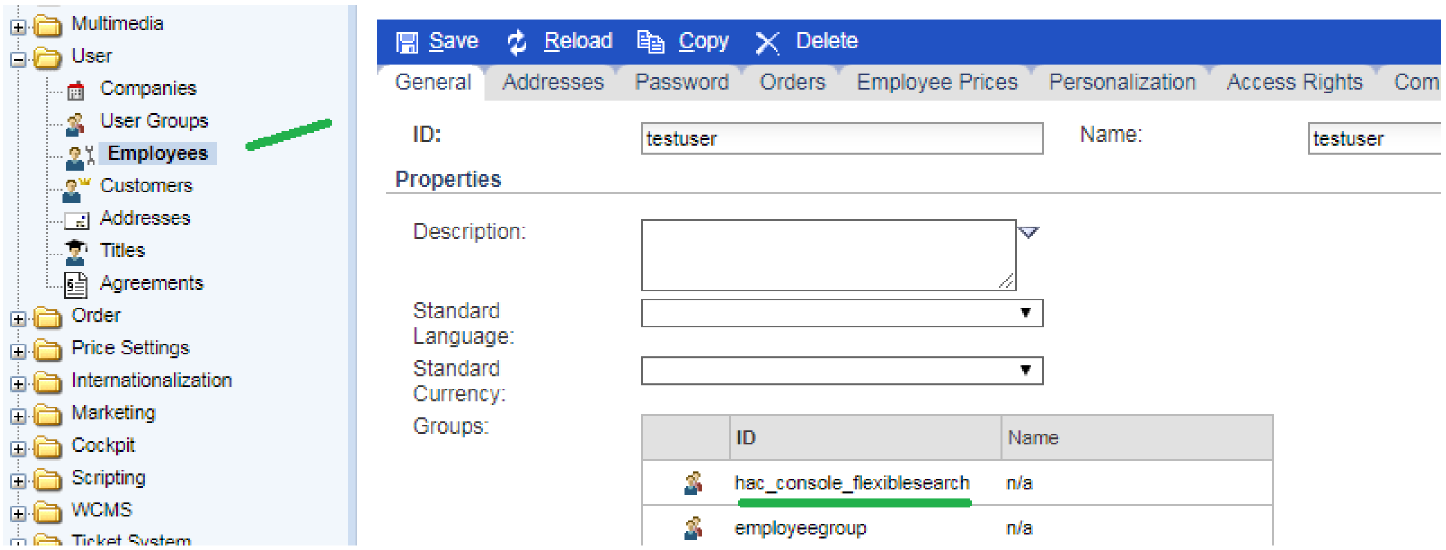
Groups:

ID	Name
admingroup	n/a
employeegroup	n/a

Q: How to give only “Console – FlexibleSearch” for created user (**testuser**).



```
        <intercept-url pattern="/monitoring/cronjobs/abort/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_CRONJOBS" req
        <intercept-url pattern="/monitoring/cronjobs/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_CRONJOBS, ROLE_HAC_
        <intercept-url pattern="/monitoring/jmx/toggle/*/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_JMX" requires-c
        <intercept-url pattern="/monitoring/jmx/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_JMX, ROLE_HAC_MONITORING
        <intercept-url pattern="/monitoring/memory/gc/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_MEMORY" requires-c
        <intercept-url pattern="/monitoring/memory/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_MEMORY, ROLE_HAC_MONI
        <intercept-url pattern="/monitoring/threaddump/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_THREADDUMP" requi
        <intercept-url pattern="/monitoring/performance/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_PERFORMANCE" req
        <intercept-url pattern="/monitoring/dynatrace/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_TRANSACTIONTRACKIN
        <intercept-url pattern="/monitoring/suspendresume/**" access="ROLE_ADMINGROUP, ROLE_HAC_MONITORING_SUSPENDRESUME"

        <intercept-url pattern="/console/scripting/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_SCRIPTINGLANGUAGES" requ
        <intercept-url pattern="/console/flexsearch/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_FLEXIBLESEARCH" require
        <intercept-url pattern="/console/impex/import/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_IMPEXIMPORT" requires
        <intercept-url pattern="/console/impex/export/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_IMPEXEXPORT" requires
        <intercept-url pattern="/console/ldap/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_LDAP" requires-channel="https
        <intercept-url pattern="/console/ldap_na/**" access="ROLE_ADMINGROUP, ROLE_HAC_CONSOLE_LDAP" requires-channel="ht
        <intercept-url pattern="/**" access="HYBRIS_NOT_INITIALIZED,ROLE_ADMINGROUP,ROLE_HAC_PLATFORM_TENANTS,ROLE_HAC_PL
```

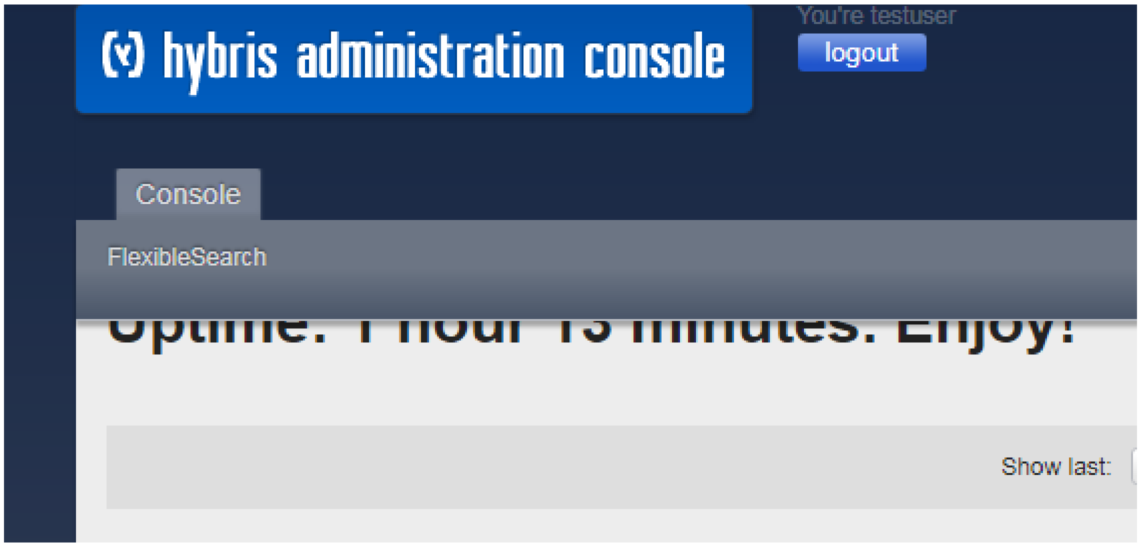


ID:	testuser
Name:	testuser

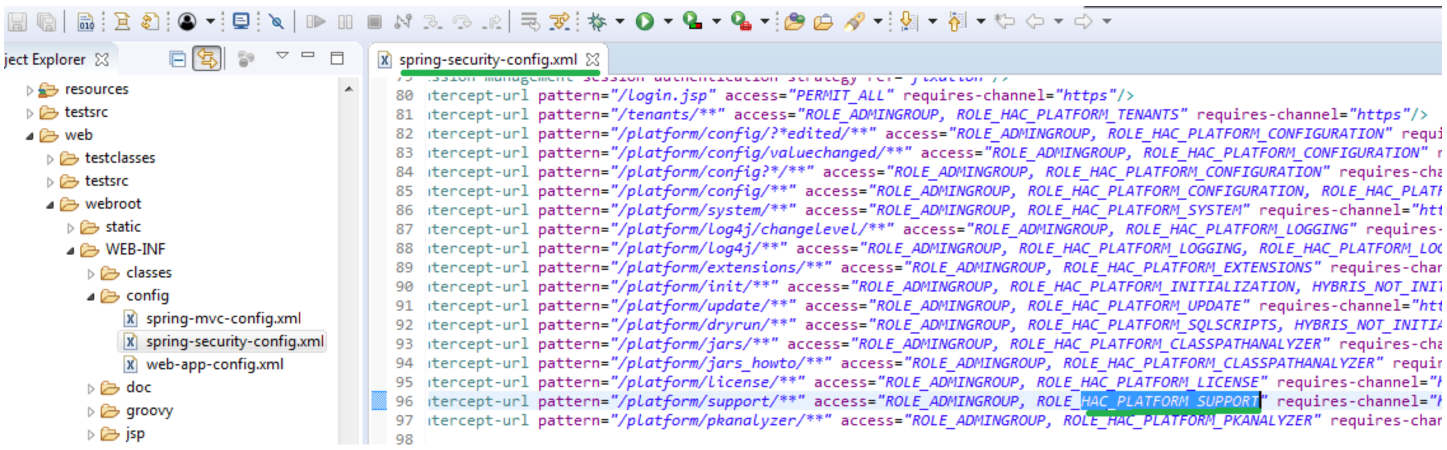
Properties

Description:	
Standard Language:	
Standard Currency:	
Groups:	

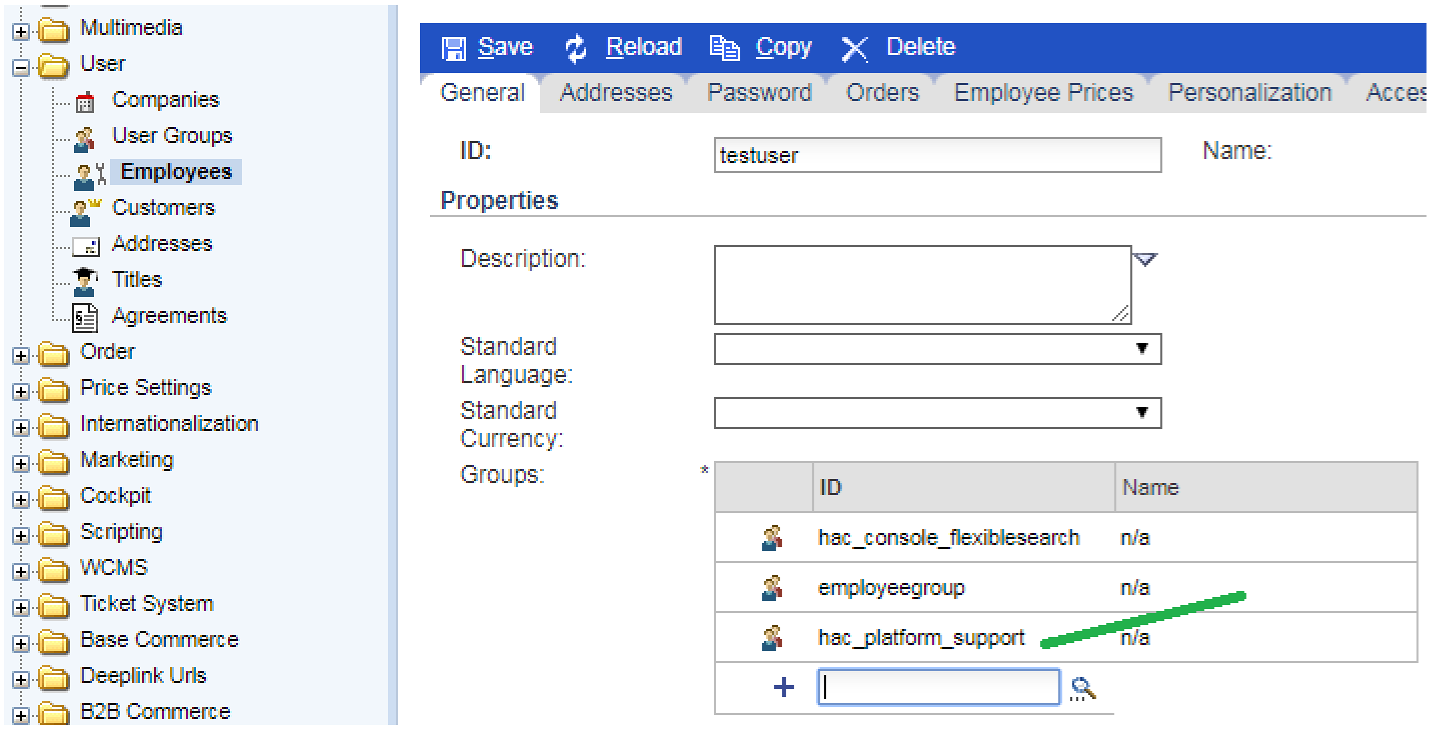
	ID	Name
	hac_console_flexiblesearch	n/a
	employeegroup	n/a


hybris administration console
You're testuser
logout
Console
FlexibleSearch
Uptime: 1 hour 13 minutes. Enjoy!
Show last:

Q: How to provide **log** access (or) download log to created user (**testuser**).



```
ıtercept-url pattern="/login.jsp" access="PERMIT_ALL" requires-channel="https"/>
ıtercept-url pattern="/tenants/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_TENANTS" requires-channel="https"/>
ıtercept-url pattern="/platform/config/?*edited/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CONFIGURATION" requi
ıtercept-url pattern="/platform/config/valuechanged/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CONFIGURATION" r
ıtercept-url pattern="/platform/config?*/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CONFIGURATION" requires-cha
ıtercept-url pattern="/platform/config/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CONFIGURATION, ROLE_HAC_PLATF
ıtercept-url pattern="/platform/system/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_SYSTEM" requires-channel="htt
ıtercept-url pattern="/platform/log4j/changelevel/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_LOGGING" requires-
ıtercept-url pattern="/platform/log4j/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_LOGGING, ROLE_HAC_PLATFORM_LOG
ıtercept-url pattern="/platform/extensions/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_EXTENSIONS" requires-char
ıtercept-url pattern="/platform/init/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_INITIALIZATION, HYBRIS_NOT_INIT
ıtercept-url pattern="/platform/update/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_UPDATE" requires-channel="htt
ıtercept-url pattern="/platform/dryrun/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_SQLSCRIPTS, HYBRIS_NOT_INITIA
ıtercept-url pattern="/platform/jars/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CLASSPATHANALYZER" requires-cha
ıtercept-url pattern="/platform/jars_howto/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_CLASSPATHANALYZER" requir
ıtercept-url pattern="/platform/license/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_LICENSE" requires-channel="h
ıtercept-url pattern="/platform/support/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_SUPPORT" requires-channel="h
ıtercept-url pattern="/platform/pkanalyzer/**" access="ROLE_ADMINGROUP, ROLE_HAC_PLATFORM_PKANALYZER" requires-char
```



ID: testuser

Name:

Properties

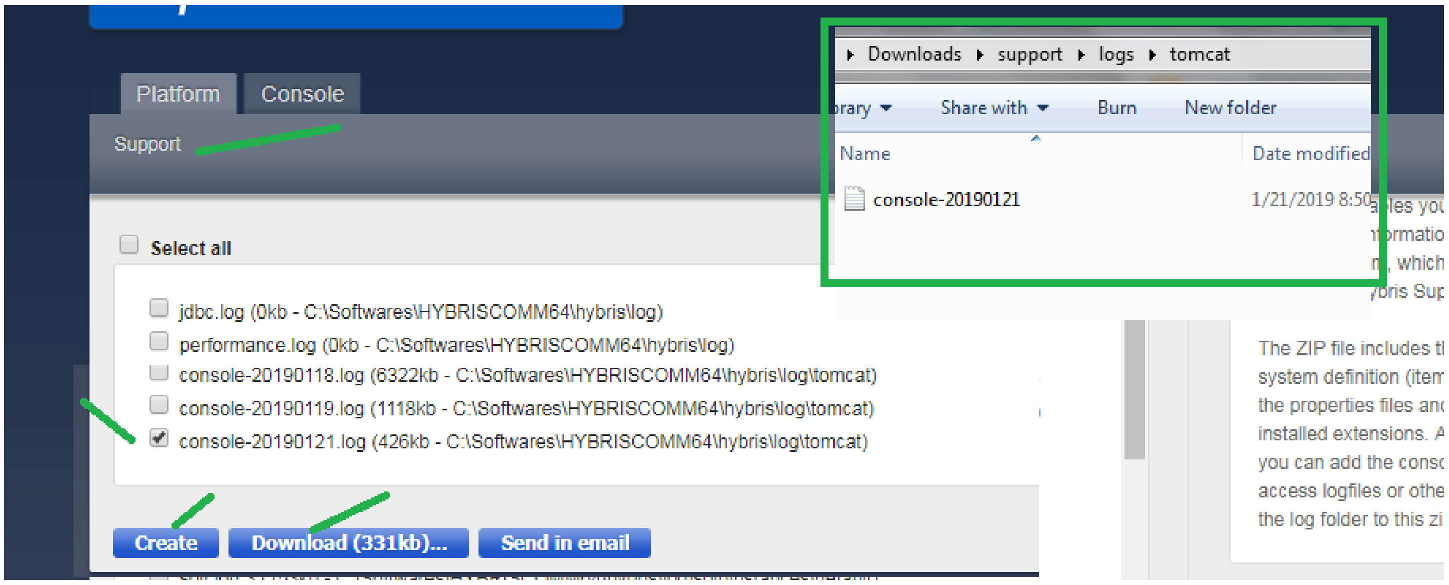
Description:

Standard Language:

Standard Currency:

Groups:

ID	Name
hac_console_flexiblesearch	n/a
employeegroup	n/a
hac_platform_support	n/a



Name	Date modified
console-20190121	1/21/2019 8:50

Select all

- jdbc.log (0kb - C:\Softwares\HYBRISCOMM64\hybris\log)
- performance.log (0kb - C:\Softwares\HYBRISCOMM64\hybris\log)
- console-20190118.log (6322kb - C:\Softwares\HYBRISCOMM64\hybris\log\tomcat)
- console-20190119.log (1118kb - C:\Softwares\HYBRISCOMM64\hybris\log\tomcat)
- [✓] console-20190121.log (426kb - C:\Softwares\HYBRISCOMM64\hybris\log\tomcat)

The ZIP file includes t
system definition (item
the properties files and
installed extensions. A
you can add the consc
access logfiles or othe
the log folder to this zi

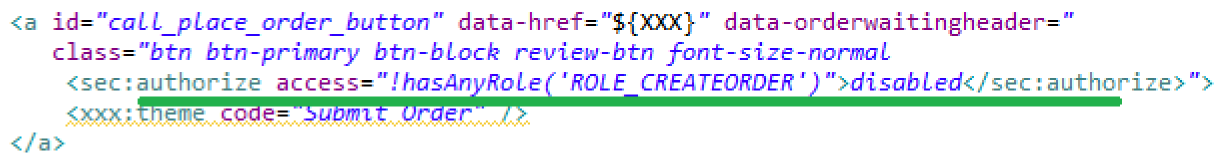
Note: - In case B2B, we will be having different types of customers.

1) Customer – Just can see the prices

2) Customer – They can see prices & Add the items to Cart

3) Customer – They can see prices, Add the items to Cart & Place Order.

Requirement = If customer having Place order role then only enable "**Submit Order**" button?.

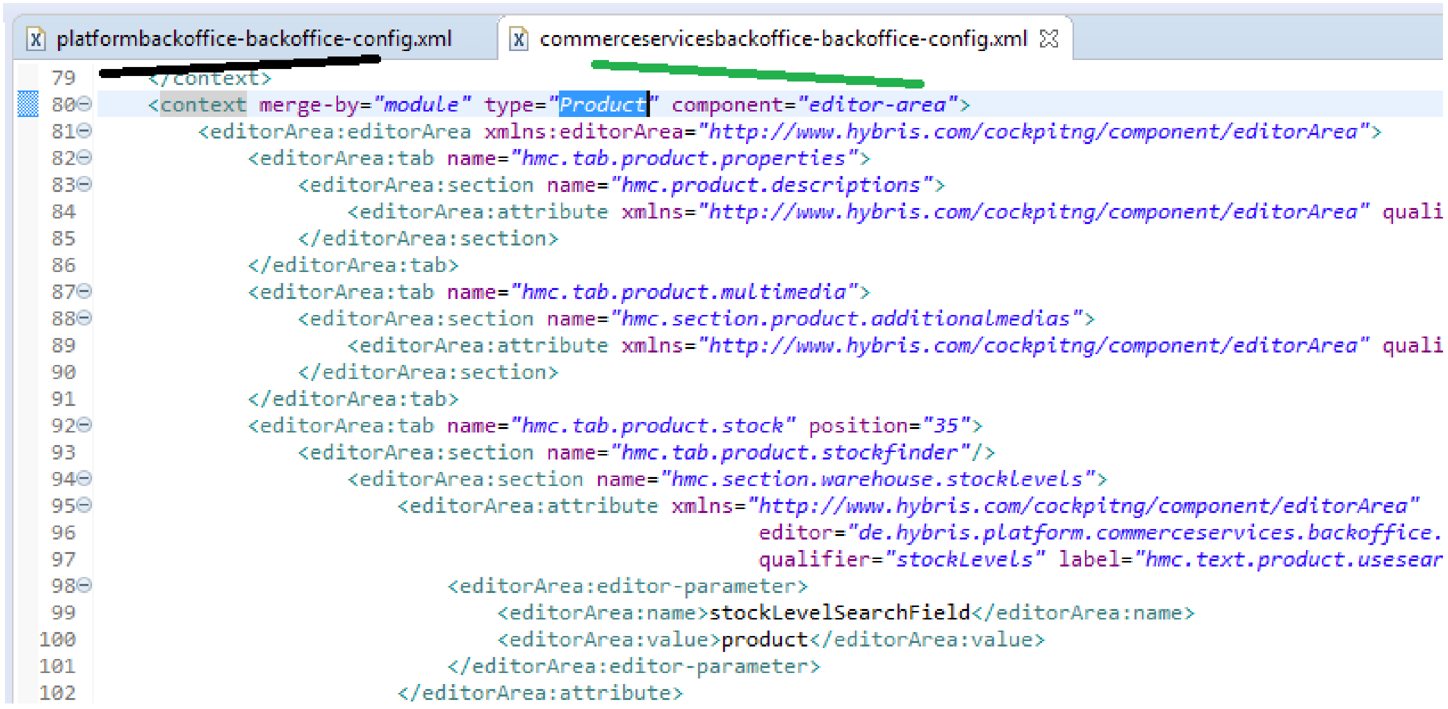


```
<a id="call_place_order_button" data-href="${XXX}" data-orderwaitingheader="
   class="btn btn-primary btn-block review-btn font-size-normal
    <sec:authorize access="!hasAnyRole('ROLE_CREATEORDER')">disabled</sec:authorize>">
    <xxx:theme code="Submit Order" />
</a>
```

Roles in Backoffice are used to specific instance of a widget / node.

Backoffice node structure is defined in **xxx-backoffice-config.xml** file.

Example: - promotionsbackoffice-backoffice-config.xml, voucherbackoffice-backoffice-config.xml, cockpit-backoffice-config.xml, platformbackoffice-backoffice-config.xml etc.



platformbackoffice-backoffice-config.xml
commerceservicesbackoffice-backoffice-config.xml

```
    </context>
    <context merge-by="module" type="Product" component="editor-area">
        <editorArea:editorArea xmlns:editorArea="http://www.hybris.com/cockpitng/component/editorArea">
            <editorArea:tab name="hmc.tab.product.properties">
                <editorArea:section name="hmc.product.descriptions">
                    <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea" quali
                </editorArea:section>
            </editorArea:tab>
            <editorArea:tab name="hmc.tab.product.multimedia">
                <editorArea:section name="hmc.section.product.additionalmedias">
                    <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea" quali
                </editorArea:section>
            </editorArea:tab>
            <editorArea:tab name="hmc.tab.product.stock" position="35">
                <editorArea:section name="hmc.tab.product.stockfinder"/>
                    <editorArea:section name="hmc.section.warehouse.stocklevels">
                        <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea"
                                              editor="de.hybris.platform.commerceservices.backoffice.
                                              qualifier="stockLevels" label="hmc.text.product.usesear
                            <editorArea:editor-parameter>
                                <editorArea:name>stockLevelSearchField</editorArea:name>
                                <editorArea:value>product</editorArea:value>
                            </editorArea:editor-parameter>
                        </editorArea:attribute>
```