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?

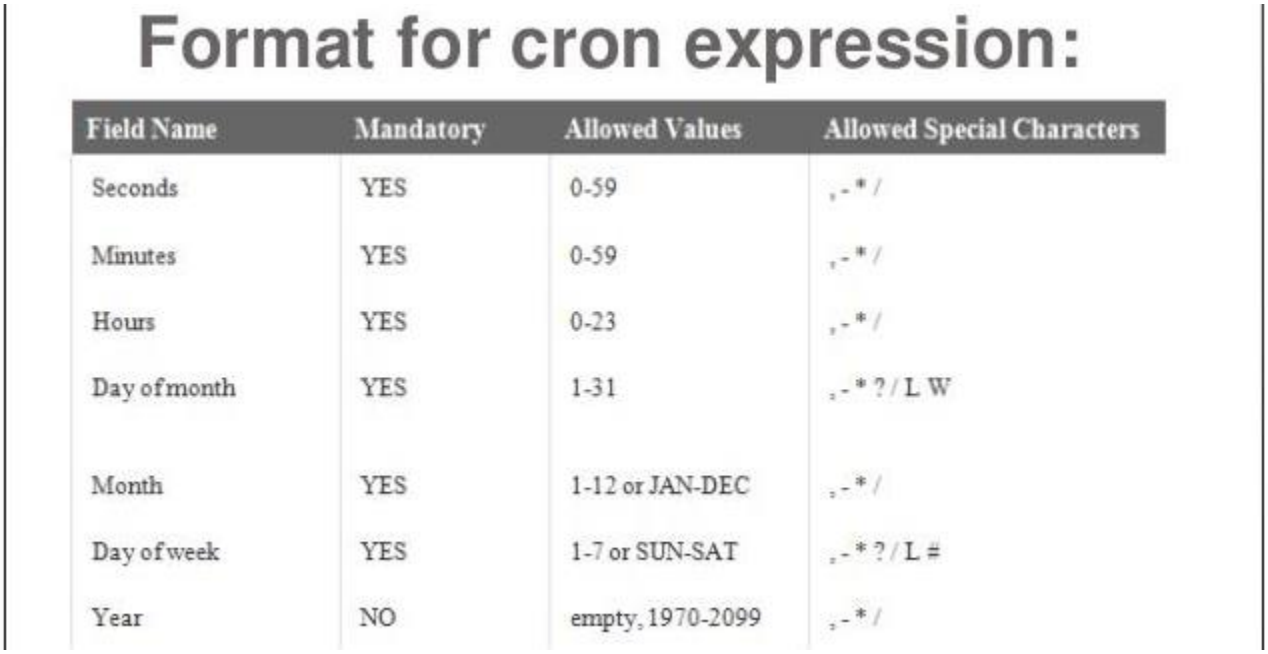
1) Trigger = This is used to schedule when to run the job.

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	, - * /

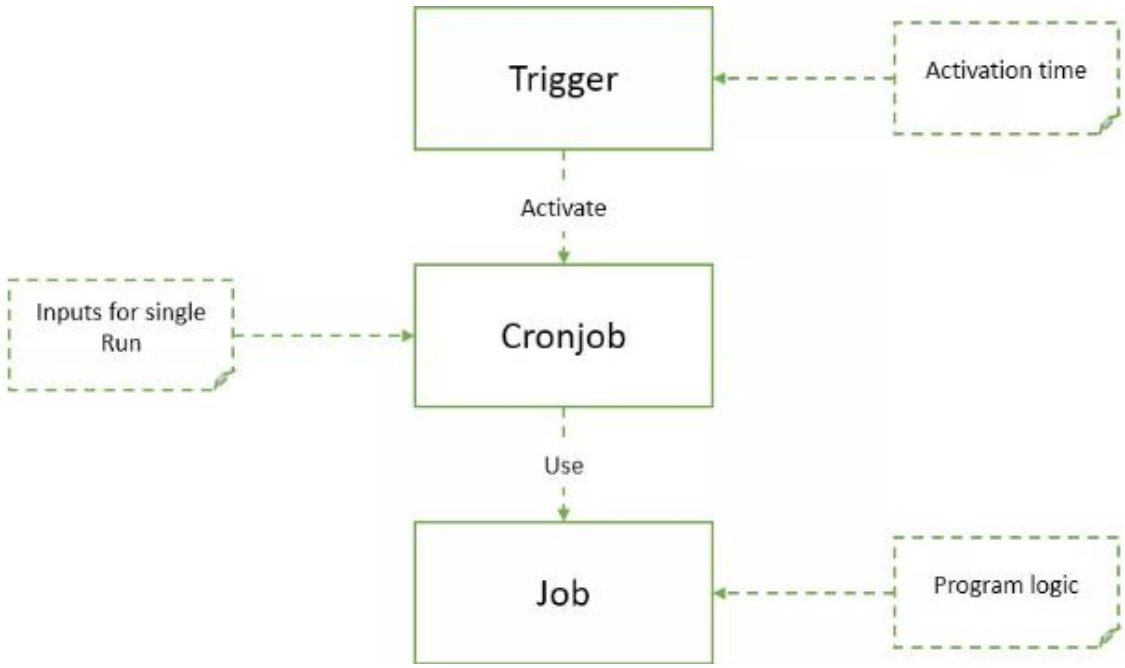
2) Cronjob = This hold the business logic which needs to be executed at specified time.

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

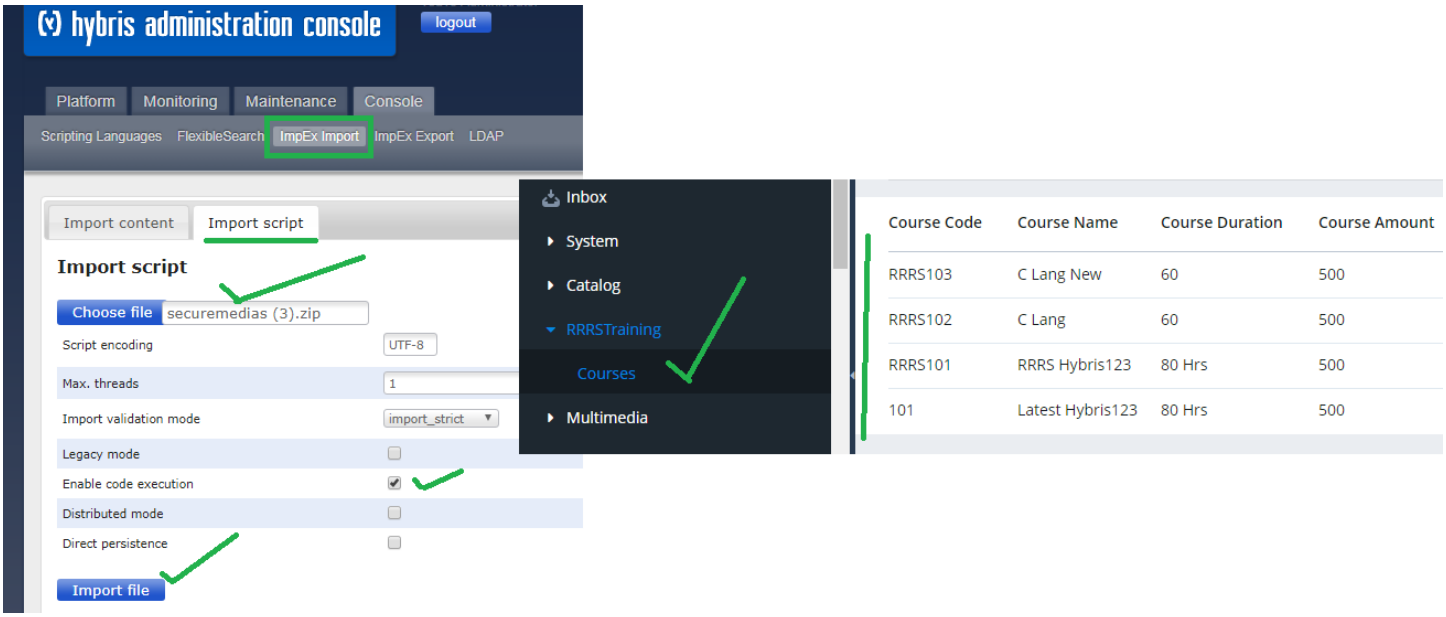
2 Ways: -

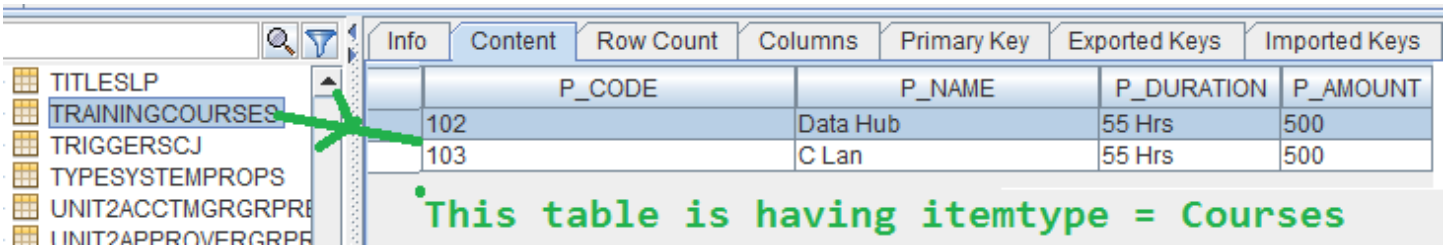
(a) Create a class which extends **AbstractJobPerformable**

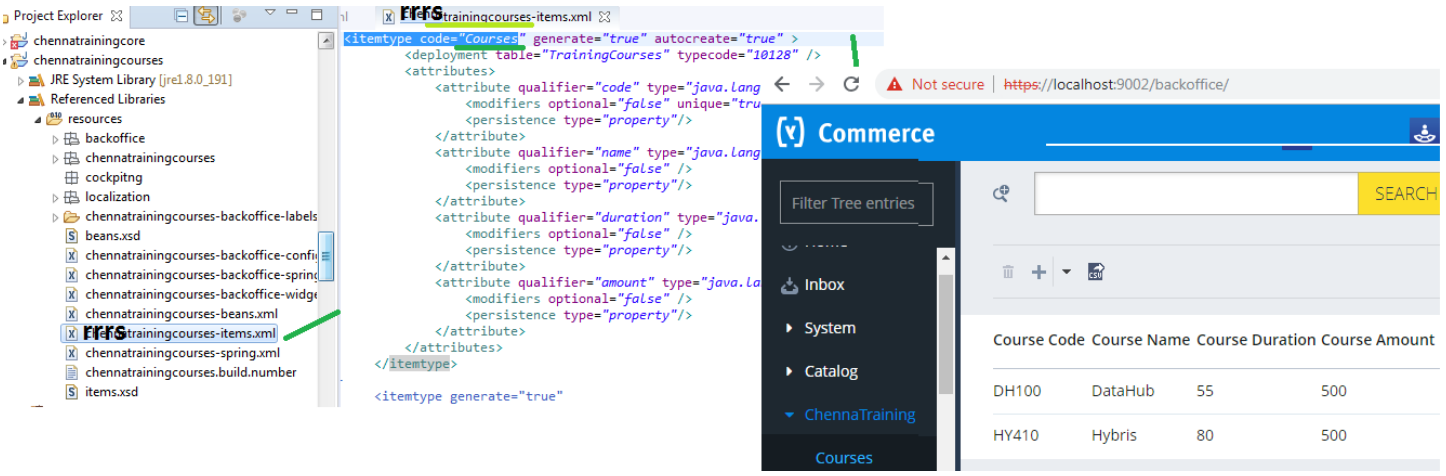
(b) Create a class which implements **JobPerformable**



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







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

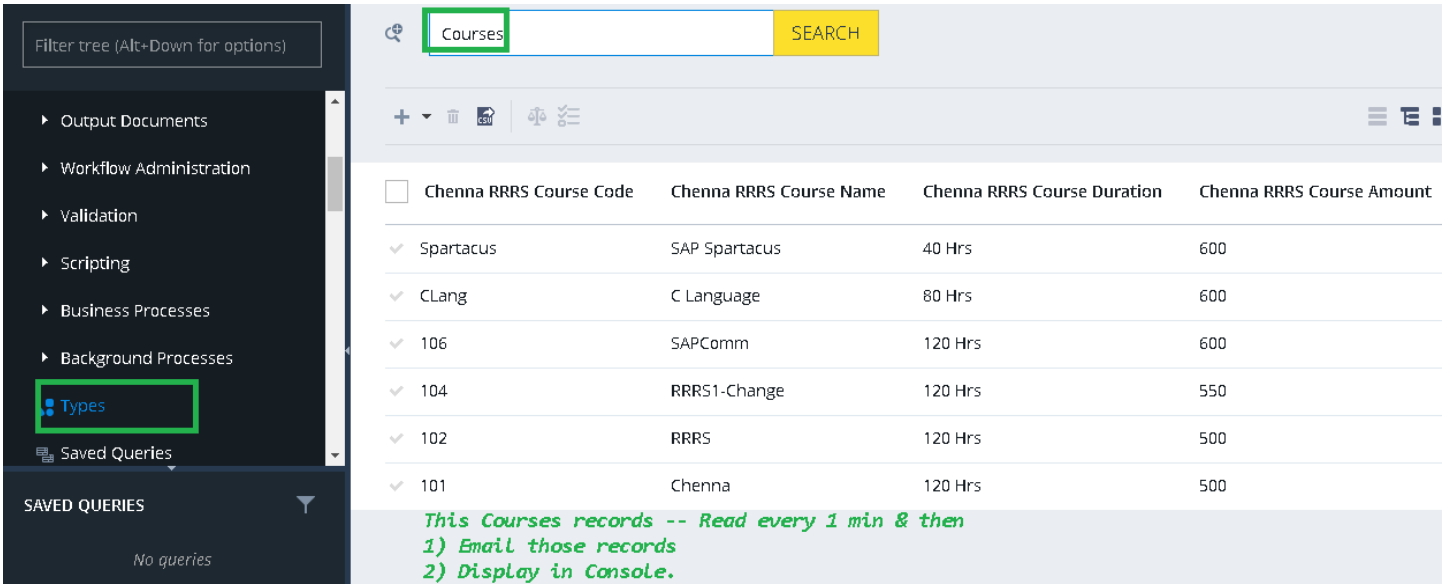
Step 1 = Create **Courses** Itemtype (or) **TrainingCourses** table.

Also – Insert the records.

Example =

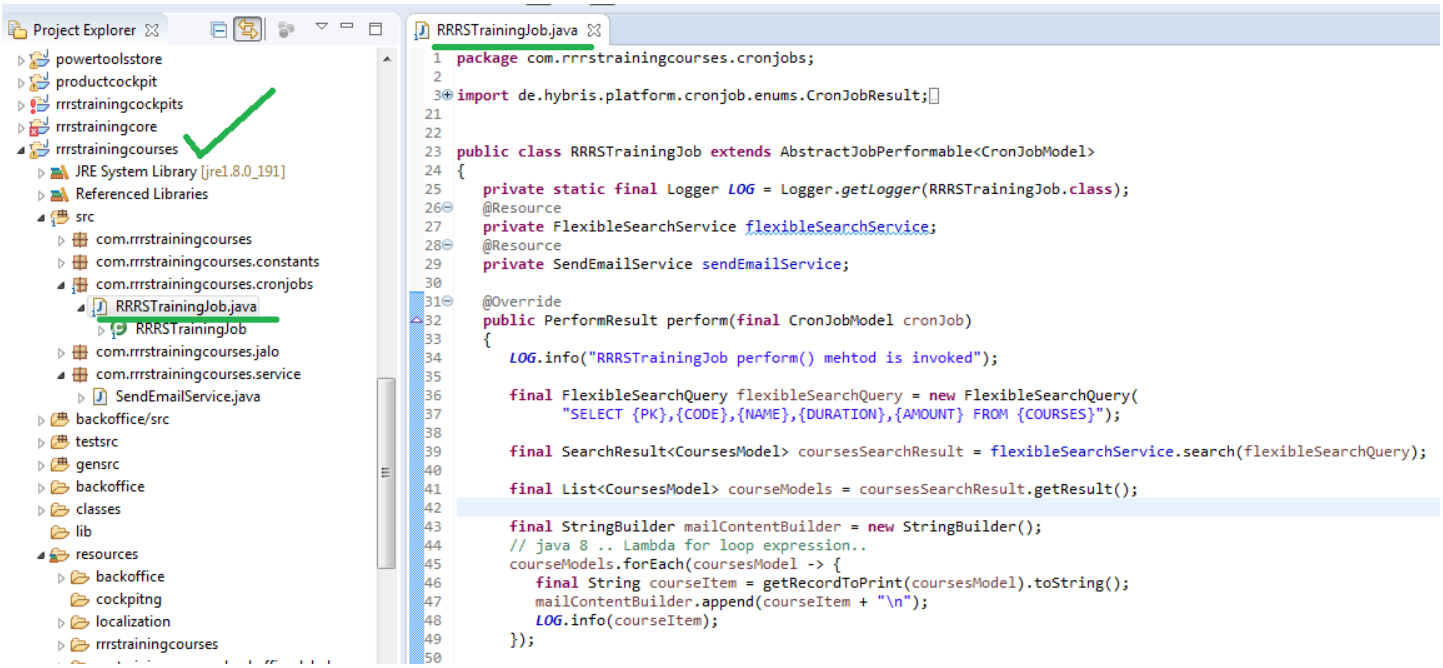
chennarrrstrainingcore-items.xml | hot-folder-store-rrrs-spring.xml | chennarrrstrainingcore-spring.xml

```
211     <!-- Example 6 =  Create Courses Item type -->
212     <!-- Example 11 =  Cronjobs -->
213     <itemtype code="Courses" generate="true" autocreate="true" >
214     <deployment table="TrainingCourses" typecode="10128" />
215     <attributes>
216         <attribute qualifier="code" type="java.lang.String">
217             <modifiers optional="false" unique="true" />
218             <persistence type="property"/>
219         </attribute>
220         <attribute qualifier="name" type="java.lang.String">
221             <modifiers optional="false" />
222             <persistence type="property"/>
223         </attribute>
224         <attribute qualifier="duration" type="java.lang.String">
225             <modifiers optional="false" />
226             <persistence type="property"/>
227         </attribute>
228         <attribute qualifier="amount" type="java.lang.Integer">
229             <modifiers optional="false" />
230             <persistence type="property"/>
231         </attribute>
232     </attributes>
233     </itemtype>
234
```

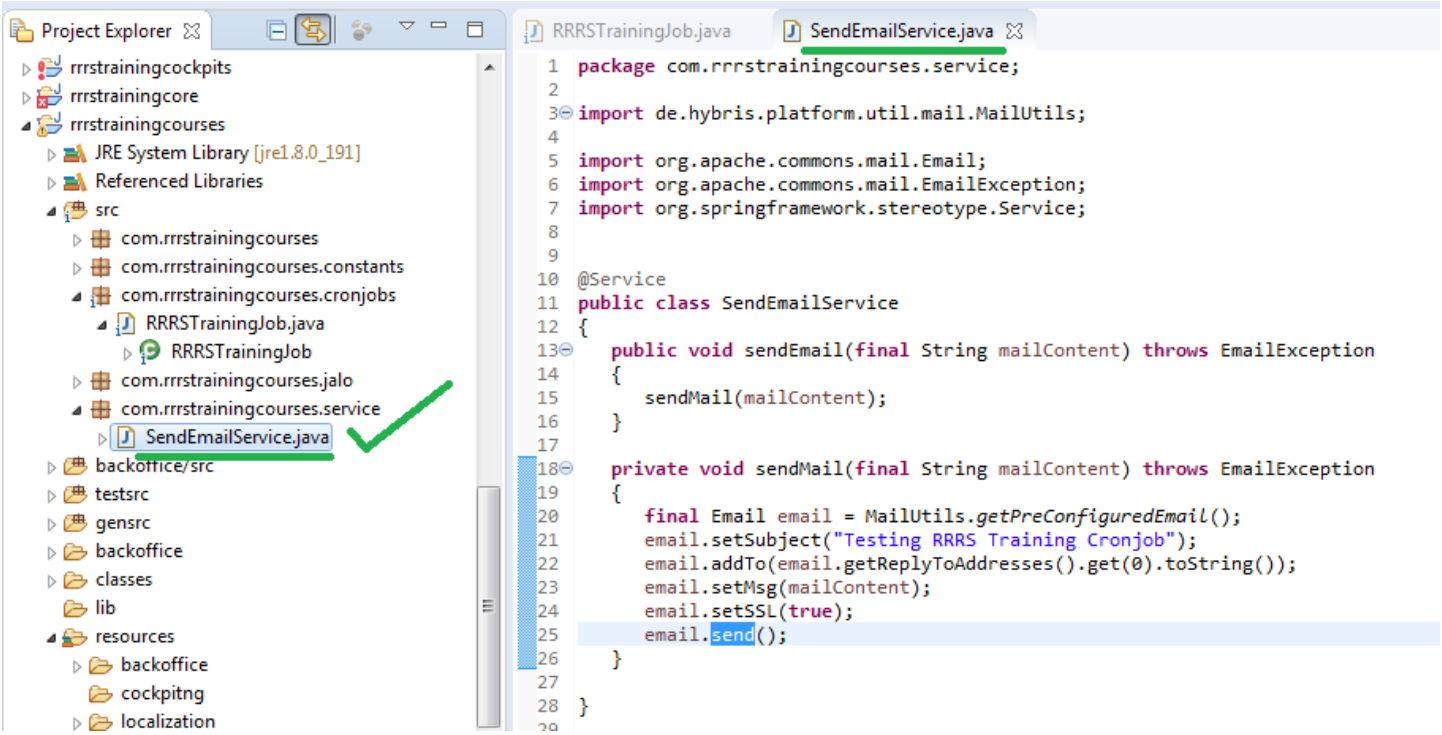


Step 2 = Create new class called "**RRRSTrainingJob.java**" by extending "**AbstractJobPerformable**".

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



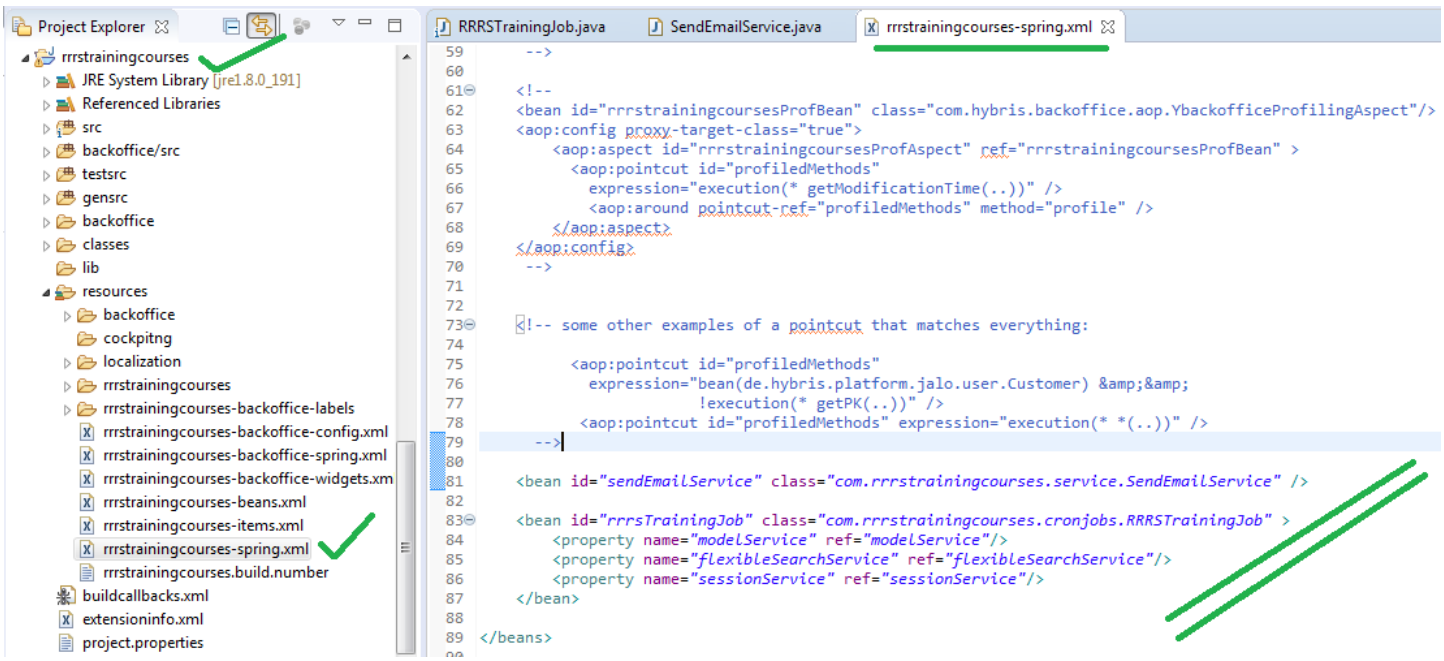
Step 3 = Create an Email Service called "**SendEmailService.java**"



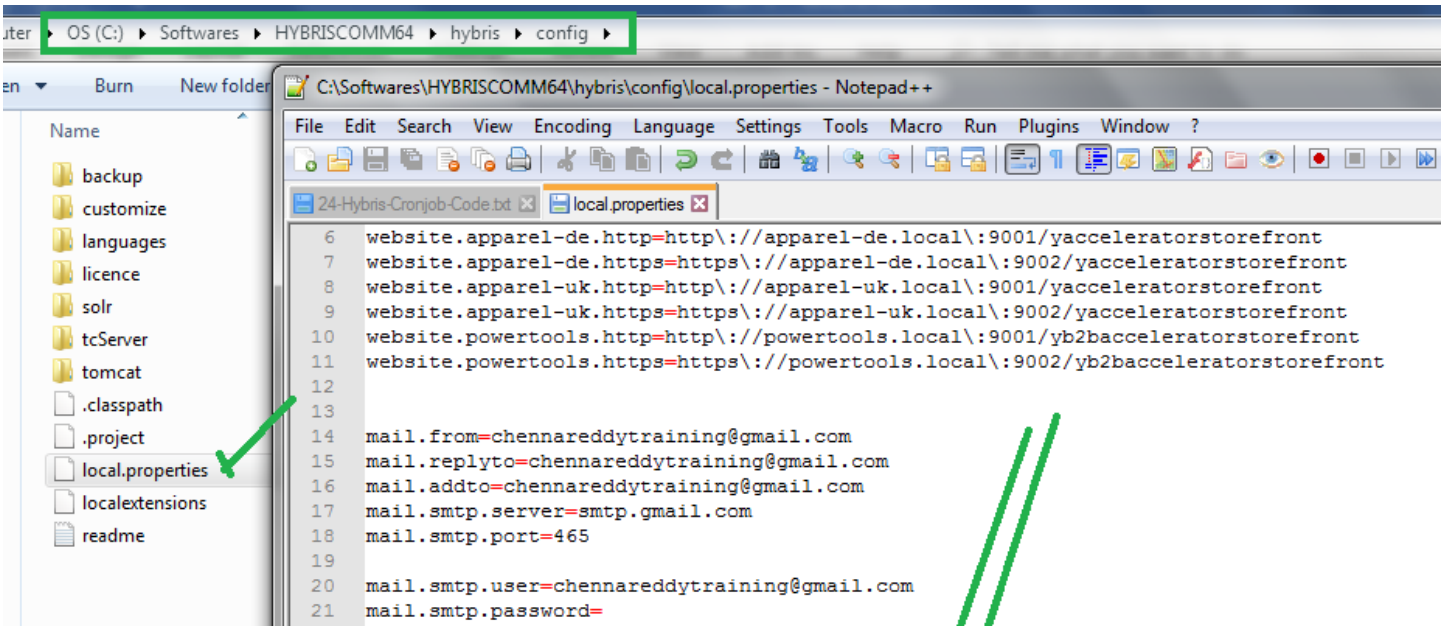
Step 4 = we have written "**RRRSTrainingJob.java**" file

& "**SendEmailService.java**".

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

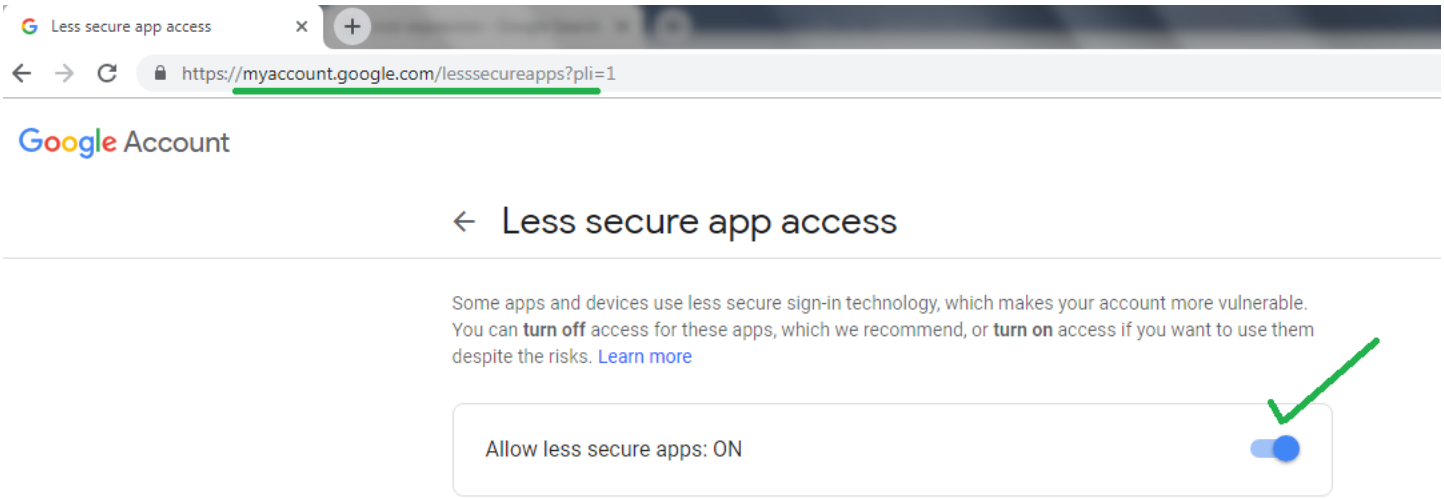


Step 5 = Define email properties in “local.properties” file



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

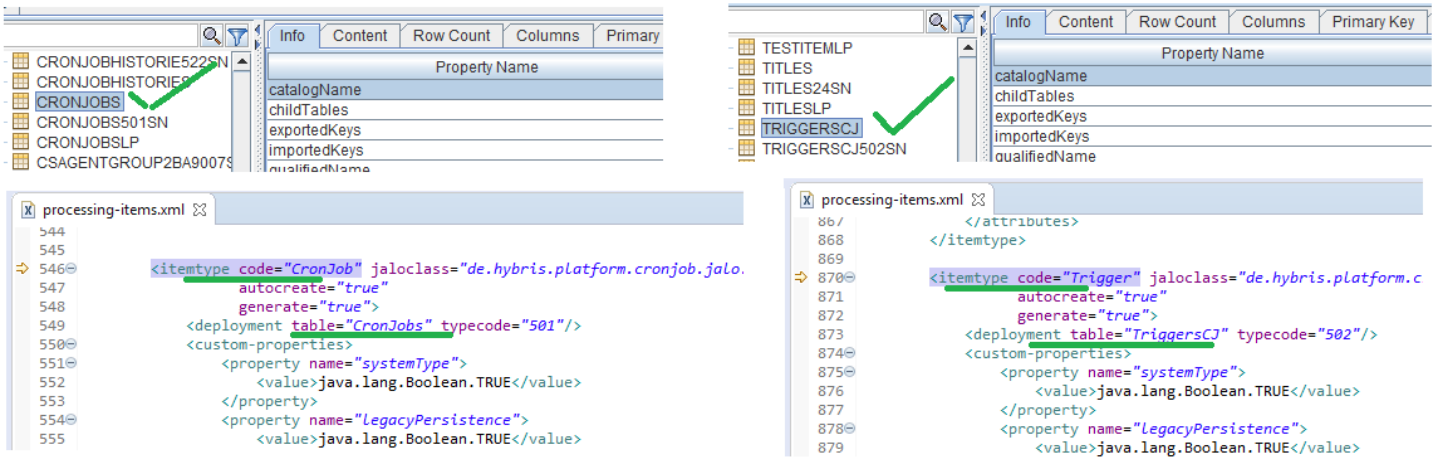


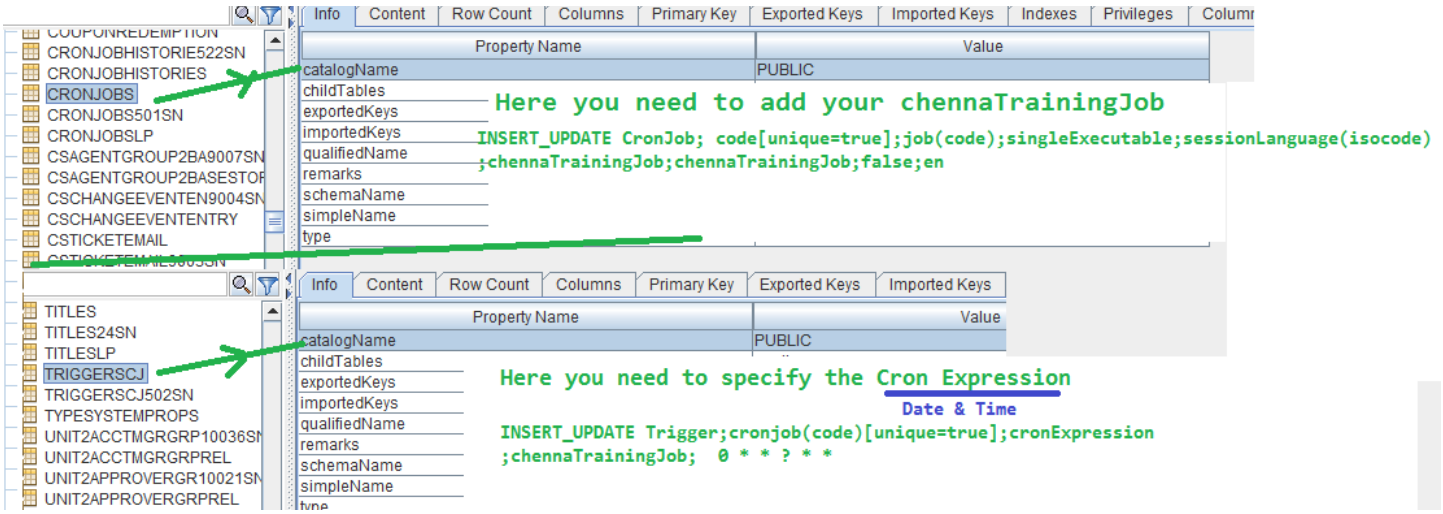
Step 7 = Do the build (**ant clean all**)

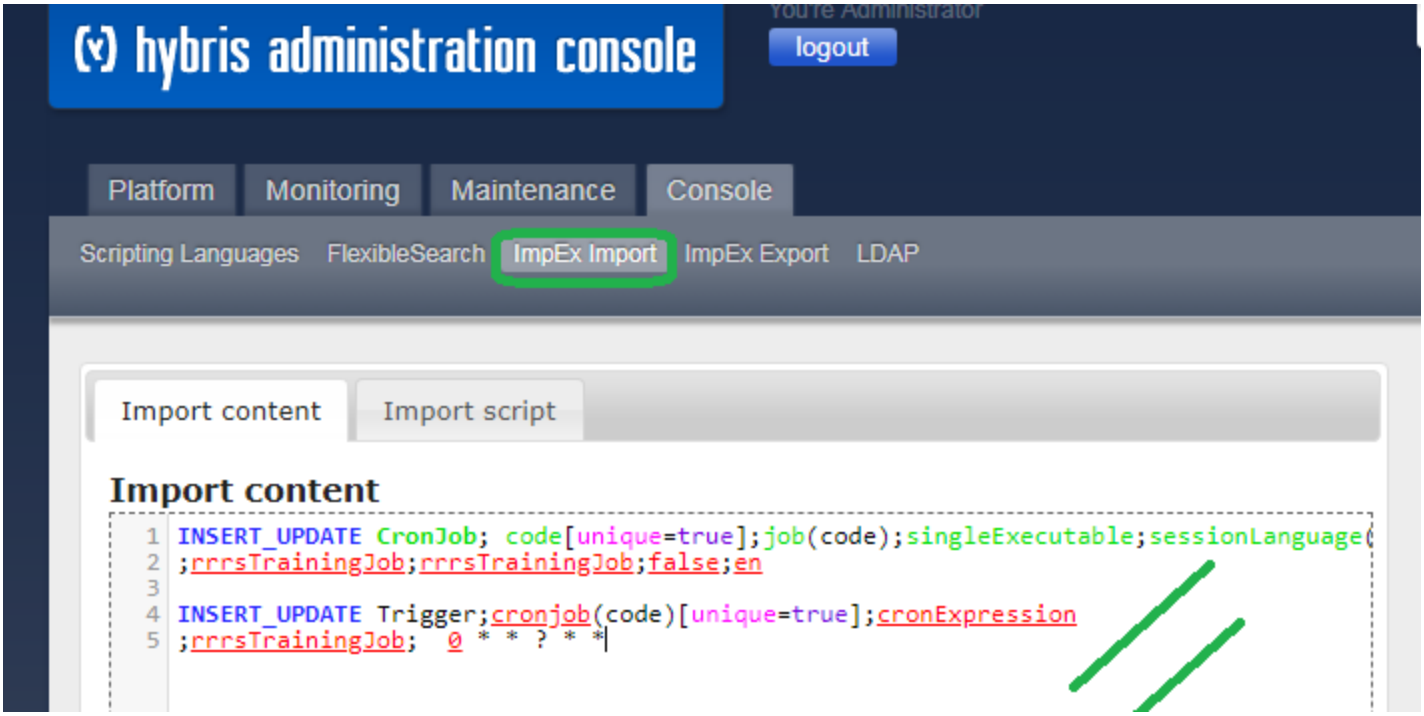
Step 9 = Start the Server (**hybrisserver.bat**)

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

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



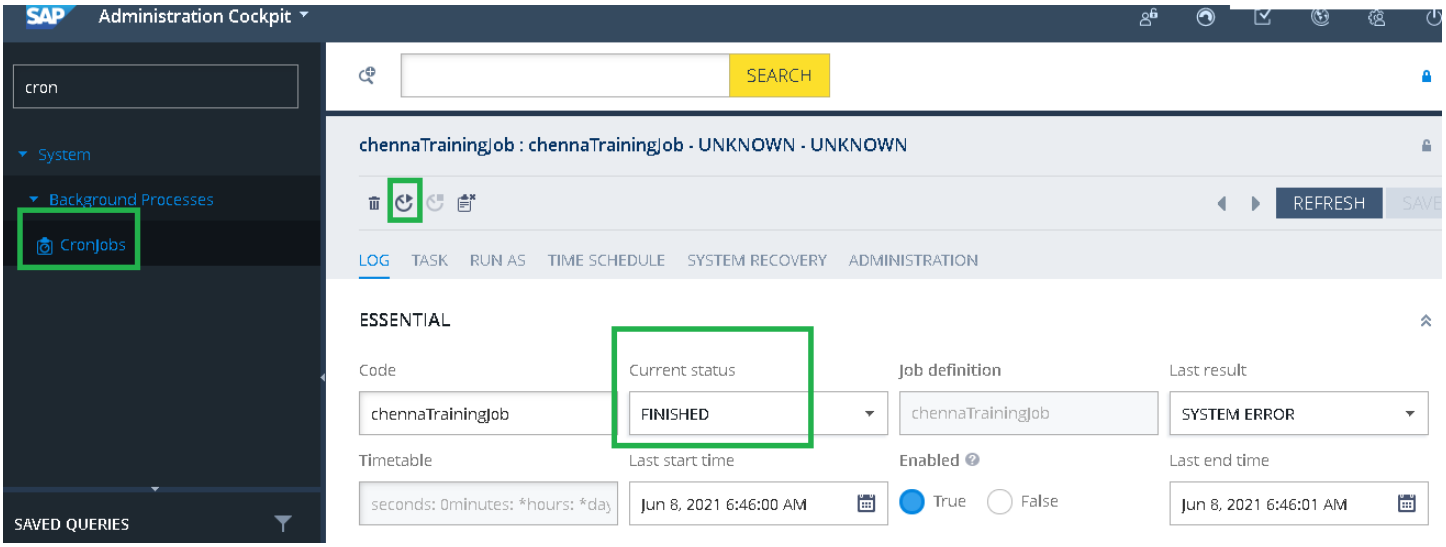





```
INSERT_UPDATE CronJob; code[unique=true];job(code);singleExecutable;sessionLanguage(isocode)
;rrrsTrainingJob;rrrsTrainingJob;false;en

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

Step 11 = Test the Results: -

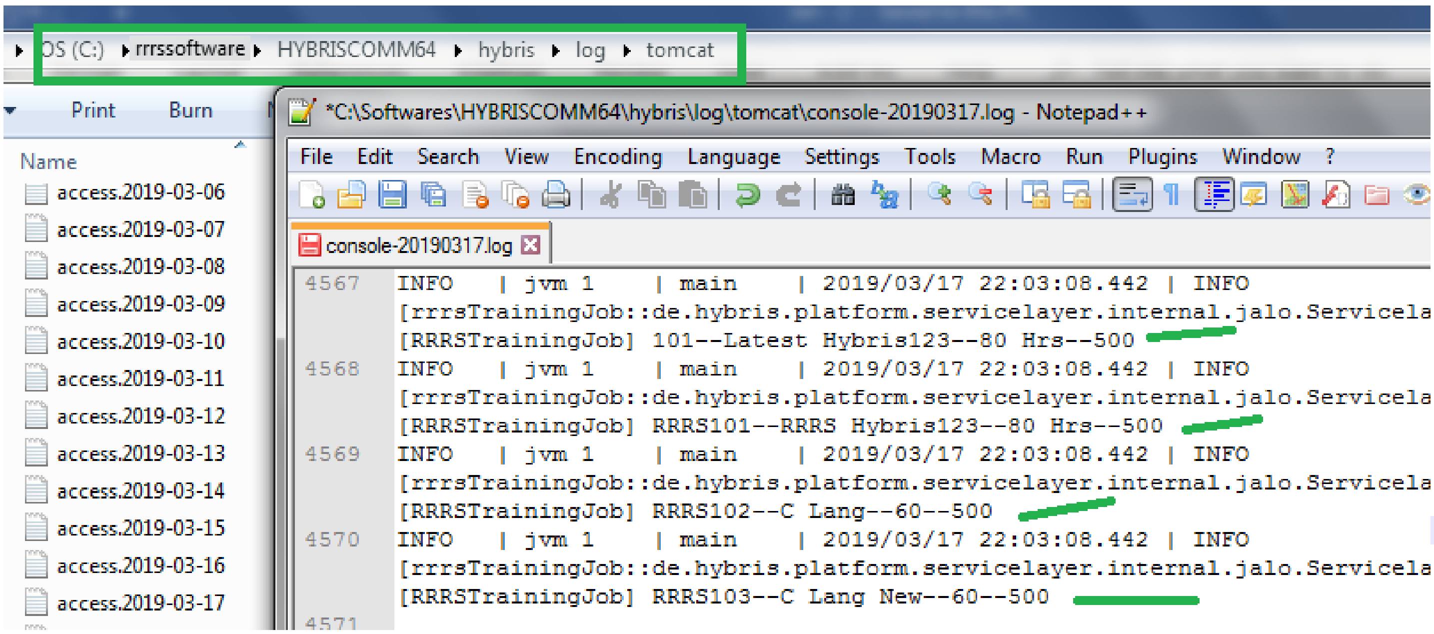


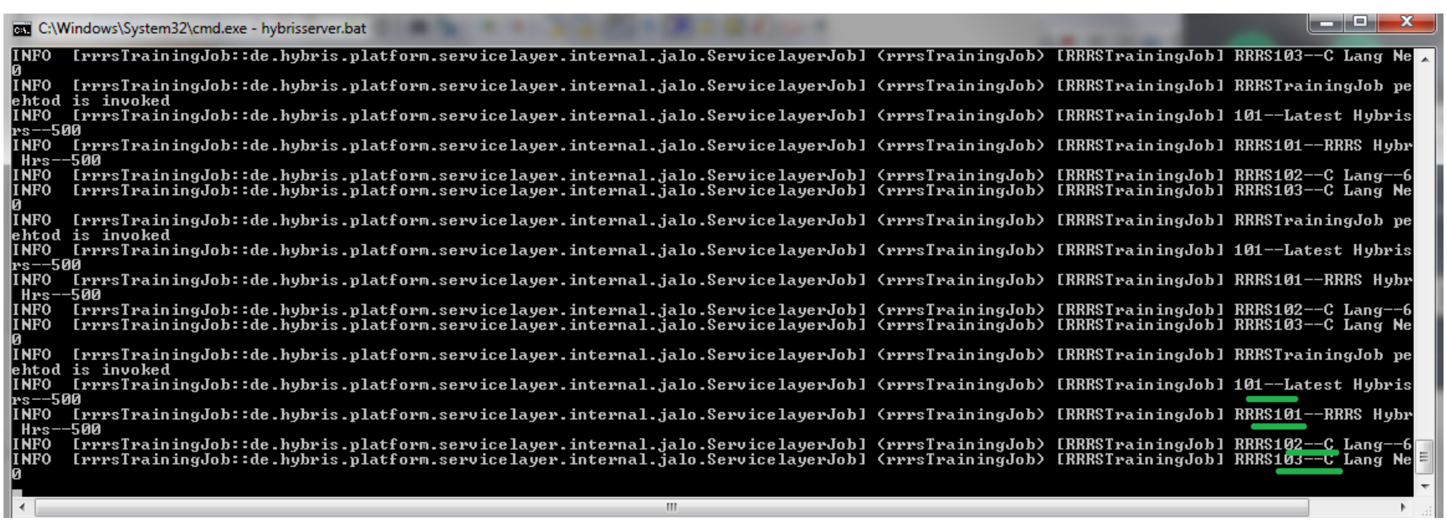
Testing RRRS Training Cronjob Inbox ×

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

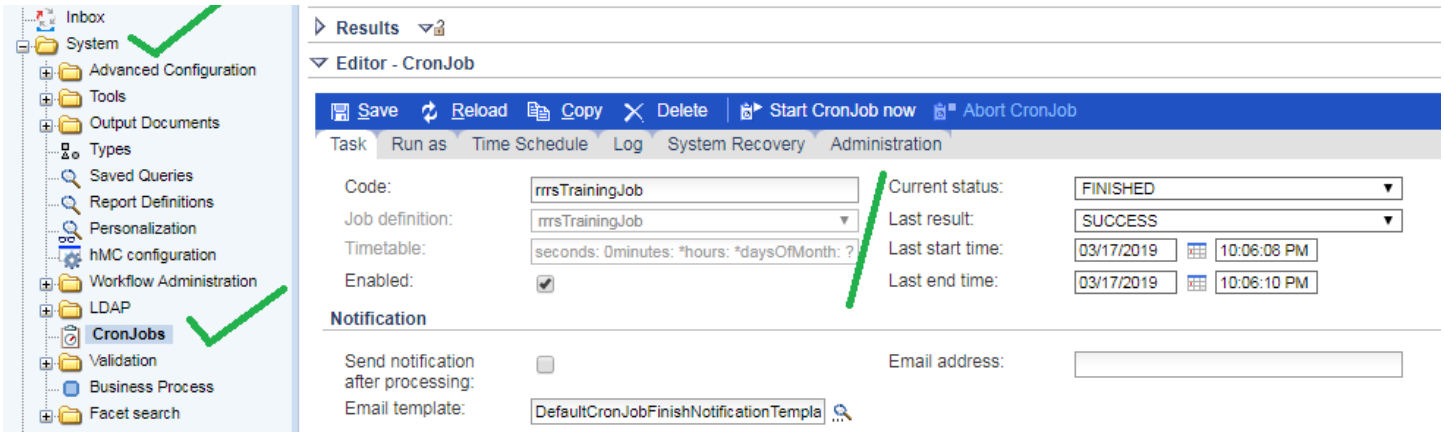
to me

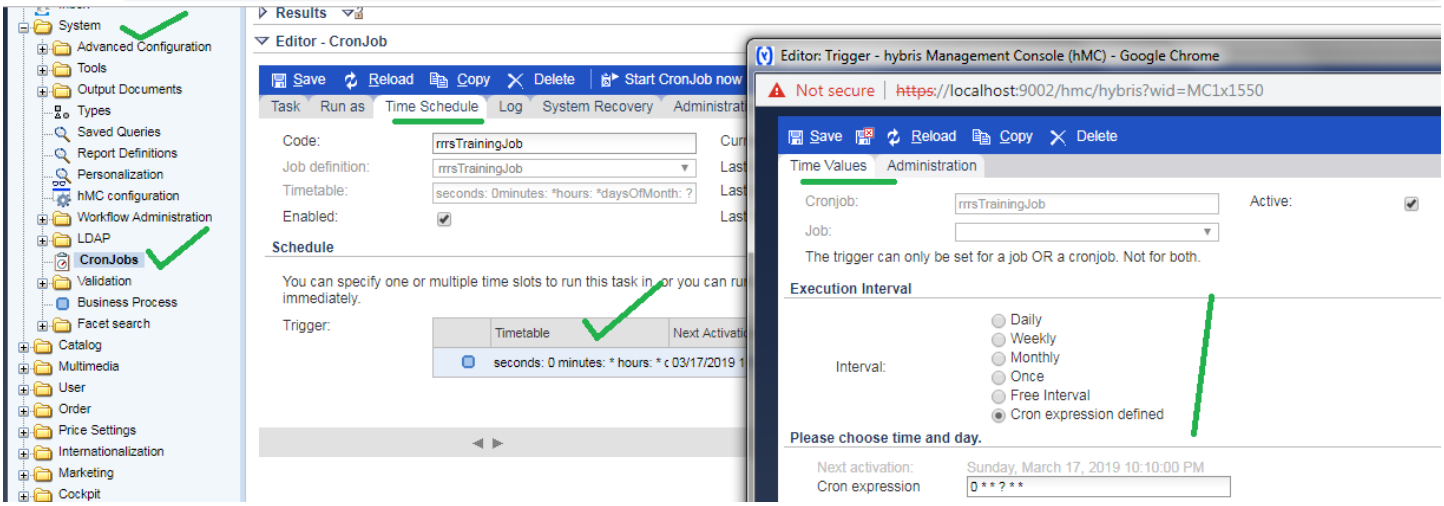
```
101--Latest Hybris123--80 Hrs--500
RRRS101--RRRS Hybris123--80 Hrs--500
RRRS102--C Lang--60--500
RRRS103--C Lang New--60--500
```





Note: - We can also run the Cronjobs manually.





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} = 'RRRSTrainingIJob'

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 "/WEB-INF/views/responsive/cms/assistedservicecomponent.jsp" 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

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.

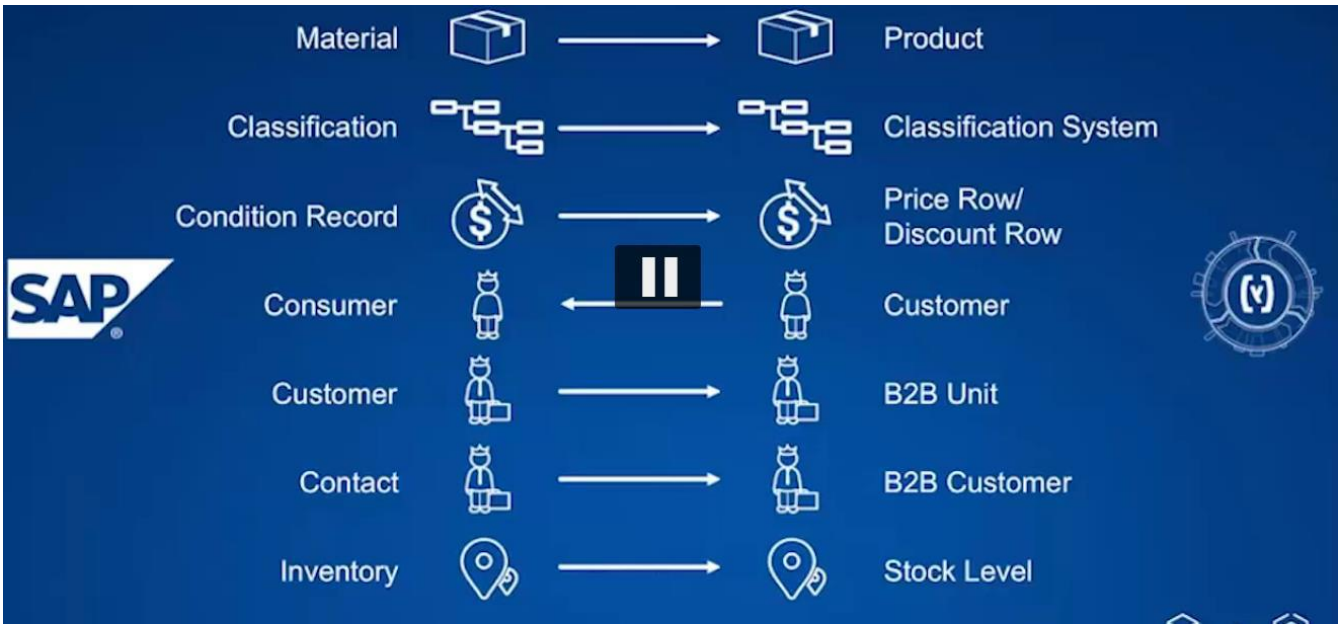
Also, **Async** integration – We can do in N number of ways.

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: -

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

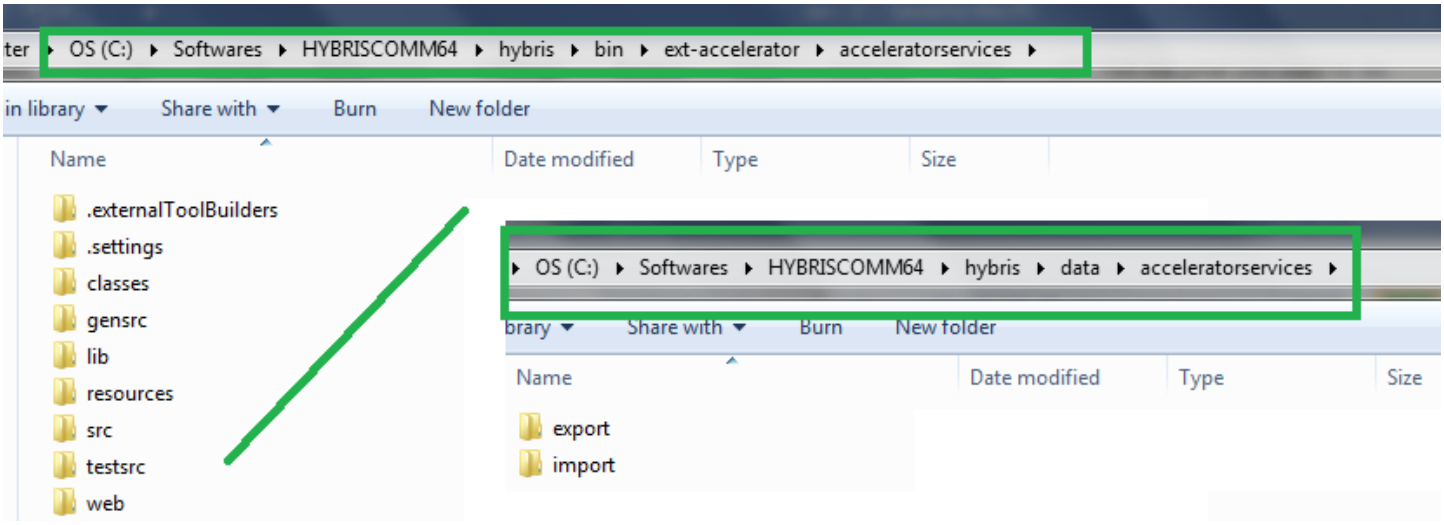
✓ The decision point (order status change) need to be notified (Push either email / mobile) to customer. If push notification is required then it is best to replicate data.

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

Explain Hot Folder

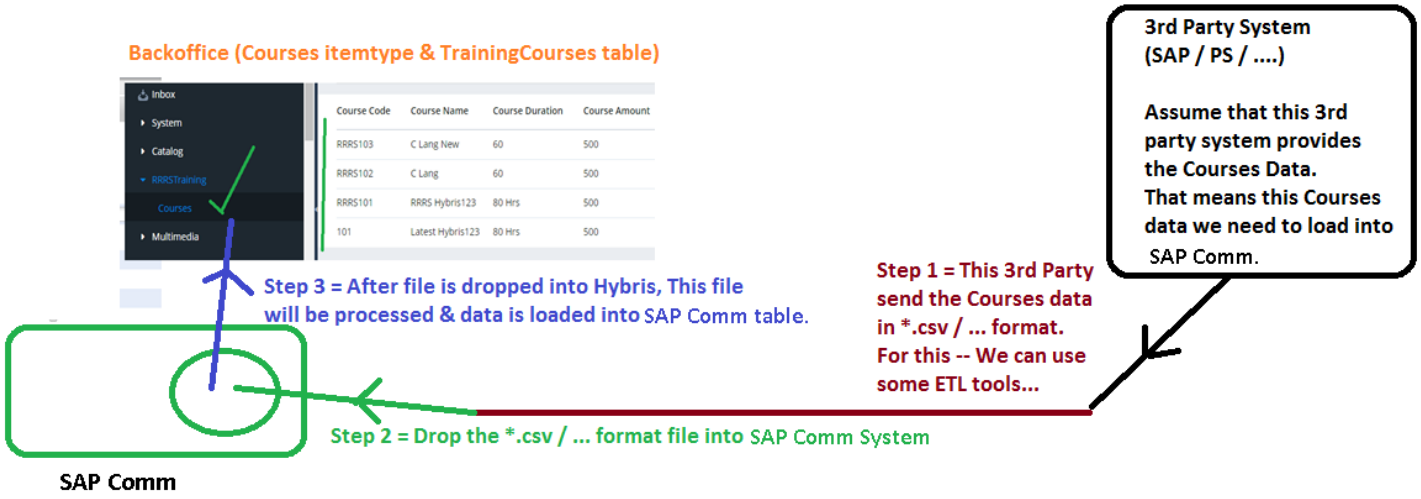
You have already seen how you can use ImpEx files to **import data** into the system. “SAP Comm” supports Hot Folders, which are folders from which data can be automatically imported into the platform by simply placing the data inside of the folder.

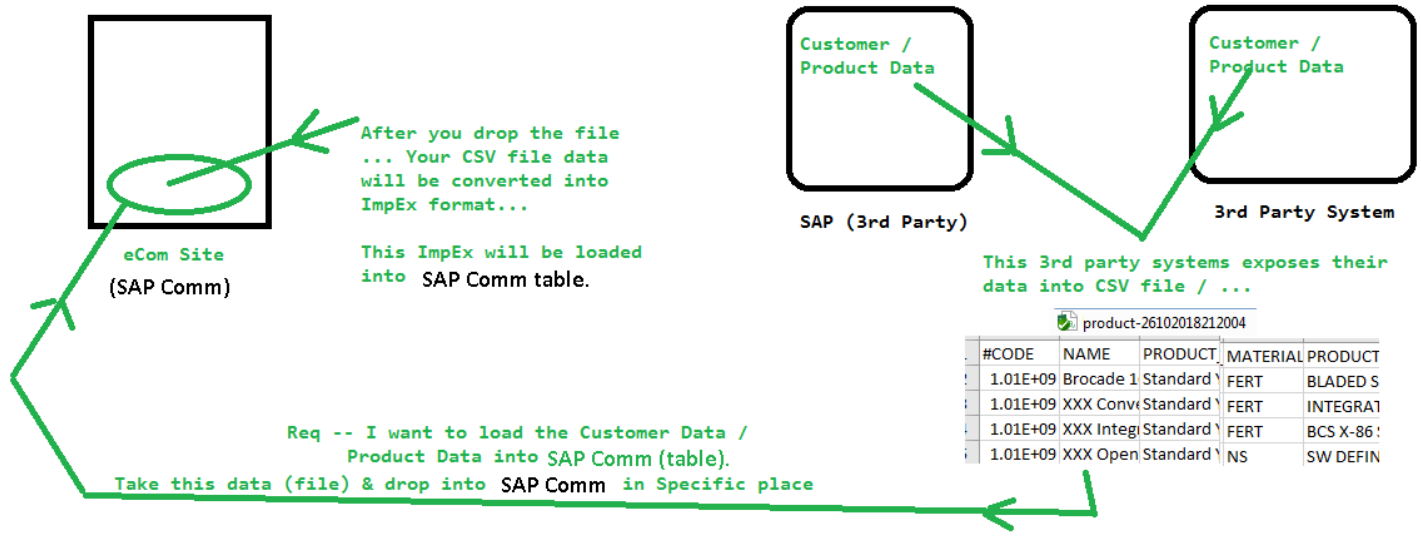
“SAP Comm” **acceleratorservices** extension template comes with a batch package that enables automated importing of data from hot folders.



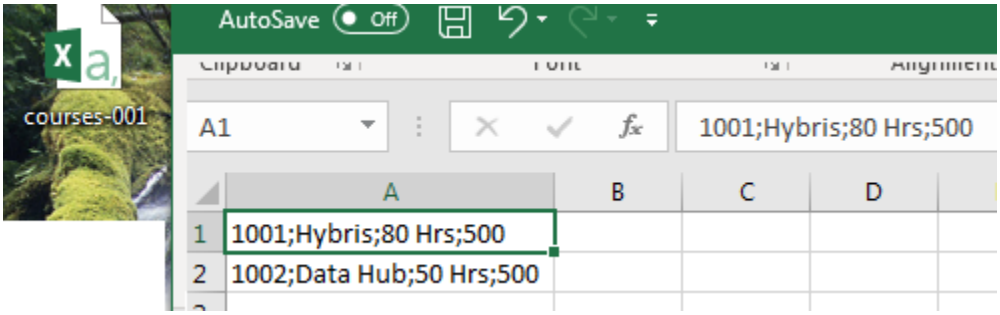
The infrastructure enables using simple CSV files (internally translated into “SAP Comm” ImpEx scripts) to import content directly into the product catalogs.

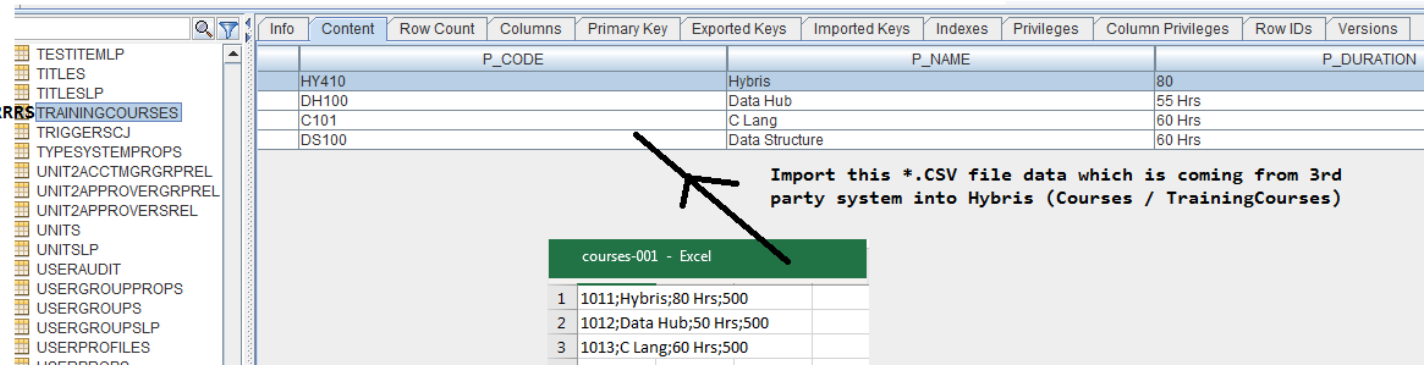
This infrastructure uses **Spring Integration** FWK. Spring Integration provides a pluggable, highly configurable service-based design that can be extended as required.



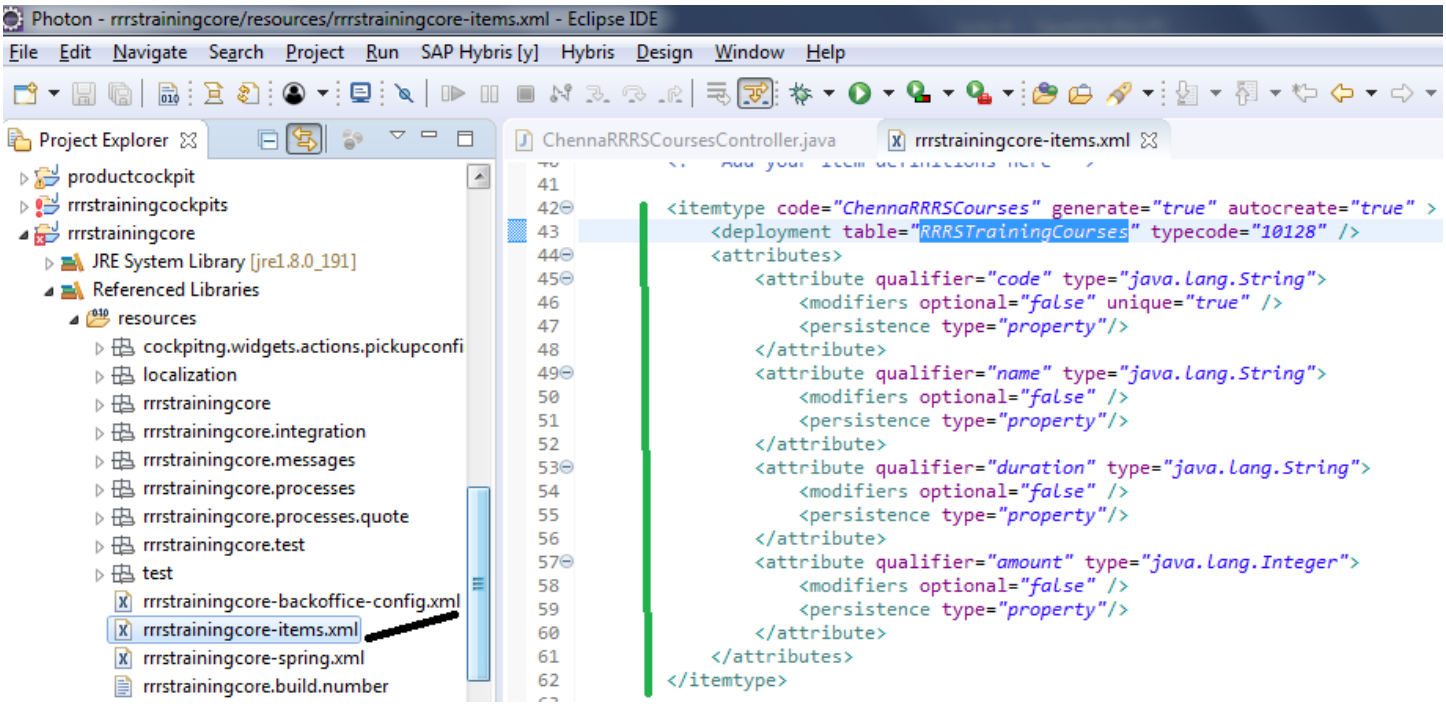


Note: - Assume that, your 3rd party system giving below Courses.csv file.



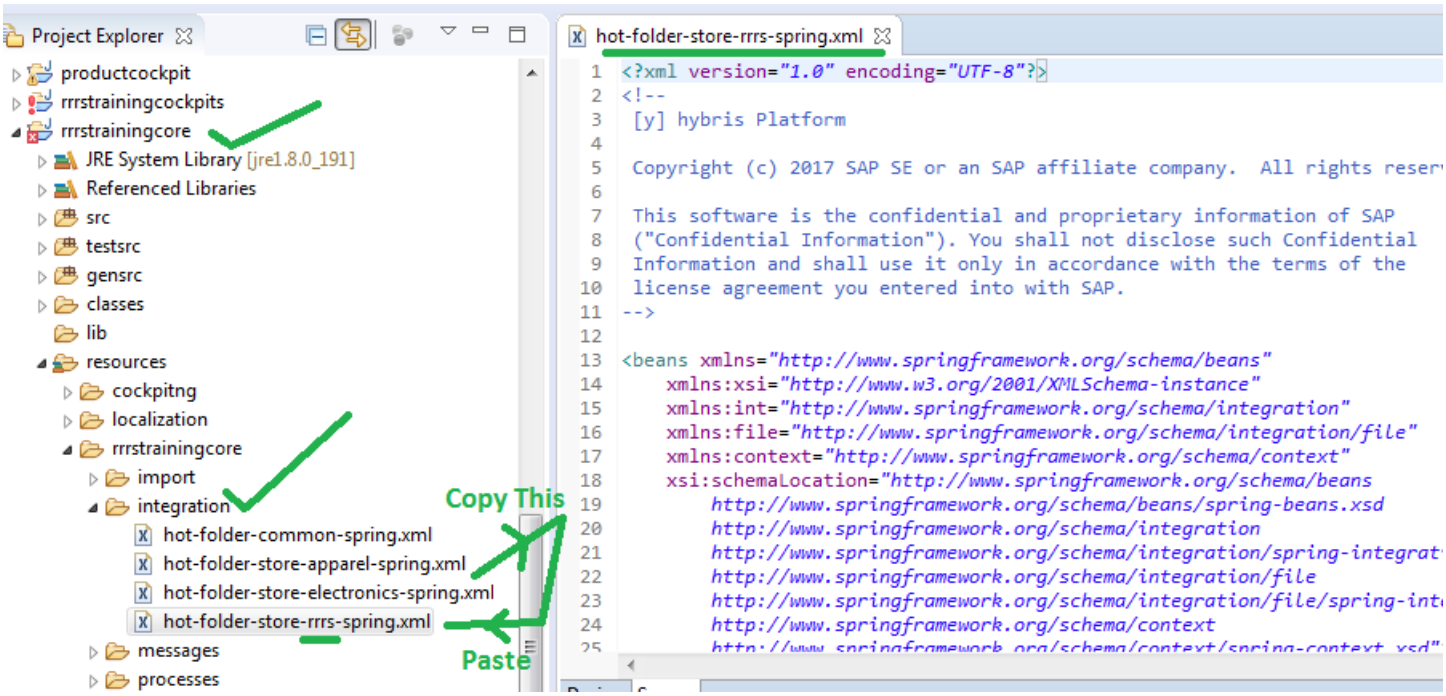


Step 1 = Create New Item Type called "**ChennaRRRSCourses**" & Table Name = "**RRRSTrainingCourses**"



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

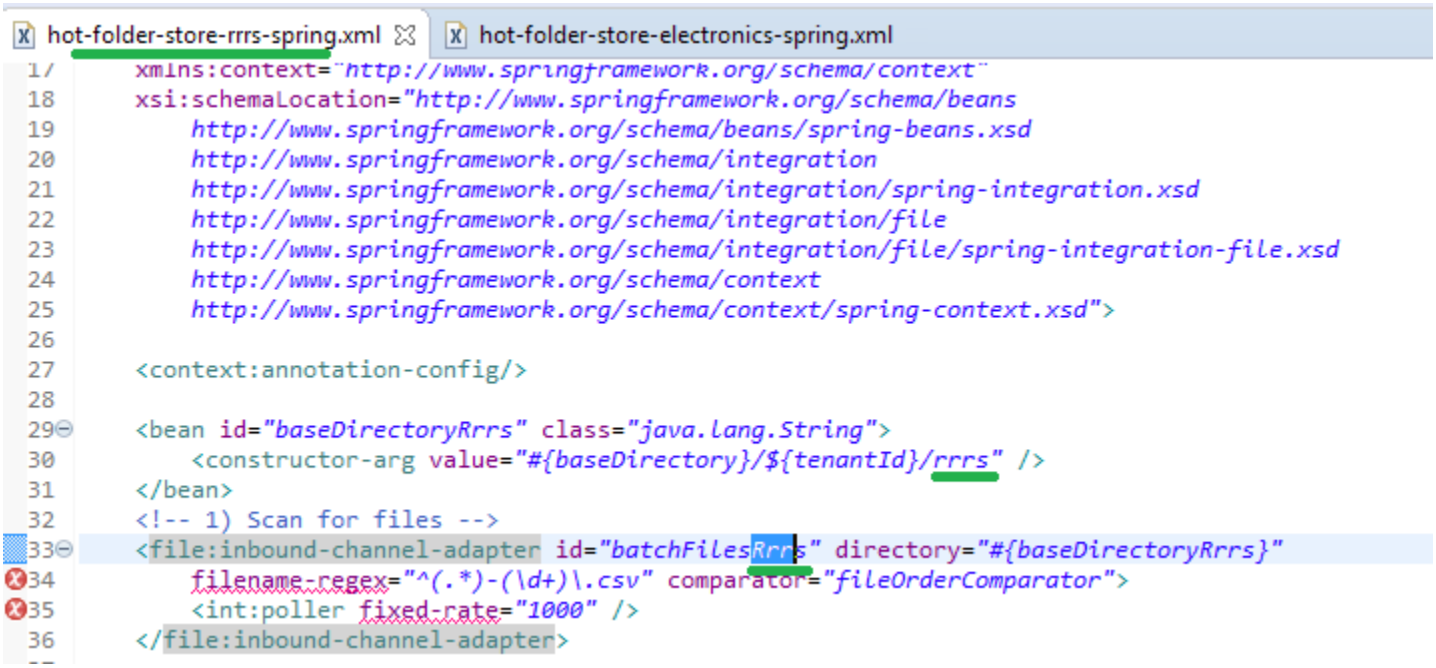
Copy "**hot-folder-store-electronics-spring.xml**" & Paste with different name "**hot-folder-store-rrrs-spring.xml**"



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**



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>
```

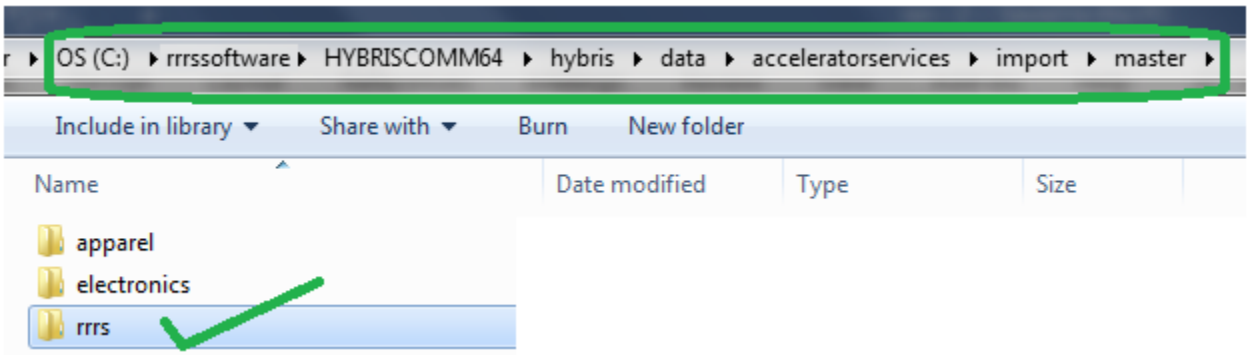
2) Inbound adapter = This scan for the matching files

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

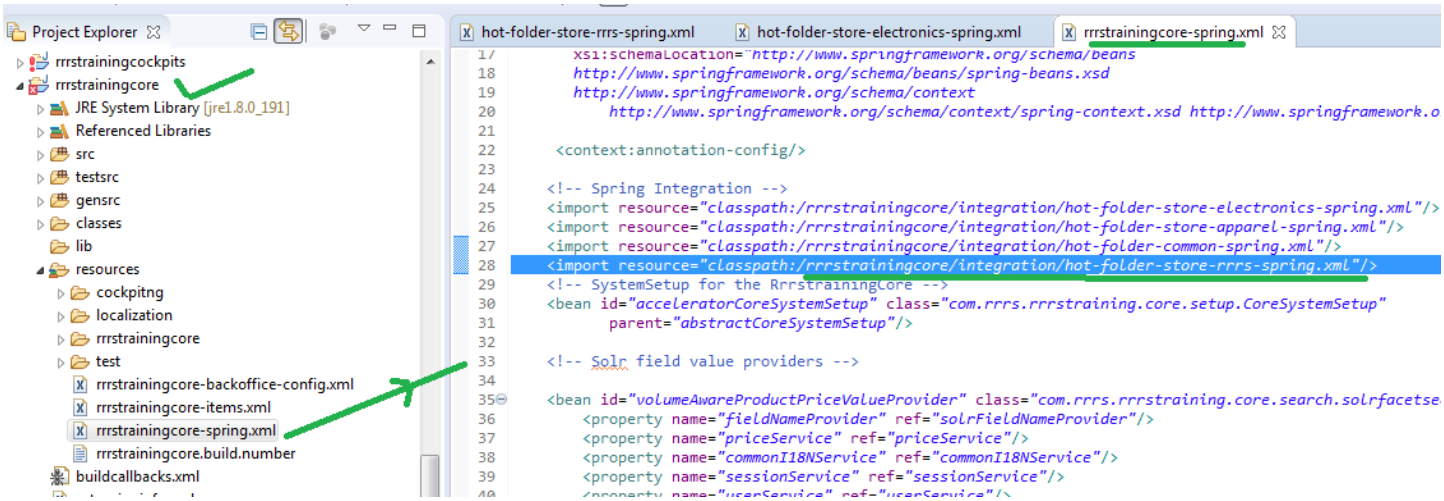
3) Outbound adapter = This takes the file & move for processing.

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

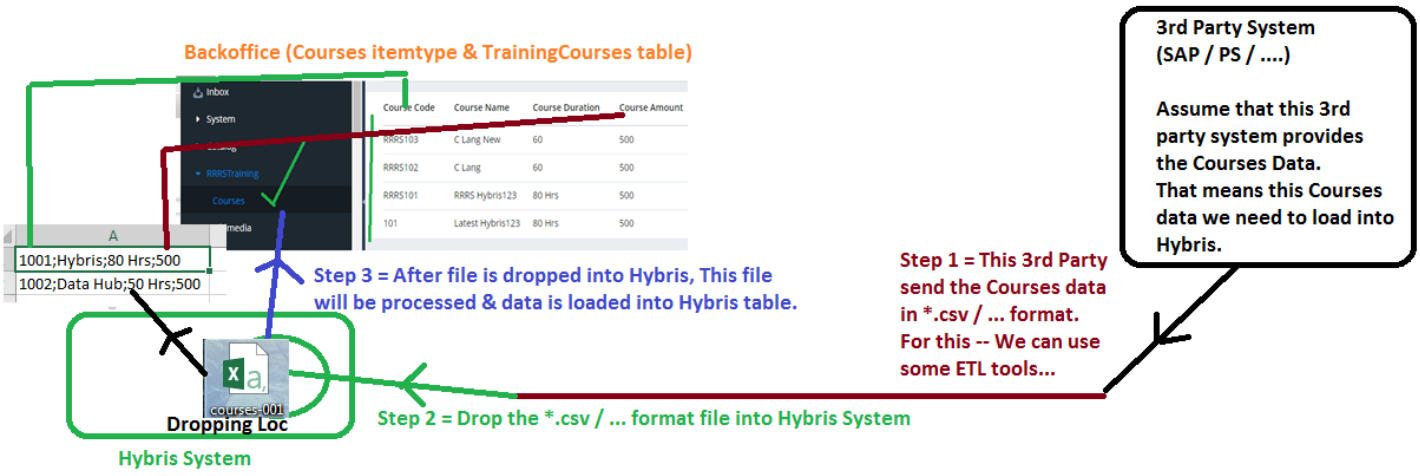
Step 4 = Create "**rrrs**" folder.



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



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



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**.

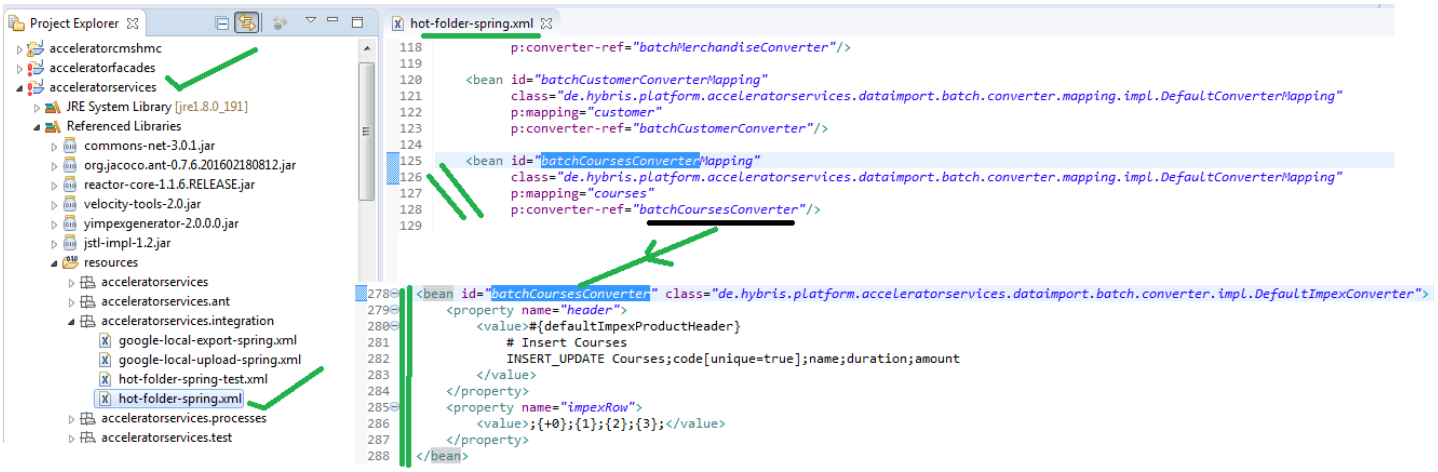
Do the Mapping like: -

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

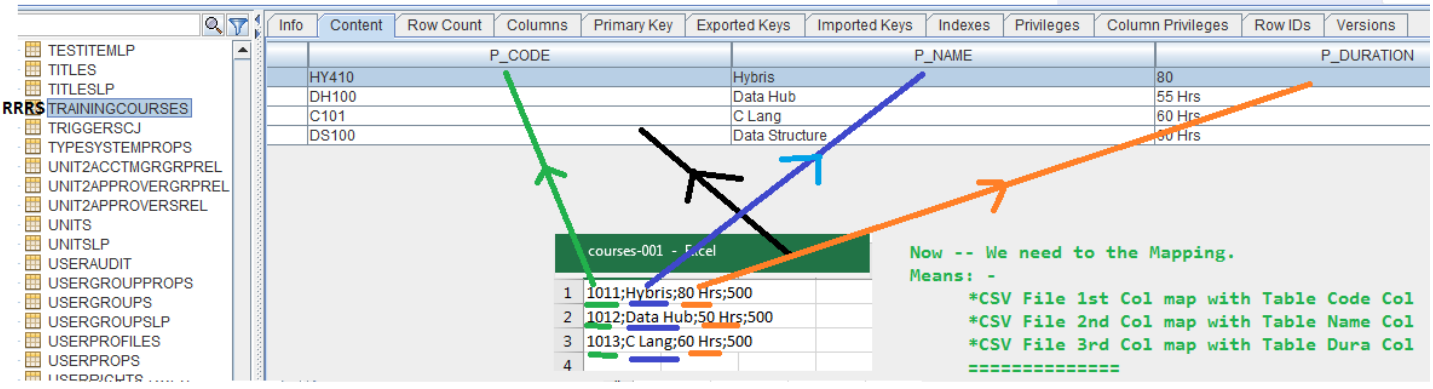
Col B data should go to “Course **Name**”

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

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



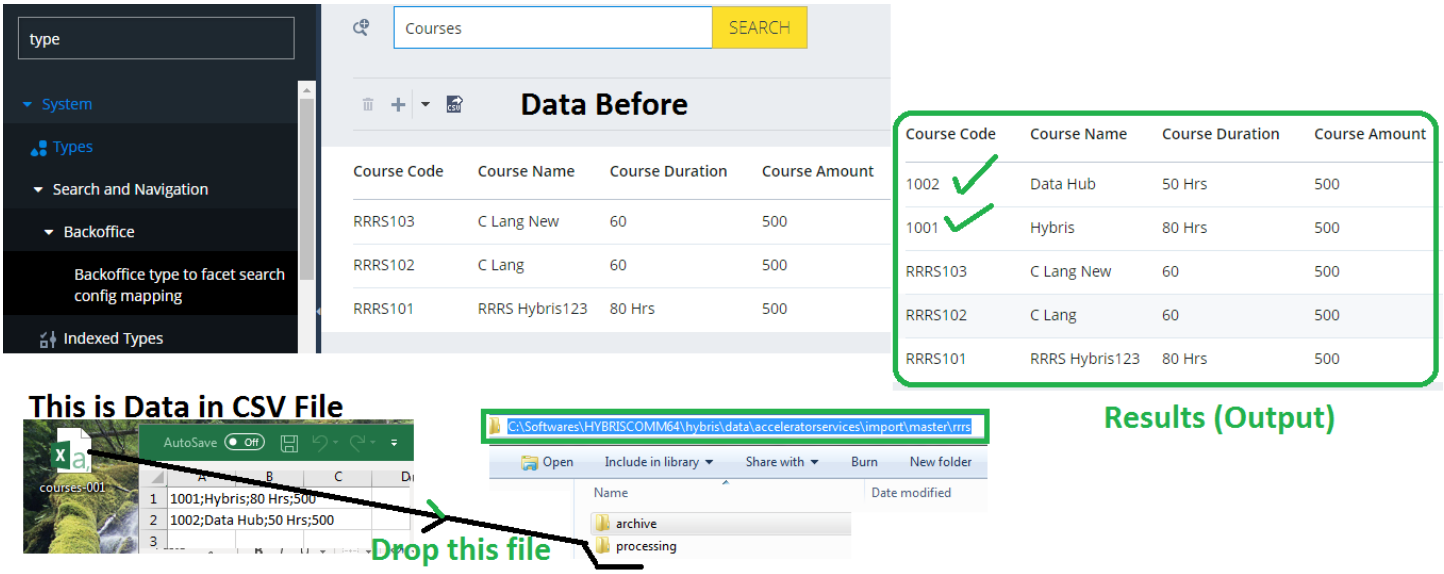
Note: - {+0} = Here + represents required field.

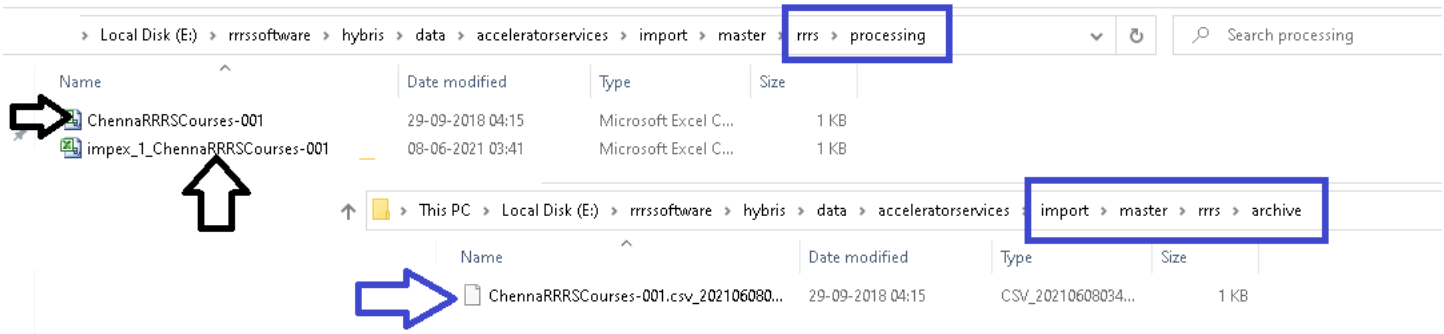


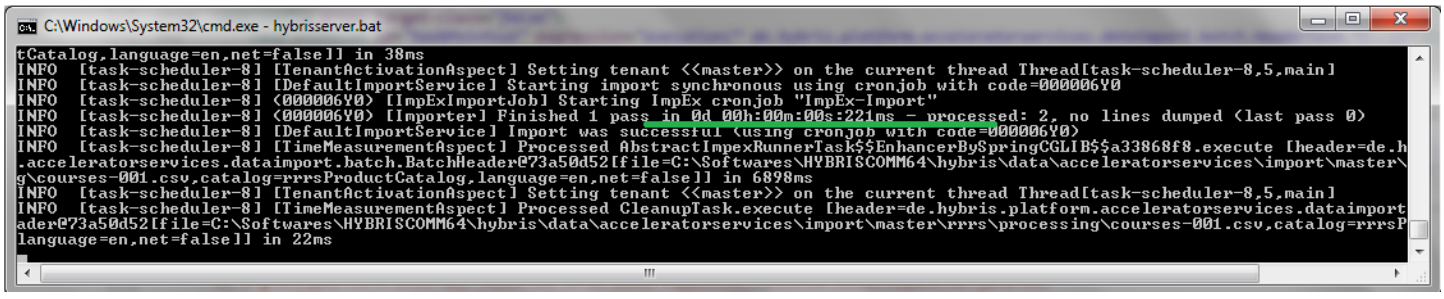
Step 6 = Do the build (**ant clean all**)

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

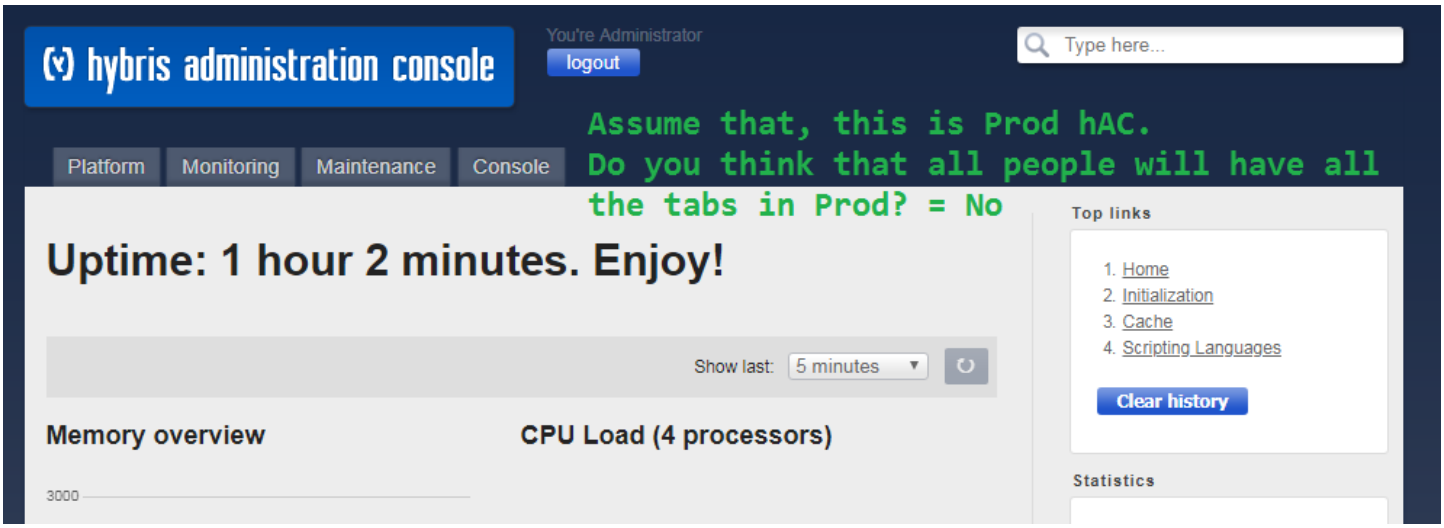
Step 8 = Test the Results



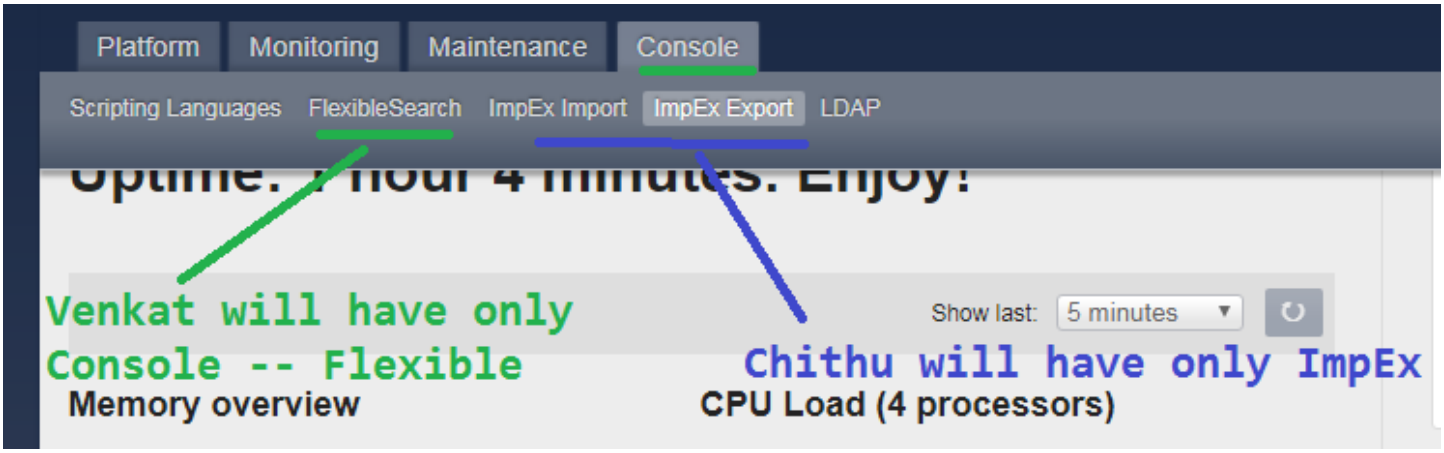




“SAP Comm” / Commerce Security & Roles Configuration



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



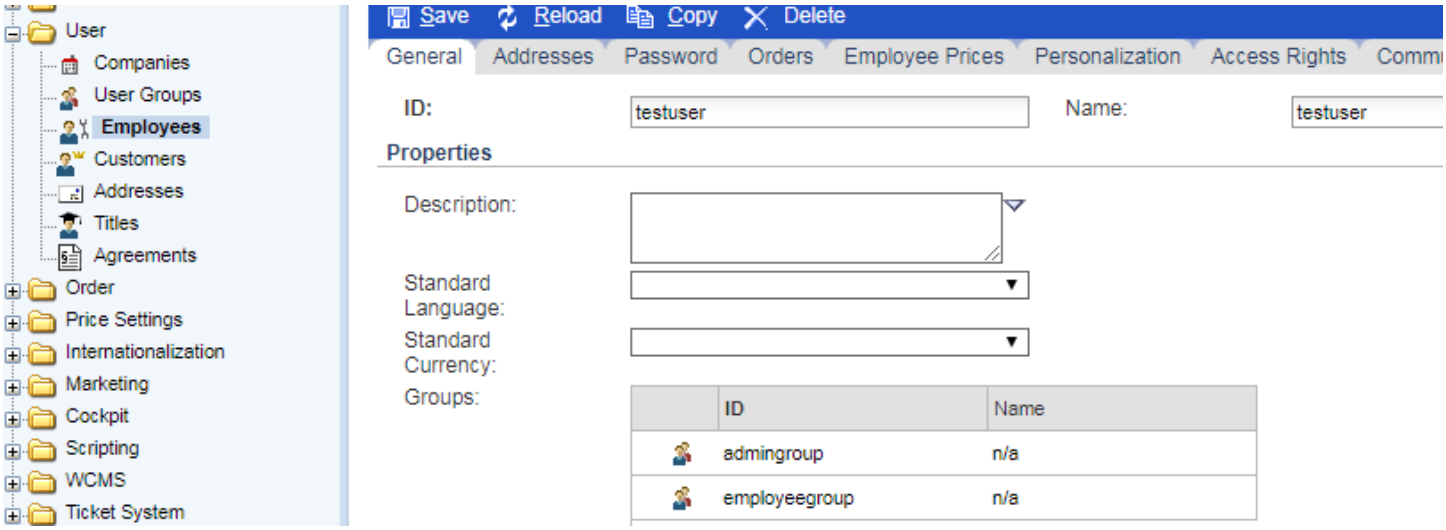
“SAP Comm” Security → based on Spring Security.

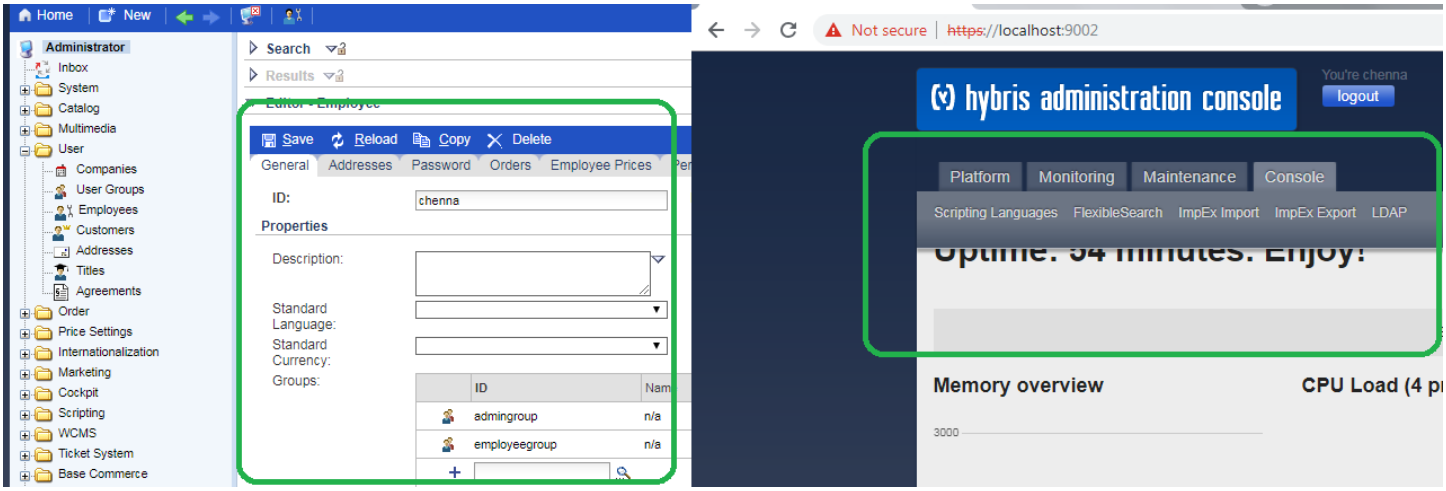
Example: - Create **1 User** & that user should have only “Console tab” & 2 Options (Scripting Language & Flexible Search).

Backoffice / hMC – Users – Employee –

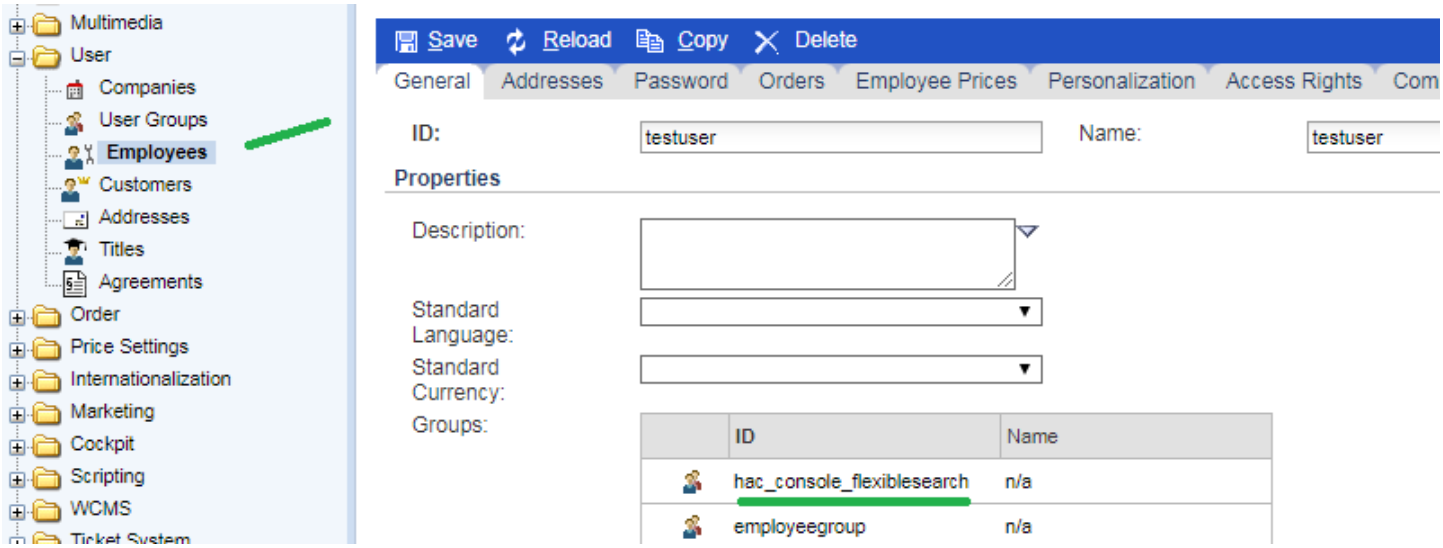
ID = testuser (or) chenna

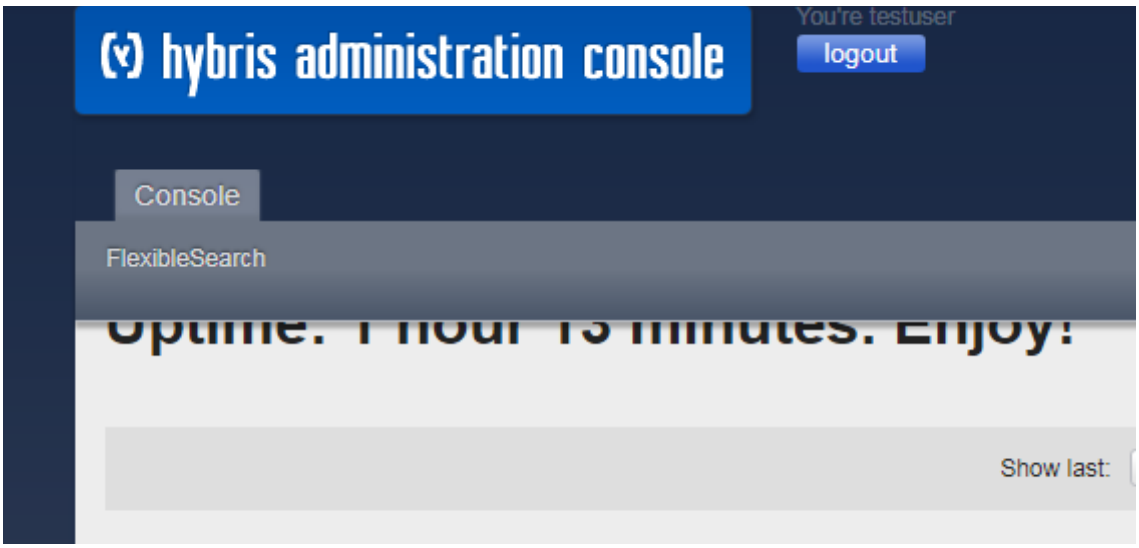
Step 1 = Let's create user with admin group.



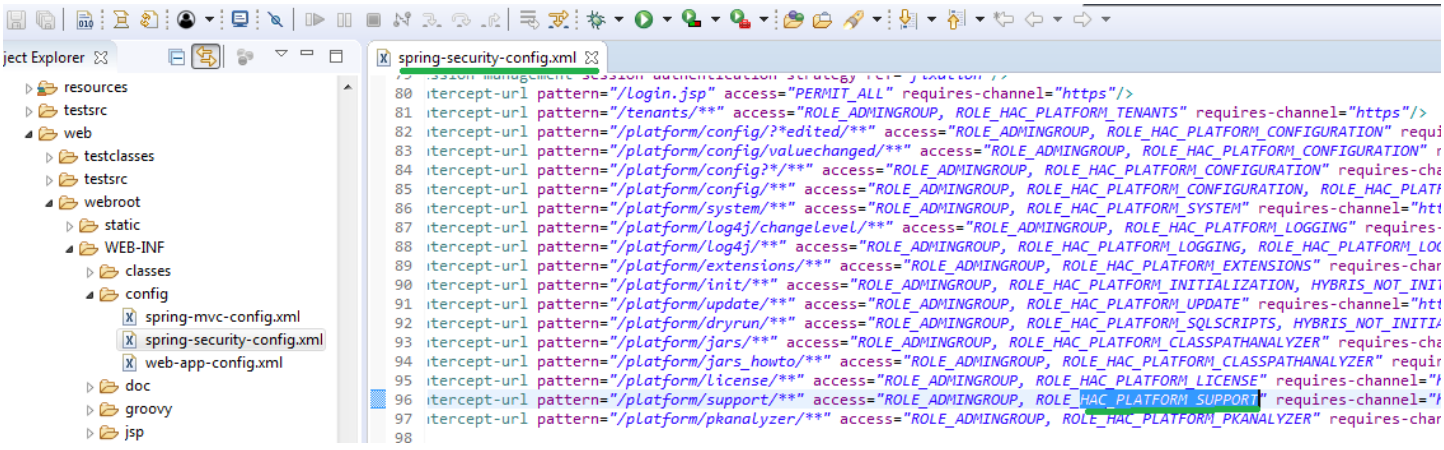


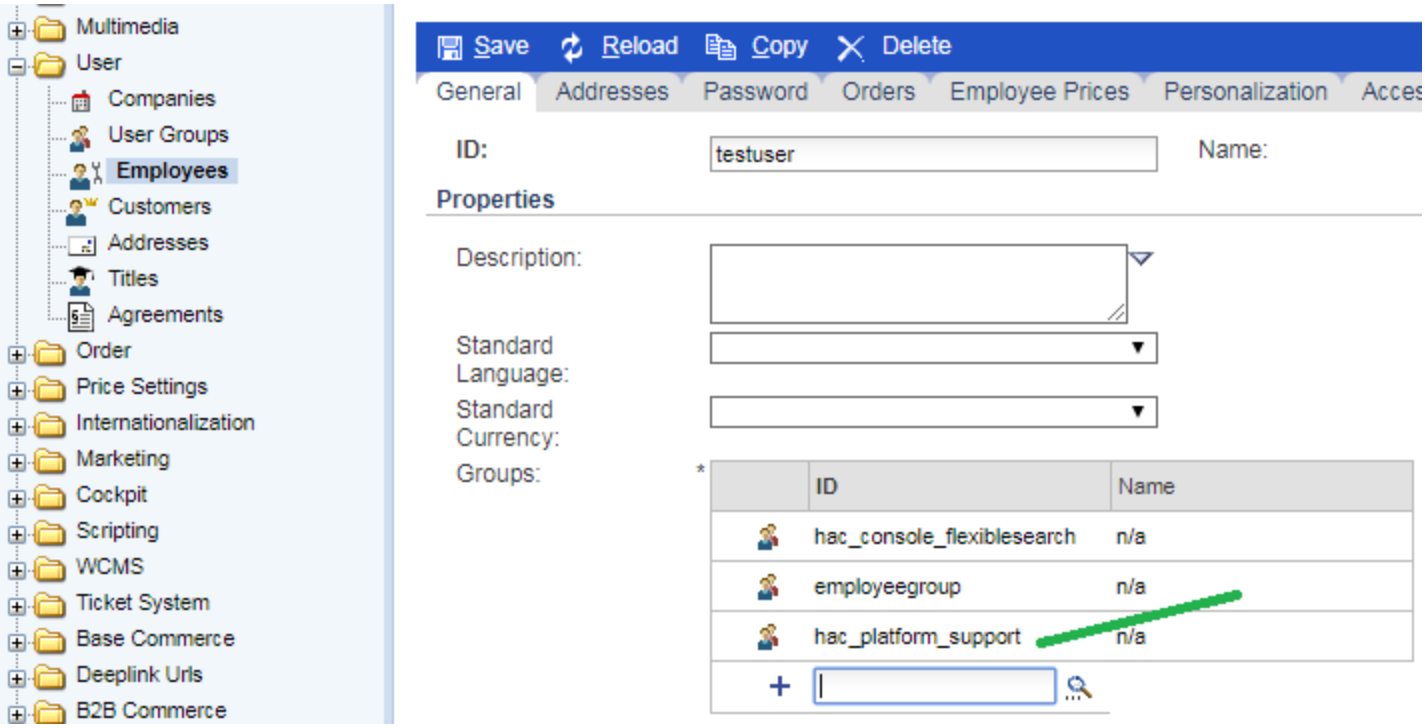
Q: How to give only "Console – FlexibleSearch" for created user (**testuser**).

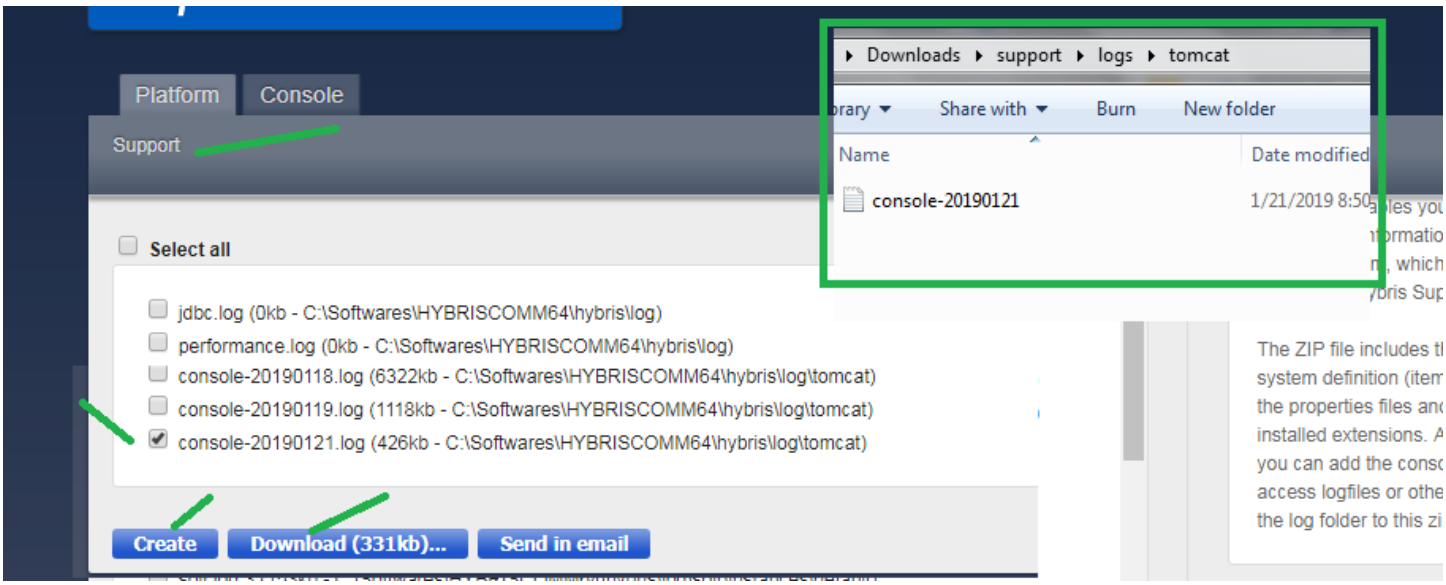




Q: How to provide **log** access (or) download log to created user (**testuser**).




Multimedia
User
Companies
User Groups
Employees
Customers
Addresses
Titles
Agreements
Order
Price Settings
Internationalization
Marketing
Cockpit
Scripting
WCMS
Ticket System
Base Commerce
Deeplink Urls
B2B Commerce
Save
Reload
Copy
Delete
General
Addresses
Password
Orders
Employee Prices
Personalization
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


Platform
Console
Support
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)
Create
Download (331kb)...
Send in email
Downloads
support
logs
tomcat
Share with
Burn
New folder
Name
Date modified
console-20190121
1/21/2019 8:50

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.

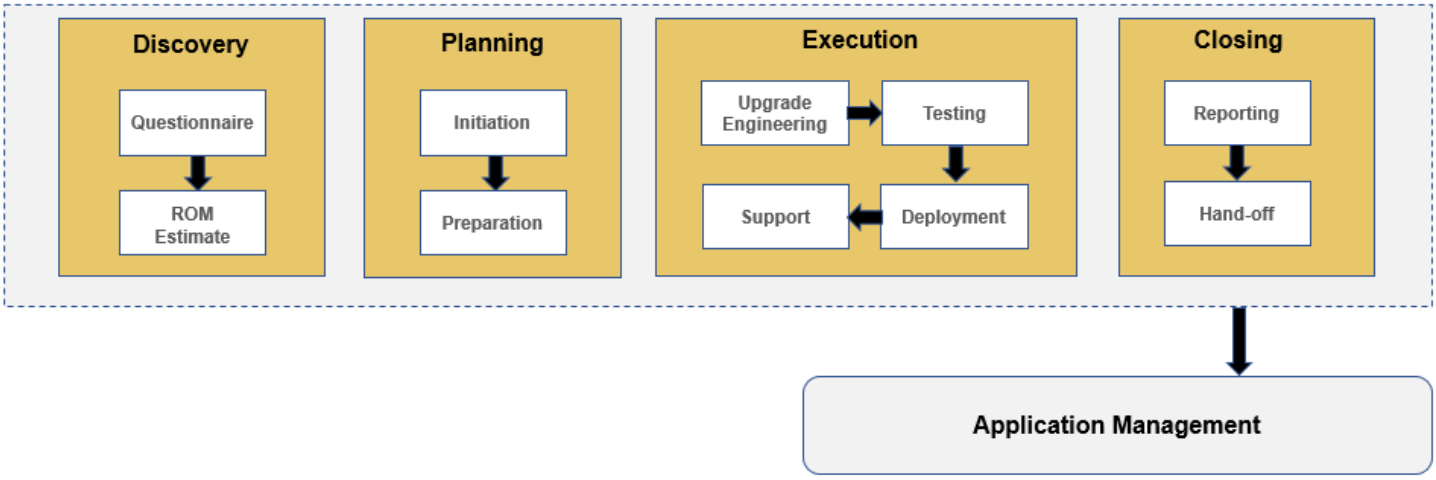
Example: - promotionsbackoffice-backoffice-config.xml, voucherbackoffice-backoffice-config.xml, cockpit-backoffice-config.xml, platformbackoffice-backoffice-config.xml etc.

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

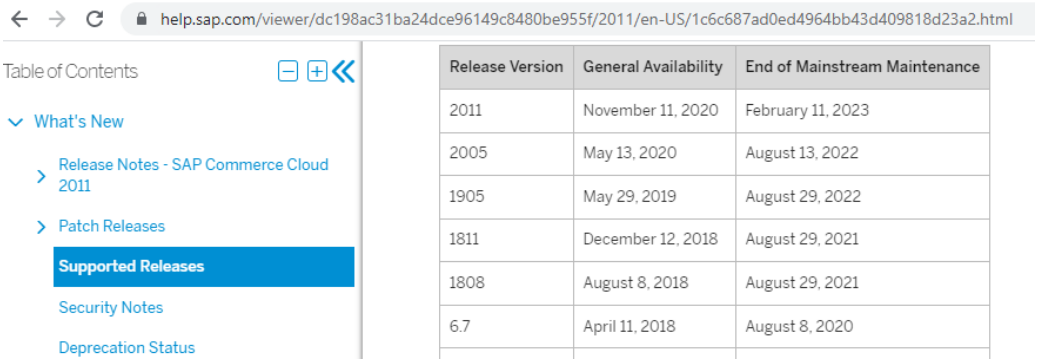
```
79      </context>
80      <context merge-by="module" type="Product" component="editor-area">
81          <editorArea:editorArea xmlns:editorArea="http://www.hybris.com/cockpitng/component/editorArea">
82              <editorArea:tab name="hmc.tab.product.properties">
83                  <editorArea:section name="hmc.product.descriptions">
84                      <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea" quali
85                  </editorArea:section>
86              </editorArea:tab>
87              <editorArea:tab name="hmc.tab.product.multimedia">
88                  <editorArea:section name="hmc.section.product.additionalmedias">
89                      <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea" quali
90                  </editorArea:section>
91              </editorArea:tab>
92              <editorArea:tab name="hmc.tab.product.stock" position="35">
93                  <editorArea:section name="hmc.tab.product.stockfinder"/>
94                      <editorArea:section name="hmc.section.warehouse.stocklevels">
95                          <editorArea:attribute xmlns="http://www.hybris.com/cockpitng/component/editorArea"
96                                                editor="de.hybris.platform.commerceservices.backoffice.
97                                                qualifier="stockLevels" label="hmc.text.product.usesear
98                              <editorArea:editor-parameter>
99                                  <editorArea:name>stockLevelSearchField</editorArea:name>
100                                 <editorArea:value>product</editorArea:value>
101                             </editorArea:editor-parameter>
102                         </editorArea:attribute>
```

Scenario = Upgrade [Example “1811 ---> 2011”]?

Software Name	1811	2011
Oracle JRE / JDK	Java 8	Java 11
SAP JVM	8.1	SapMachine 11
Apache Tomcat	8.5.57	8.5.56, bundled
Database	HANA 1.00 SPS12, HANA 2.0	HANA 1.00 SPS12, HANA 2.0
Solr	7.7	8.6
Browsers	IE 10, 11, MS Edge & Other latest	Microsoft Edge
Commerce Directory Structure	bin [platform, **Exts** & custom]	bin [platform, **modules** & custom]
Recipes	b2c	cx
Spring [aop, aspects, beans, context]	4.3.21	5.2.9
Spring Security [Config, Core]	4.2.6	5.2.3
Backoffice changes	ABC has good amount of changes	Cross check those with steps
Spring oAuth2	2.2.0	2.3.6
Site Responsive Impact	Yes	We need to verify



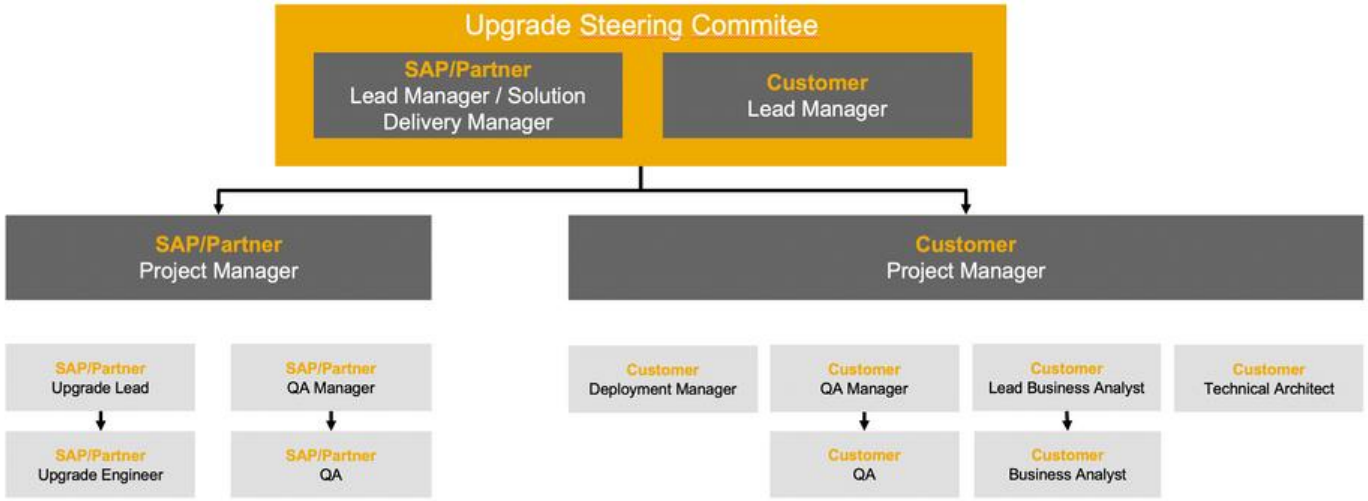
Q = How to find “End of Mainstream Maintenance”?



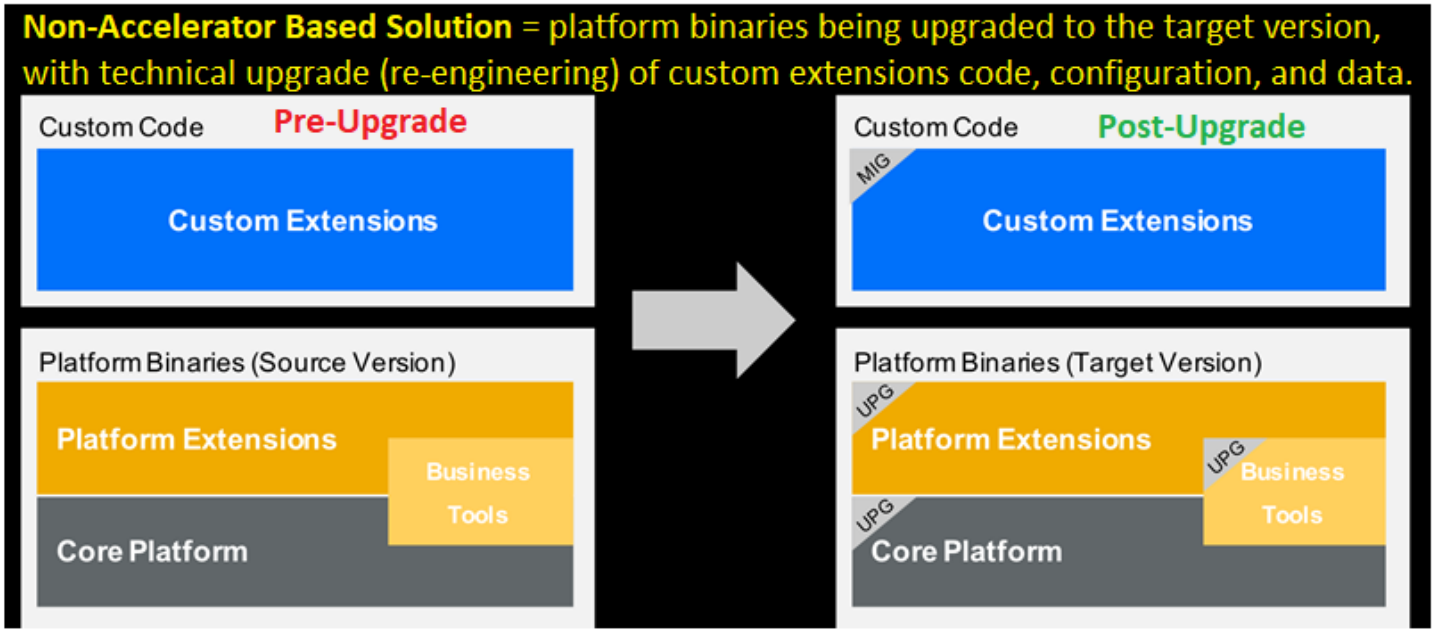
Release Version	General Availability	End of Mainstream Maintenance
2011	November 11, 2020	February 11, 2023
2005	May 13, 2020	August 13, 2022
1905	May 29, 2019	August 29, 2022
1811	December 12, 2018	August 29, 2021
1808	August 8, 2018	August 29, 2021
6.7	April 11, 2018	August 8, 2020

Upgrade System Doc =

https://www.sap.com/cxworks/article/441185299/upgrading_your_sap_commerce_solution

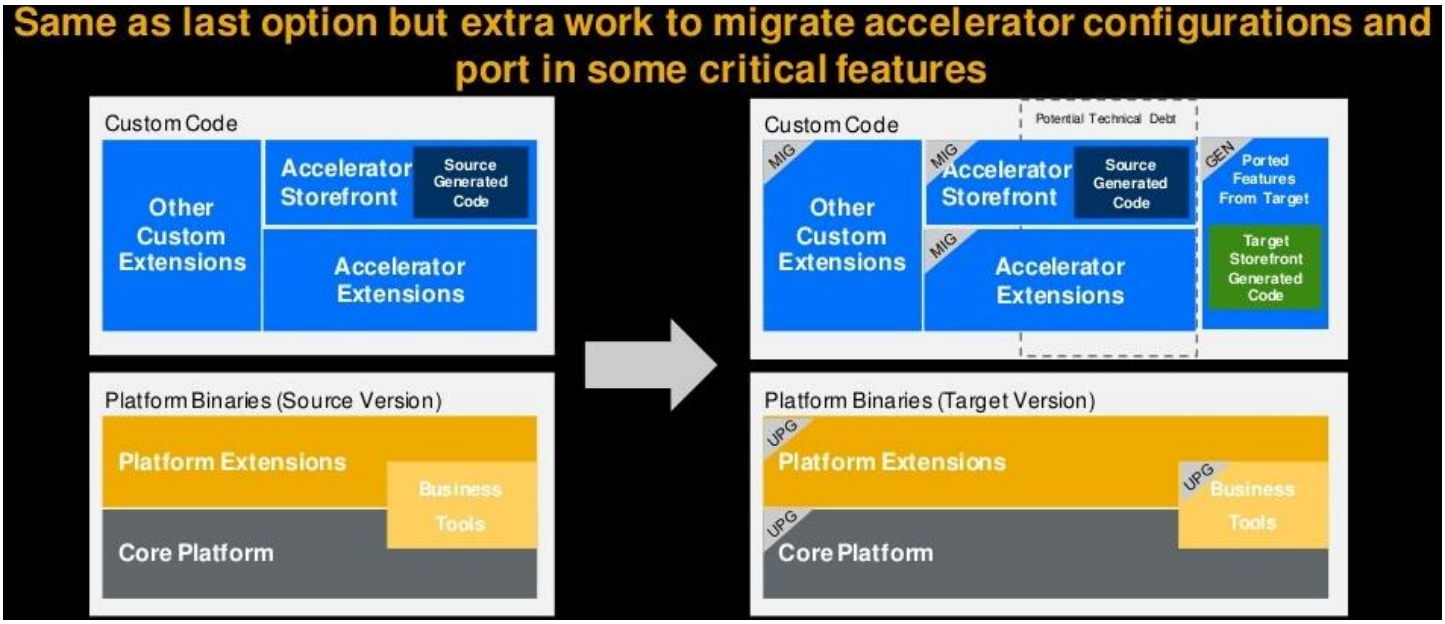


Non-Accelerator Based Solution =



	Upgraded?	Technically Migrated?	Comments
Core Platform	Yes	No	N/A
Platform extensions	Yes	No	N/A
Business Tools	Yes	No	N/A
Business Tools customizations and custom configuration	No	Yes	When customer build additional configuration (implementation) in hMC or Cockpits to provide custom business logic.
Custom extensions	No	Yes	Will probably require changes to your custom code to ensure it can compile, system can start, and upgrade running system can run without errors.

Accelerator Based Solution =



Q = Software architecture impacts?

1) Third party compatibility

2) Leverage Commerce APIs

3) Move to Spartacus

Q = Upgrade plan estimates?

Resource Model 50/50 Onshore-Offshore

Role	Month 1	Month 2	Month 3	Month 4
Project Manager	0.25	0.25	0.25	0.25
Architect	0.25	0.25	0.25	0.25
Team / Technical Lead	1.00	1.00	1.00	1.00
Functional / Tester	1.00	1.00	1.00	1.00
Sr Comm Developer	1.00	1.00	1.00	1.00
Comm / UI Developer	1.00	1.00	1.00	1.00
Total	4.5	4.5	4.5	4.5

DRAFT

Assessment

Role	Month 0
Project Manager	0.50
Architect	1.00
Team / Technical Lead	1.00
Functional / Tester	1.00
UI / Comm Developer	1.00
Total	4.5

DRAFT

Core Platform	Platform core functionality. This is what is found in the hybris/bin/platform directory
Platform extensions	Platform core extensions usually placed in hybris/bin/ext-(...) . These extensions provide extensive functionality and are built upon the core platform
Business Tools	The tools that are used for performing business functionality. These are comprised of SAP Administration Console, hMC, backoffice and cockpits
Custom extensions	Any extension that changes or extends behavior of the platform, usually placed in hybris/bin/custom
Platform Template	Standard templates that can be used to accelerate development. These include B2C/B2B accelerator, Telco etc.
Template Customizations	Any customization that extends/overwrites the behavior of the generated code from the platform template

Q = Explain Technical Approach Steps?

1) Regression Testing [Unit, Integration, UI, Security, Performance]
2) Update Deployment Procedure [update the type system]
3) Business Data Technical Migration
4) Configuration Technical Migration
    a. local.properties
    b. localextensions.xml
    c. buildcallback.xml
    d. customize folder [hybris/config/customize]
    e. Tomcat and TcServer Configuration
5) Data Model Changes
6) Custom Code Technical Migration
    a. Java Code [hybris/bin/custom]. There are several API changes: -
        i. Method signature has changed
        ii. Method name has changed
        iii. Method was moved to another class
        iv. Method is deprecated or was removed
        v. Method visibility has changed
        vi. Interface has additional method declarations
        vii. Class was renamed
        viii. Class location (package) has changed
        ix. Class is deprecated or was removed
        x. Functional module (set of interfaces, classes or methods) was re-engineered (refactored) between source & target.
    b. Spring Context
    c. Third Party Libraries
    d. Integrations
    e. Performance fixes
    f. Security

Upgrade high level procedure (or) steps =

Step 1 = Download, unzip, and build the new distribution

➢ Download the SAP Comm Platform "**YYMM** [Example - 2011]" distribution and all the needed modules.

➢ There are two installation ZIP file

- CXCOM**200500**P_0-70004955.ZIP = Contains all the files to install the SAP Hybris Commerce and run it out-of-the-box
- HYBRISCRMINT00P_0-70005781.ZIP = This file contains the "SAP Comm" CRM Integration dependencies

You must download and extract the contents of the HYBRISCRMINT00P_0-70005781.ZIP file into the same directory to which you extract the SAP Commerce CXCOM**200500**P_0-70004955.ZIP file

➢ Extract all the downloaded ZIP files to a new directory, where the Target version system will be installed

➢ Run **ant clean** / **ant clean all** on the new system. This will create the config folder in the new instance of Hybris

Step 2 = Copy (and adjust if needed) your custom extensions (${HYBRIS_BIN_DIR}/custom) from your **old** instance of SAP Hybris Commerce to the **new** instance of SAP Commerce

Step 3 = Copy and adjust the configuration files

➢ From your **old** Platform location copy your **local.properties** and **localextensions.xml** configuration files to the **new** config folder (${HYBRIS_CONFIG_DIR})

➢ Copy any 3rd party database drivers if you using from ${HYBRIS_BIN_DIR}/**platform/lib/dbdriver** folder to **new** instance location

➢ If necessary, edit the **localextensions.xml** file in order to add additional extensions (for example the ones which are new in the release)

➢ If necessary, adjust the local.properties file

Step 4 = Copy data (${HYBRIS_DATA_DIR}/**hsqldb** folder)

Step 5 = Copy media files (media data files are located in ${HYBRIS_DATA_DIR}/media)

Step 6 = Reinitialize Solr Facet Search indexes.

Step 7 = Build "SAP Comm" platform

• Go to ${HYBRIS_BIN_DIR}/platform directory & execute setantenv.bat on Windows or. ./setantenv.sh on Linux.

• Execute **ant all & wait** for the build to complete

• Run ant **update system (or) "hAC – Platform – Update"**

Step 8 = "ant clean all" – Fix If any challenges.

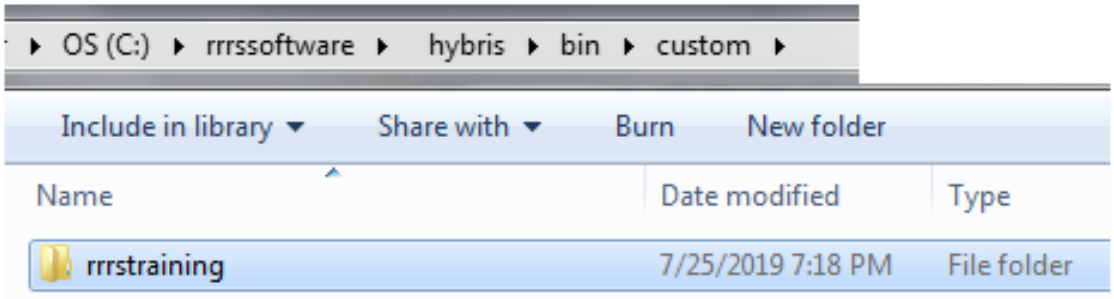
Start the server – Fix if any challenges.

Test the site – Fix if any challenges.

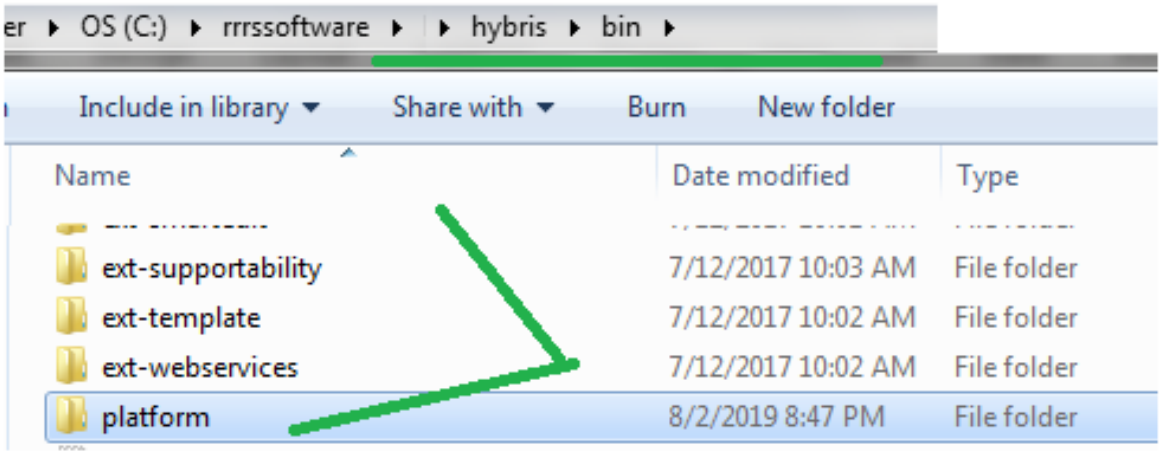
Scenario = Ant Customize

In SAP Comm, we have 2 types of files & directories: -

1) Custom Files = These we change based on client requirements.

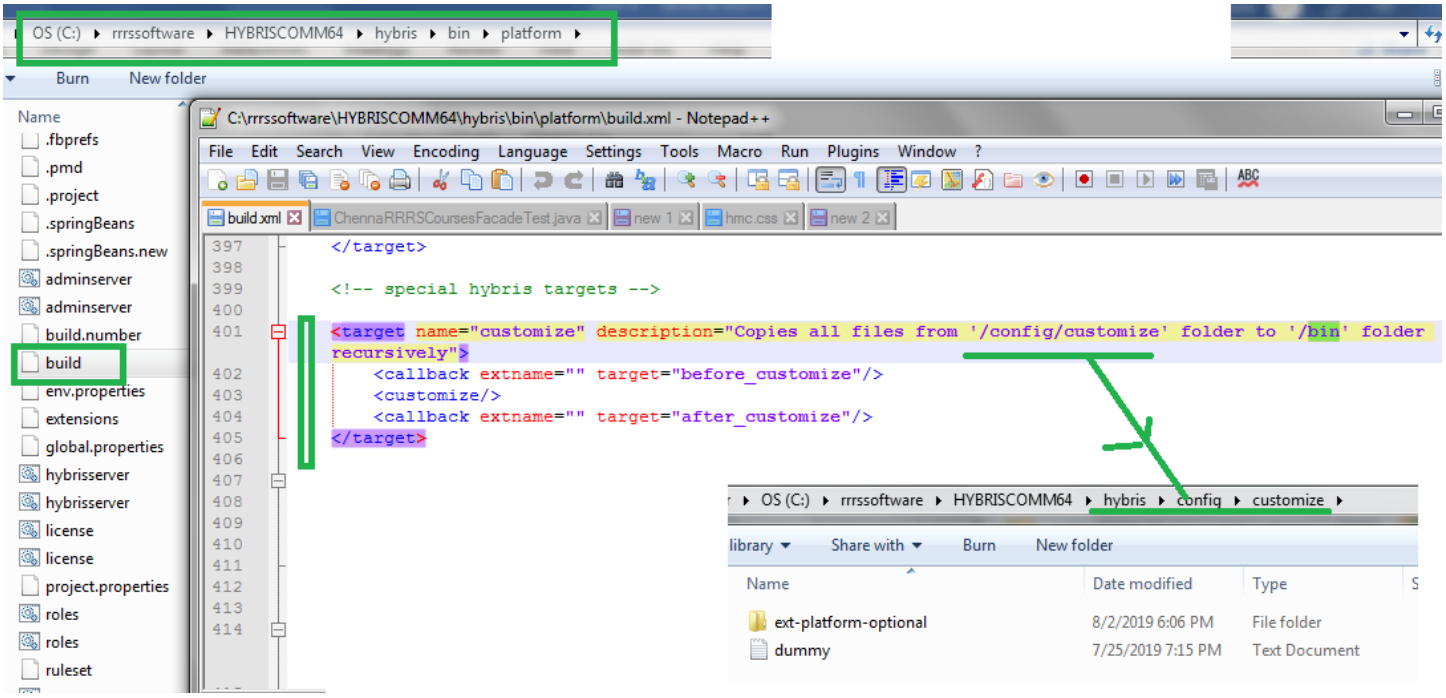


2) Read only files (or) Binary files = These generally we don't change directly.

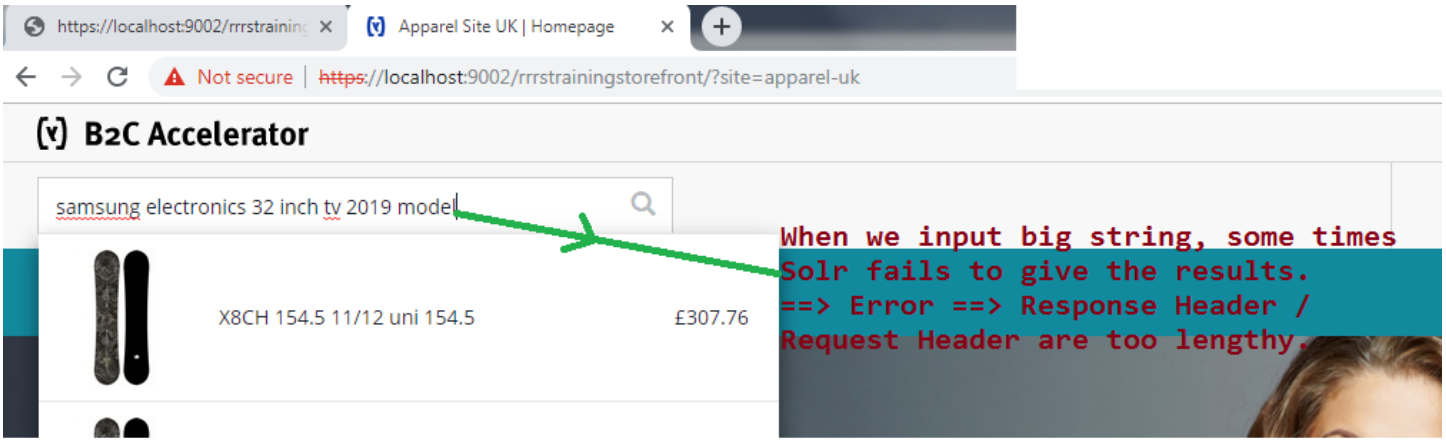


Note = Sometimes, we may need to modify these binary files (or) xml files (or) standard files (or) CSS files (or) ….

===> To do all above… We can use **ant customize**



Requirement 1 =

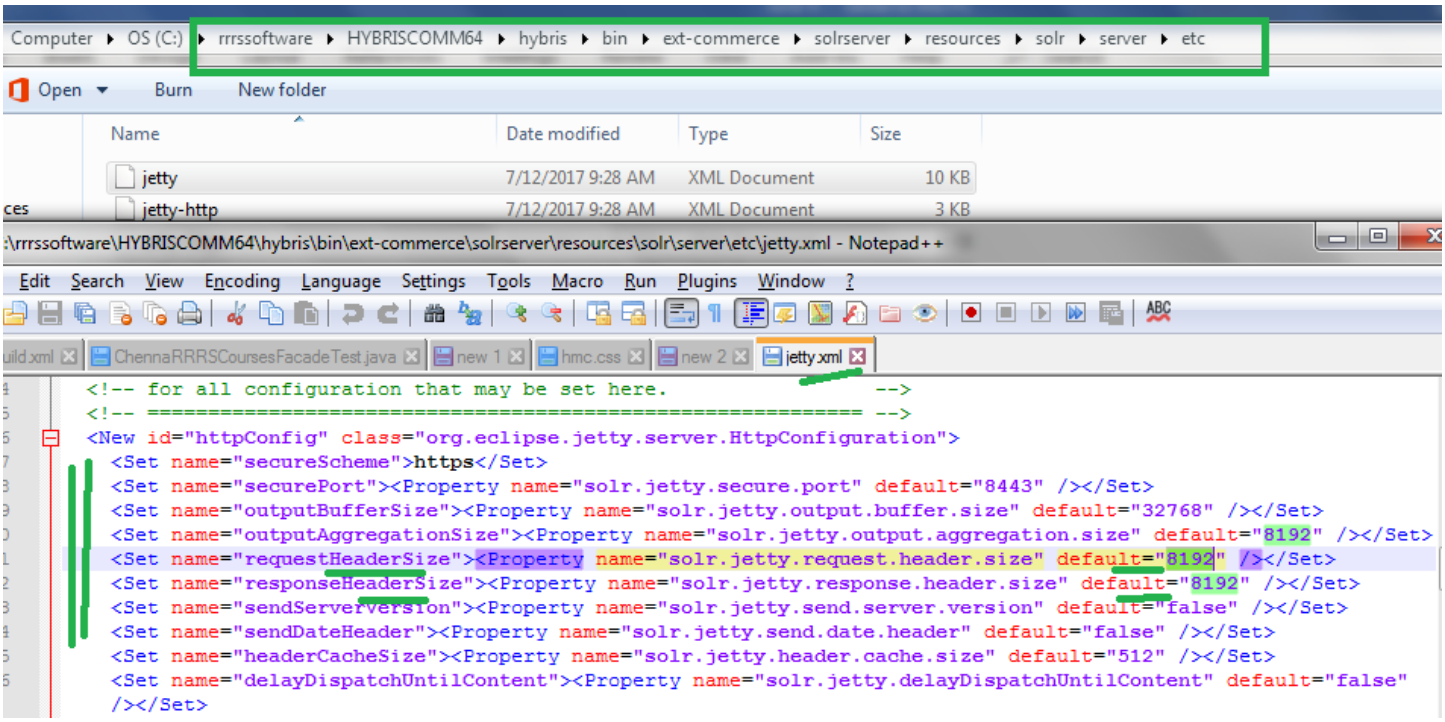


Sometimes fails to get the data when we input more characters in search box.

=== Error === Request header / Response header are too lengthy.

Q: How to fix above?

Solution =



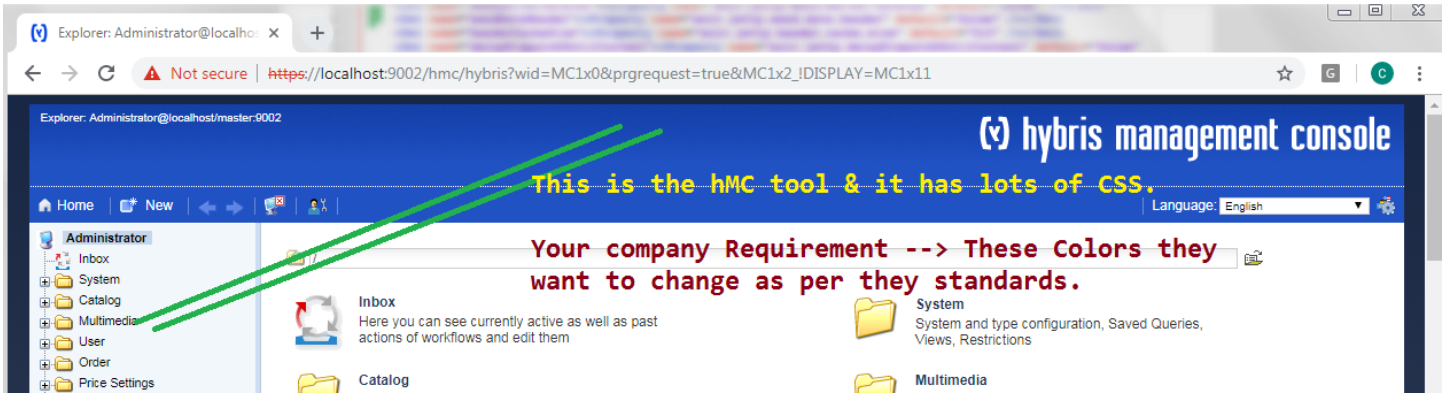
Challenge → Above is the standard xml file available in bin folder.

Generally, we don't modify anything available within \bin (or) \platform folder.

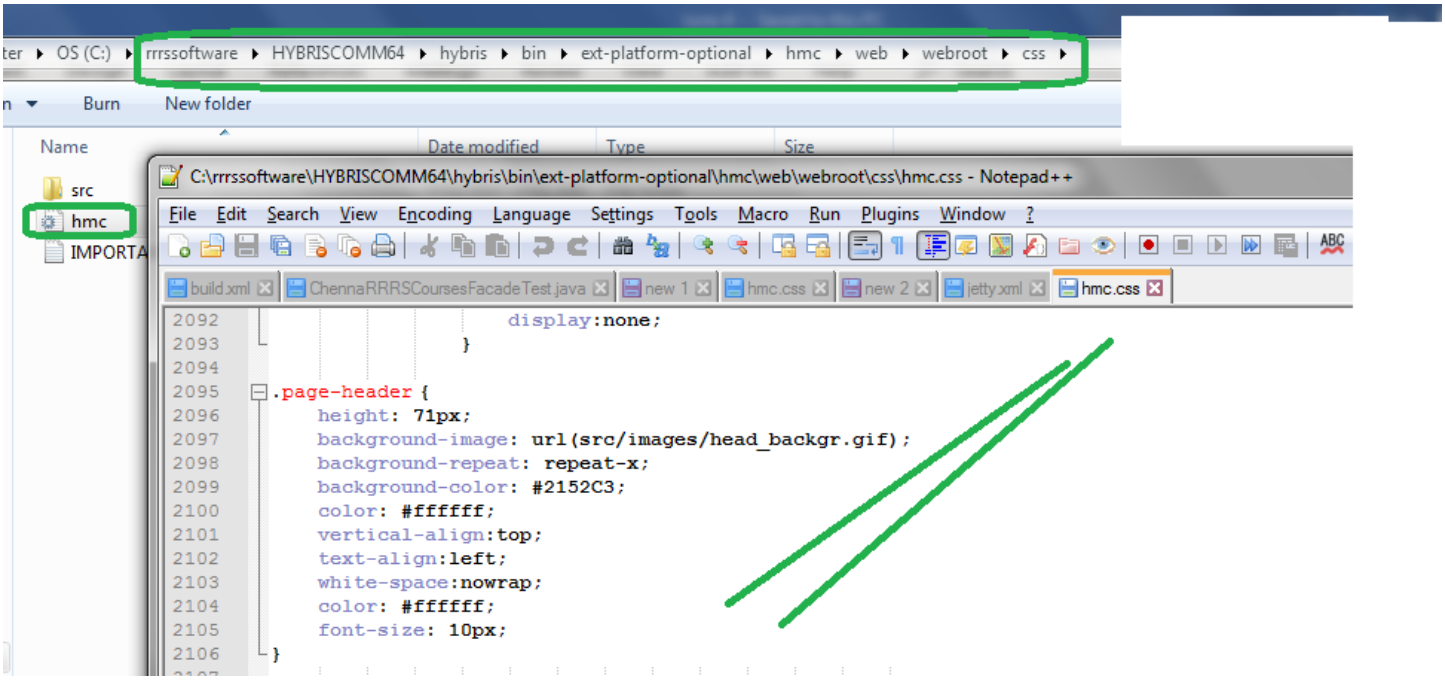
===> Bcoz, your changes may lose when upgrade happens.

Best Solution = Use "**ant customize**" for this.

Requirement 2 =



Q = Where hmc standard CSS file is available?



Challenge → Above is the standard CSS file available in bin folder.

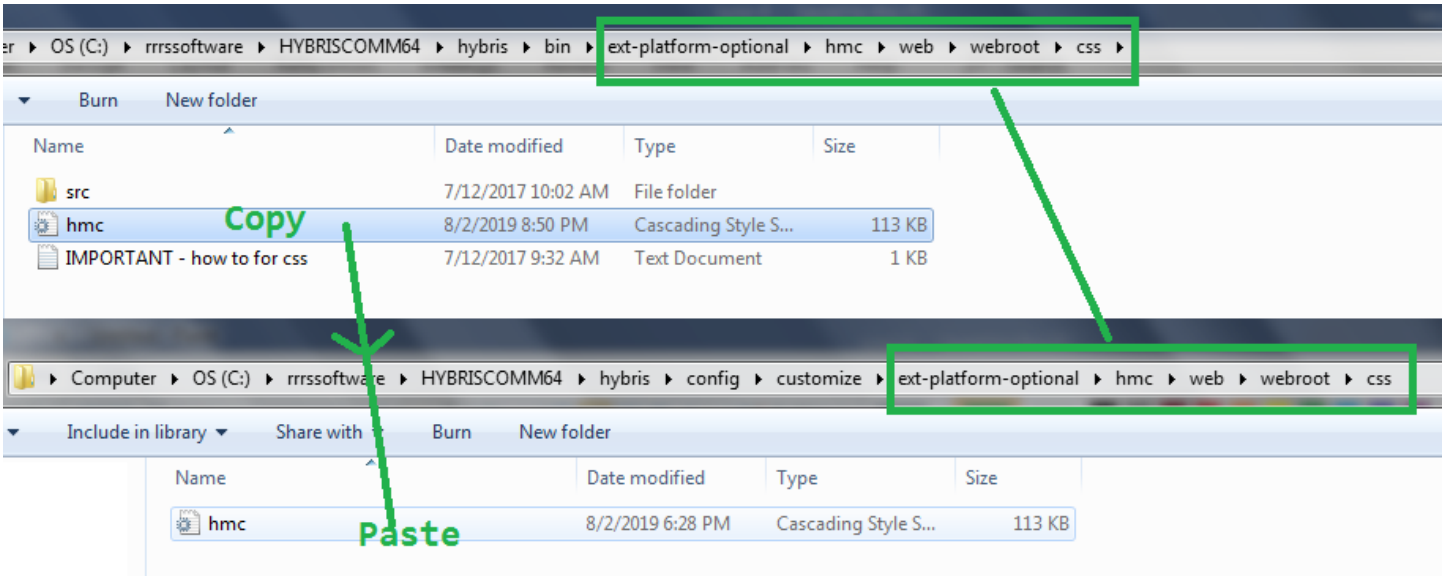
Generally, we don't modify anything available within \bin (or) \platform folder

===> Bcoz, your changes may lose when upgrade happens.

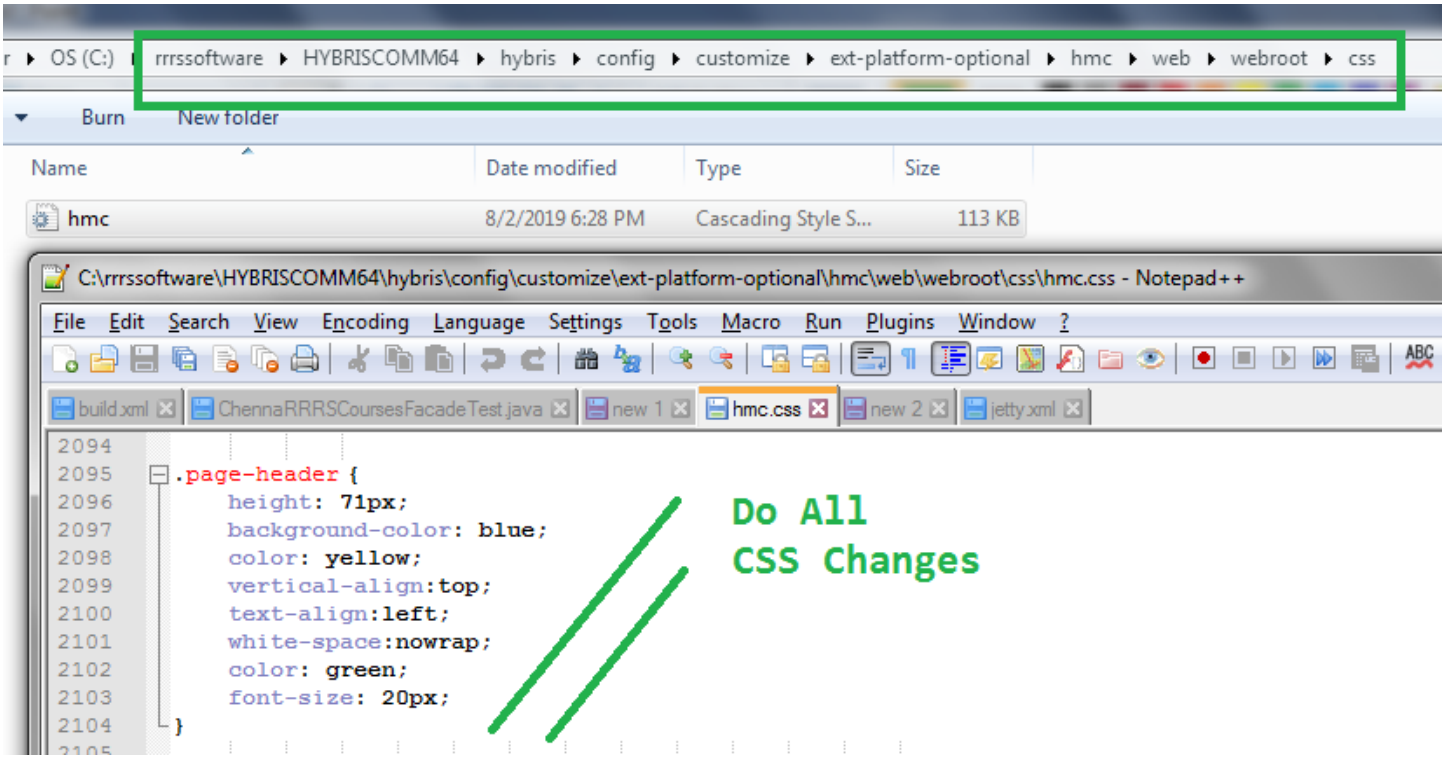
Best Solution = Use "**ant customize**" for this.

How to use “ant customize”?

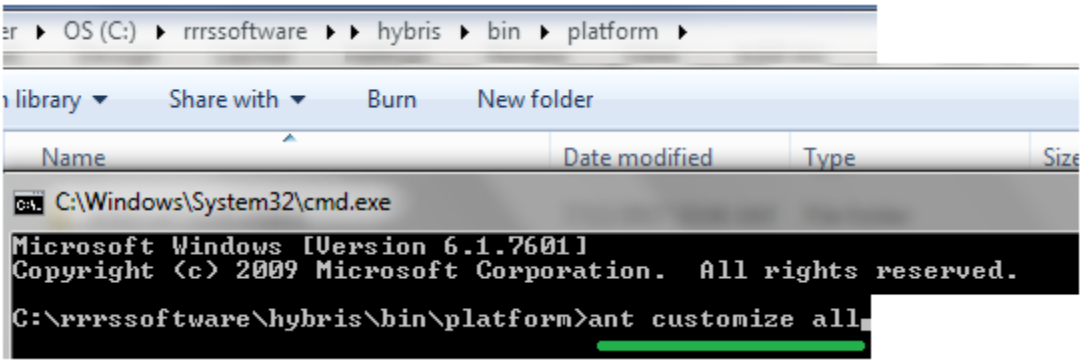
Step 1 = Copy hmc.css file from bin & paste in custom.



Step 2 = Do the CSS changes

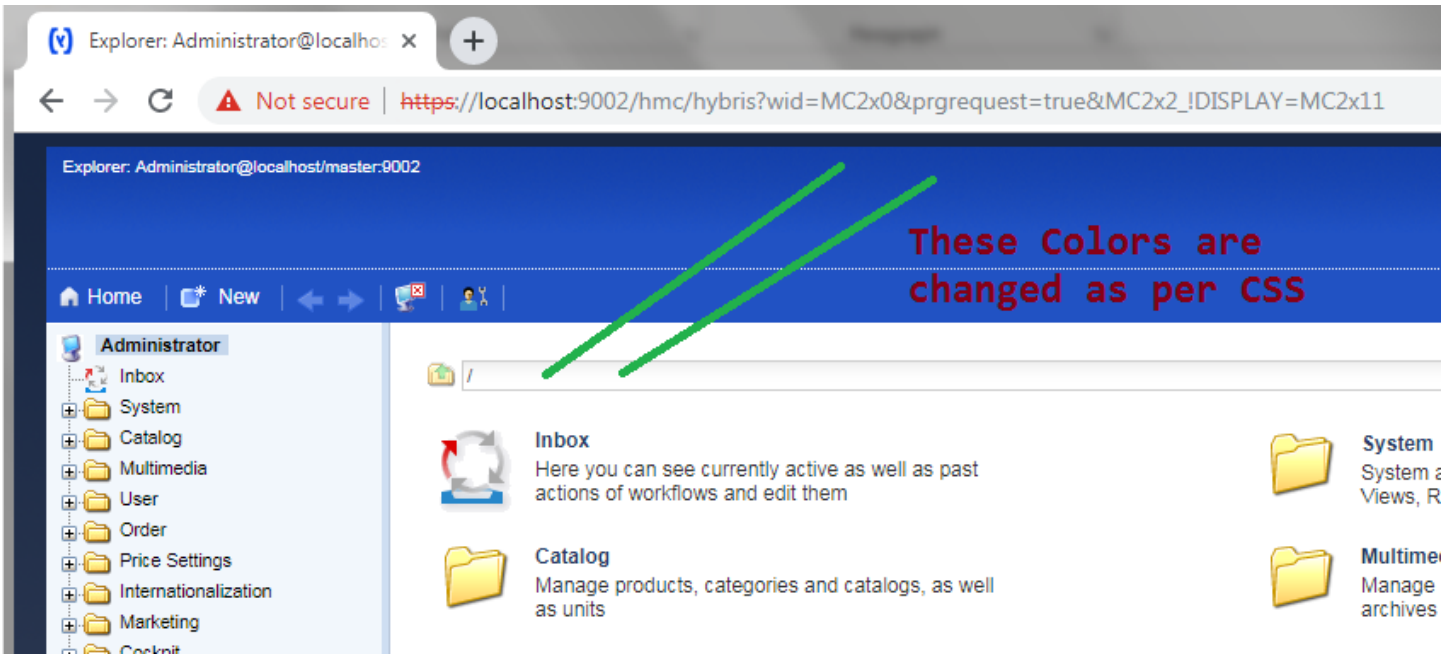


Step 3 = Do "ant customize" (or) "ant customize all" (or) "ant clean customize all"

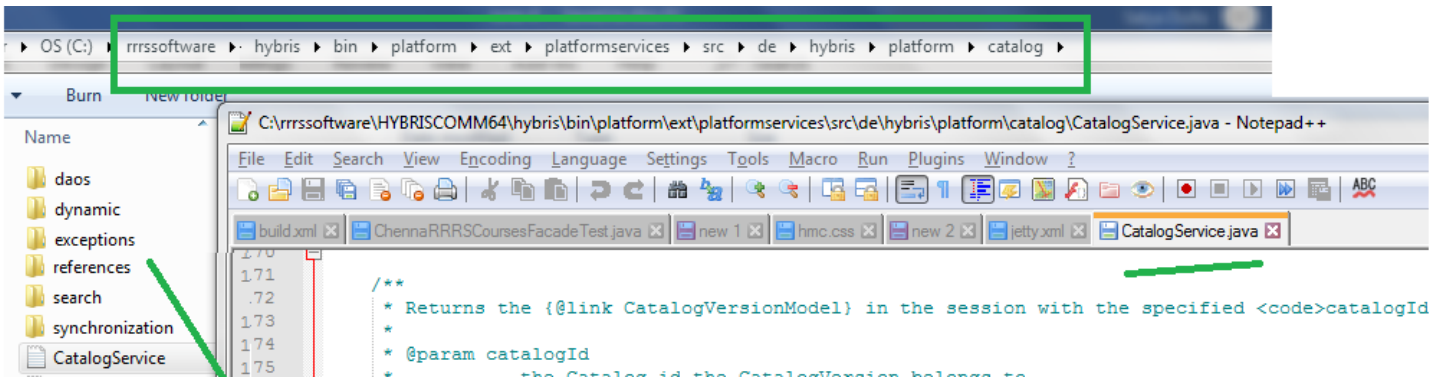


Note: - This replaces the customized files with original files.

Step 4 = Do the build … Start the Server ... Test the Results.



Another requirement [3] =

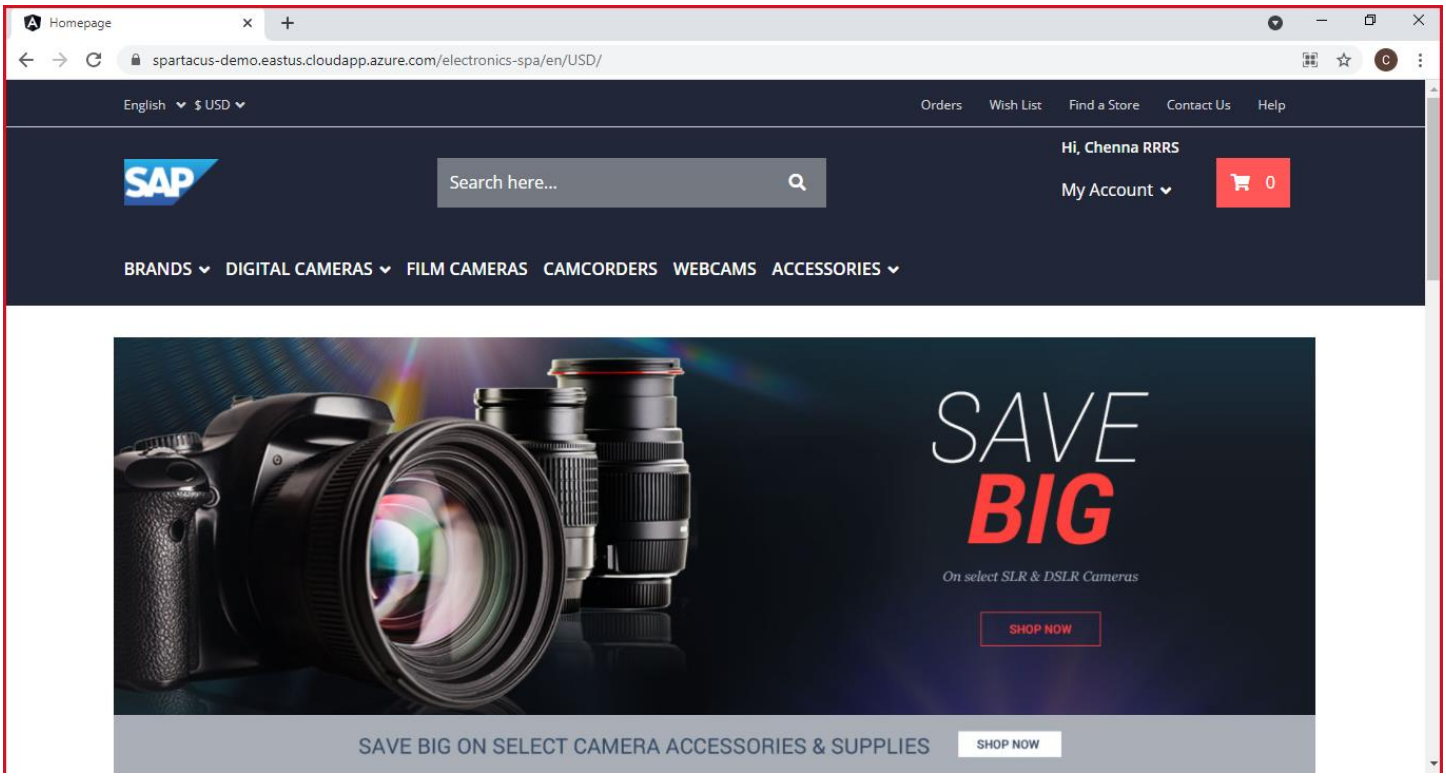


Spartacus =

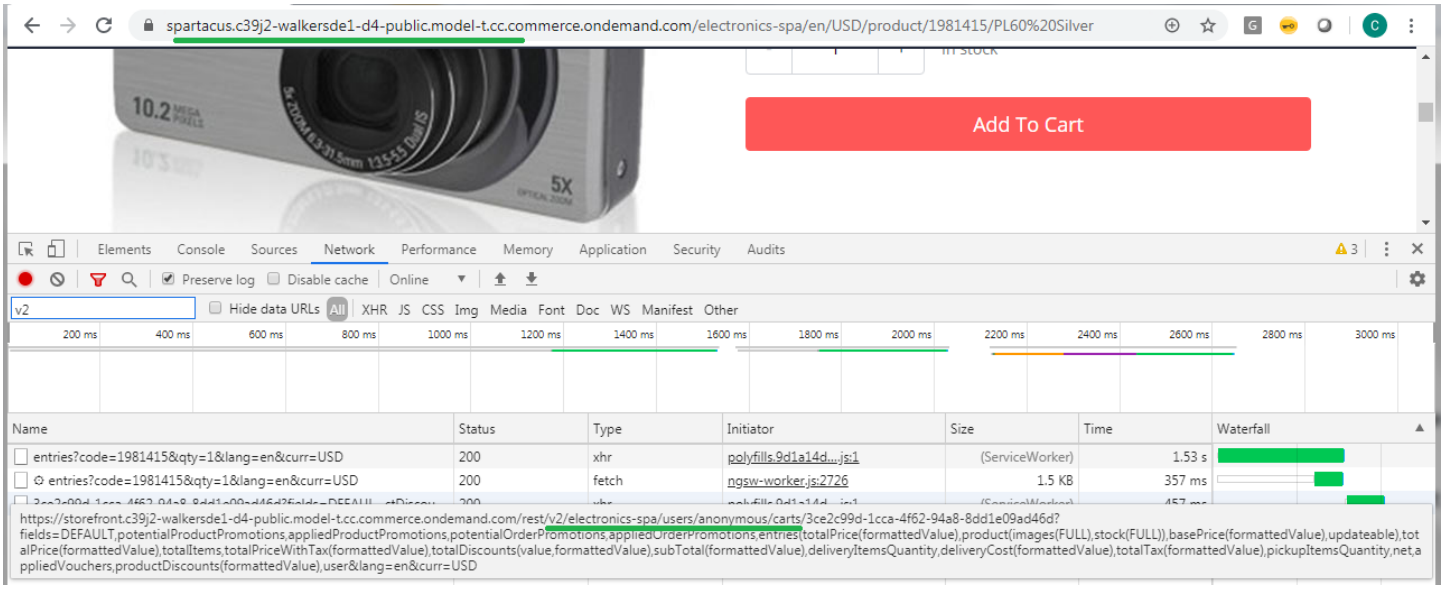
Docs = https://sap.github.io/spartacus-docs/

Demo Site URL =

https://spartacus-demo.eastus.cloudapp.azure.com/electronics-spa/en/USD/

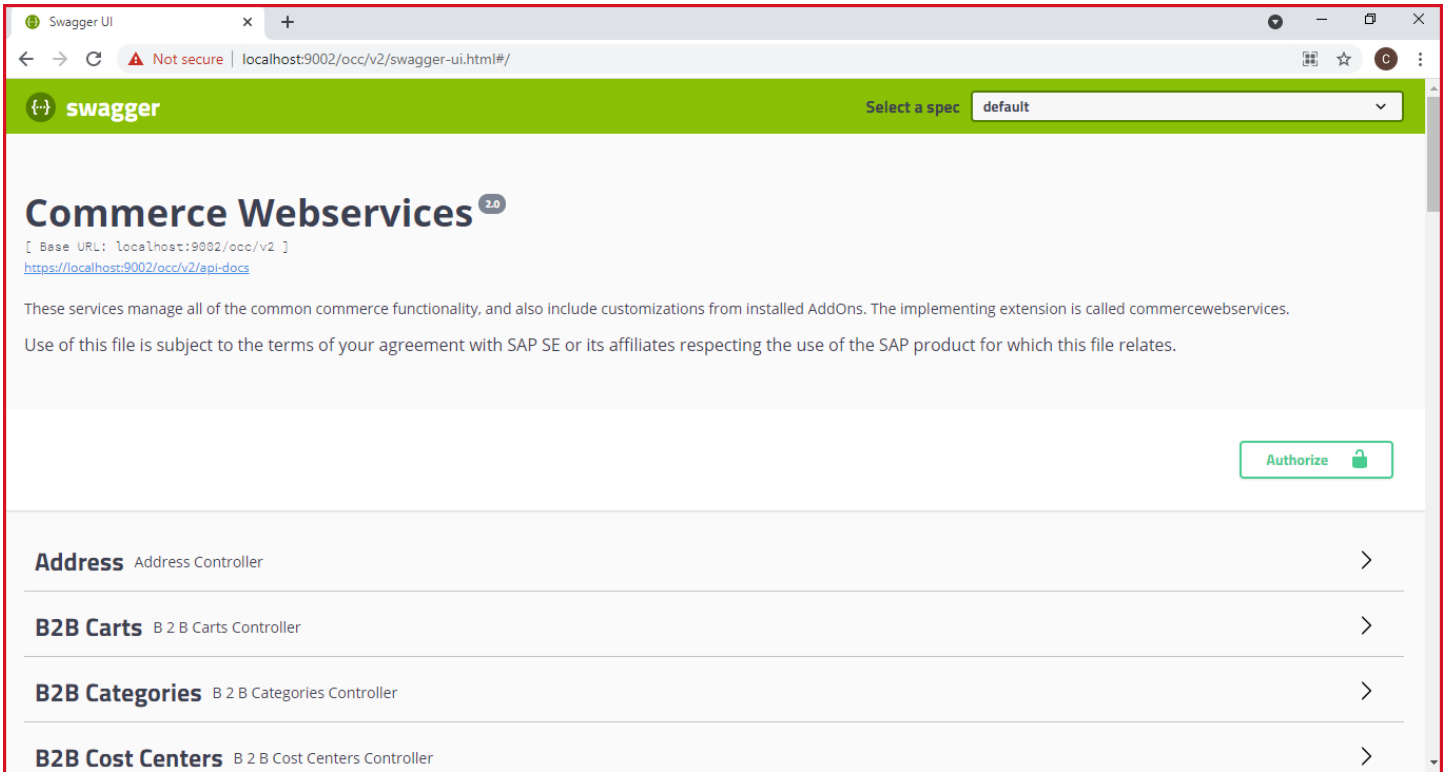


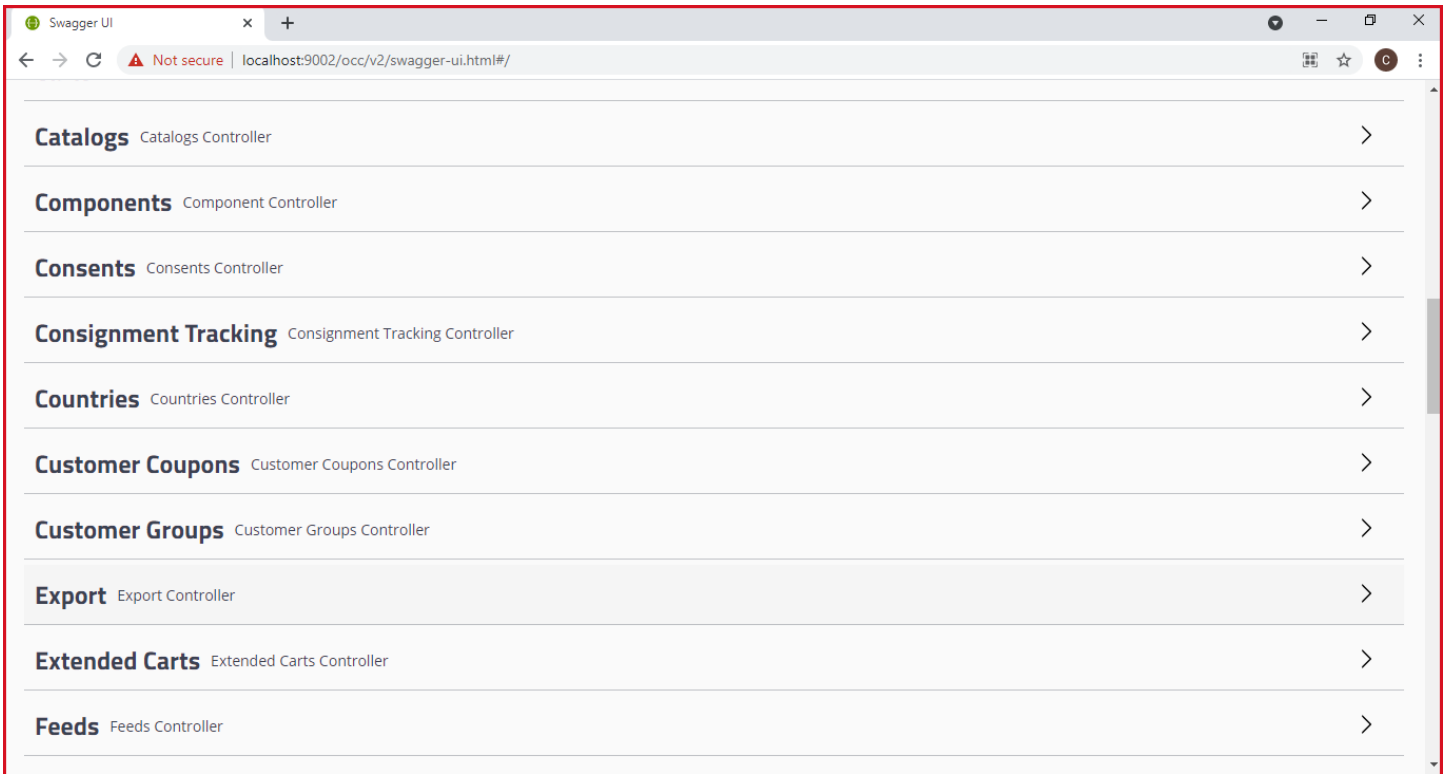
Note: - Spartacus interact with “SAP Comm” through OCC **V2**. Example: -



OCC Services =

URL = https://localhost:9002/occ/v2/swagger-ui.html#/





Scenario = B2B

Q = Explain Organizational Structure in "SAP SD"?

- **1st Level** = Organization (IBM)

  IBM Producing Plants (SAP MM) = TX, MI / ...

  - **2nd Level** = Sales Org (From where you get Products – TX / MI / …) = SAP SD
    - **3rd Level** = Distribution Channel (TX – Irving, TX – Dallas / …)
      - **4th Level** = Division
        - **5th Level** = Sales Office
          - **6th Level** = Sales Group

Note: - Sales Area (SA) = Sales Org + Distribution Channel + Division

Q = What is Business Partner?

Parties involved in a business transaction (Customer, Carrier, Employee and …). There are 4 default business partner parties (Payer, Sold-To, Ship-To and Bill-To).

Q = Explain SAP Partner Functions?

It is roles & responsibility a business partner can take on in a business Tr.

- We have 2 types of Master Data: -
  - **Vendor** (Who Provides products) = Material Management Side

    Verizon get products from Office Depo (Vendor).

In Office Depo System – Whom to send Invoice in Verizon = **BillTo** (AP Dept).

In Office Depo System – Whom to Ship in Verizon = **ShipTo** (Front Office).

o **Customer** (Who take your product) = Sales Side

Verizon sells Fios to CCN.

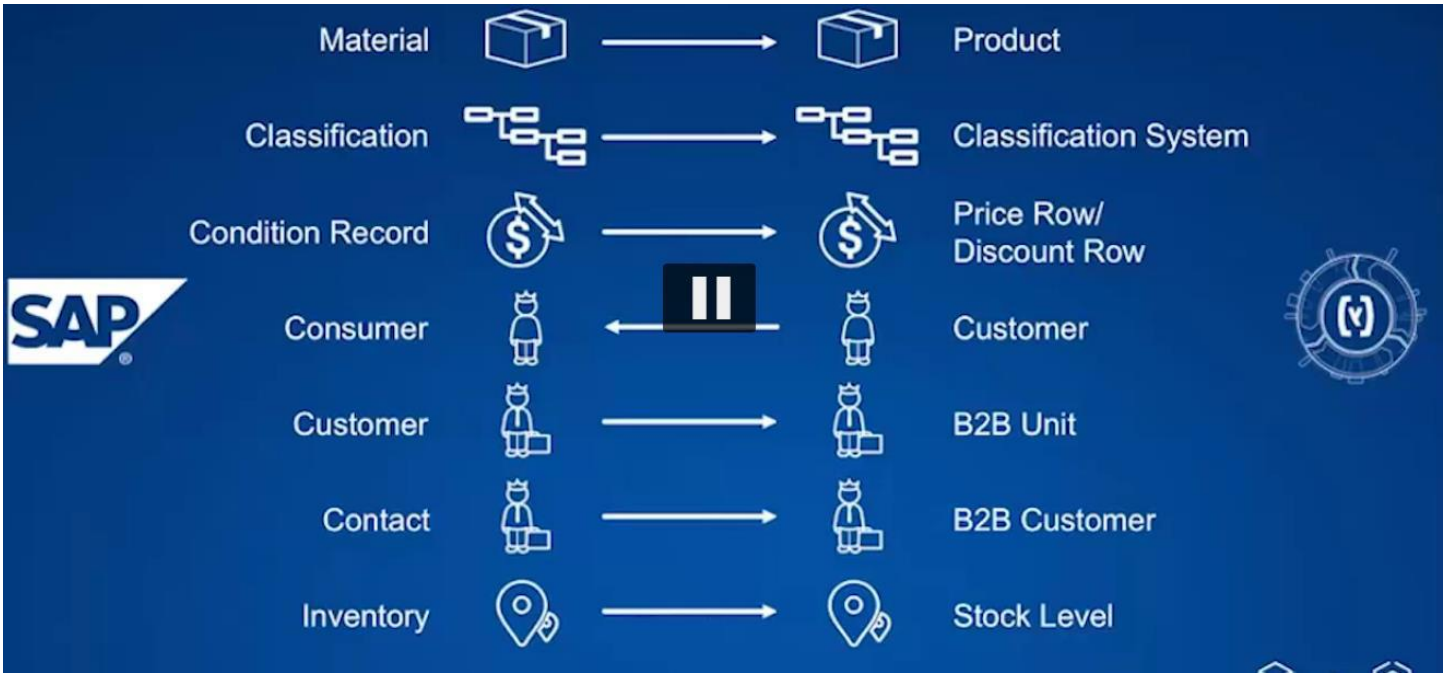
In Verizon System – Whom to Sell = **SoldTo**

In Verizon System – Whom to Invoice = **BillTo**

SRM = Company Buy (Purchase Order) &

CRM = Company Sell (Sales Order)

Q = What are the mapping entities between SAP & "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)

B2B Unit =

An Org Unit representing a Company Code, Department Node / Branch Node.

Branches of the B2B Root Unit **is B2B Org**.

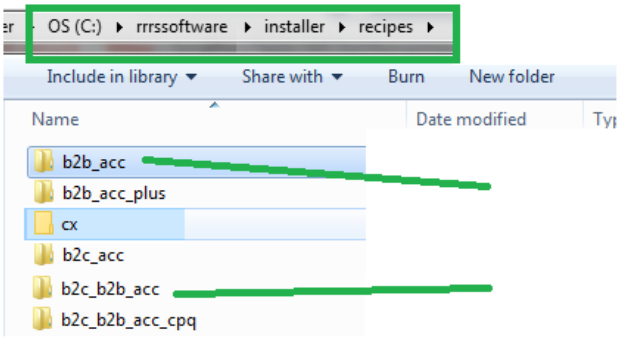
Orders are assigned to **B2B Units**.

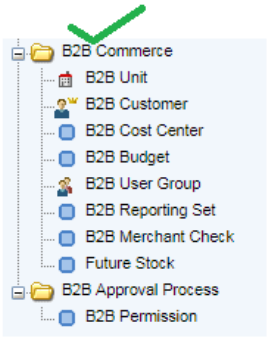
B2B Units can have an approvers group assigned.

B2B Customers are assigned to B2B Unit. They also members of B2B Group.

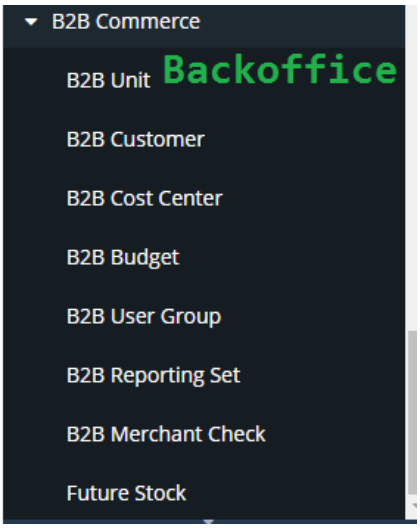
- AG = Sold To Party (Customers Who Place the Orders)
- RE = Bill To Party (Customers Who Receive the Invoice)
- RG = Payer (Customers Who Pay the Invoice)
- WE = Ship To Party (Customers Who Receive the Goods Recipient / Services)
- ZU = Author (Forwarding Agent)

B2B eCommerce is based on “B2B Unit … B2B Customer … ShipTo … SoldTo … Budget … Cost Center … B2B Approval Process … B2B Permission … “.

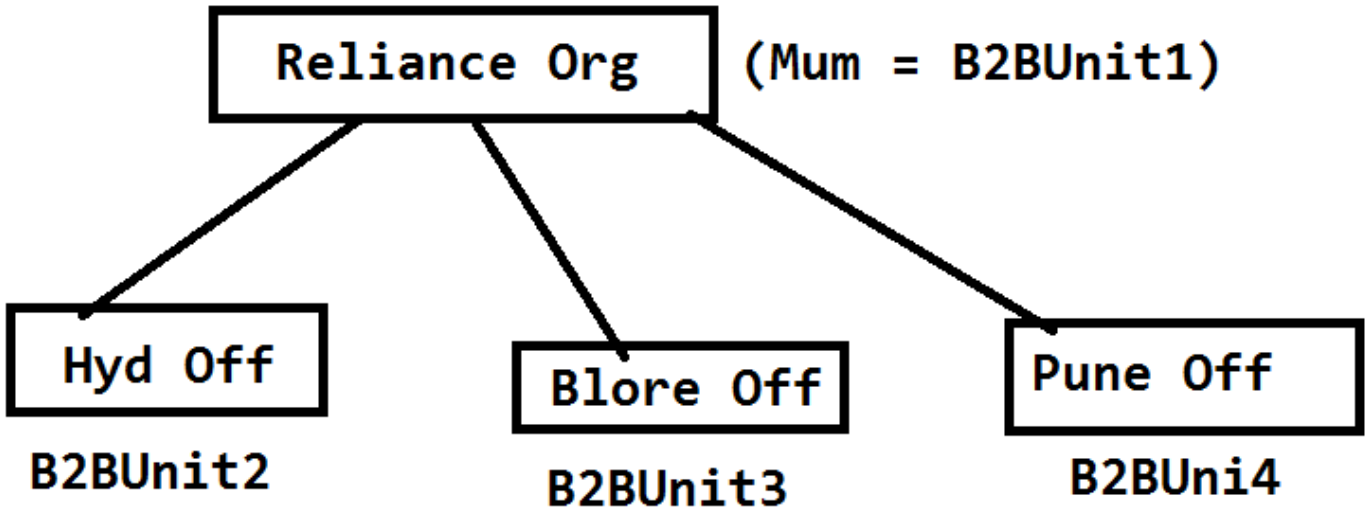


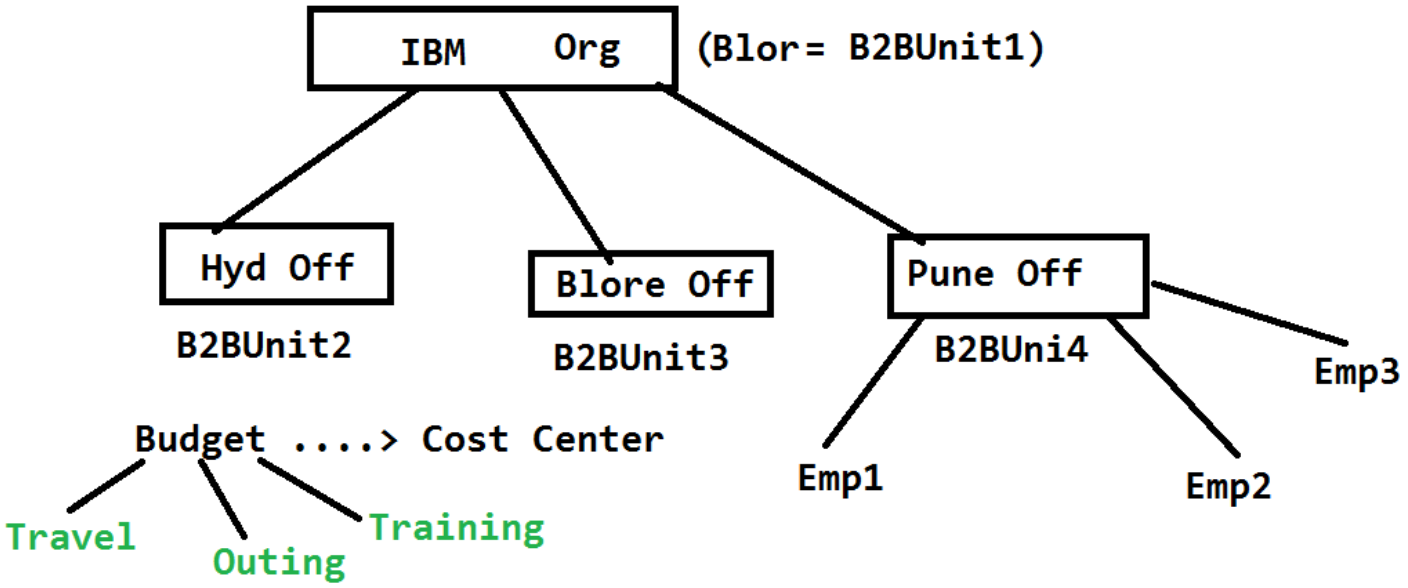


hMC

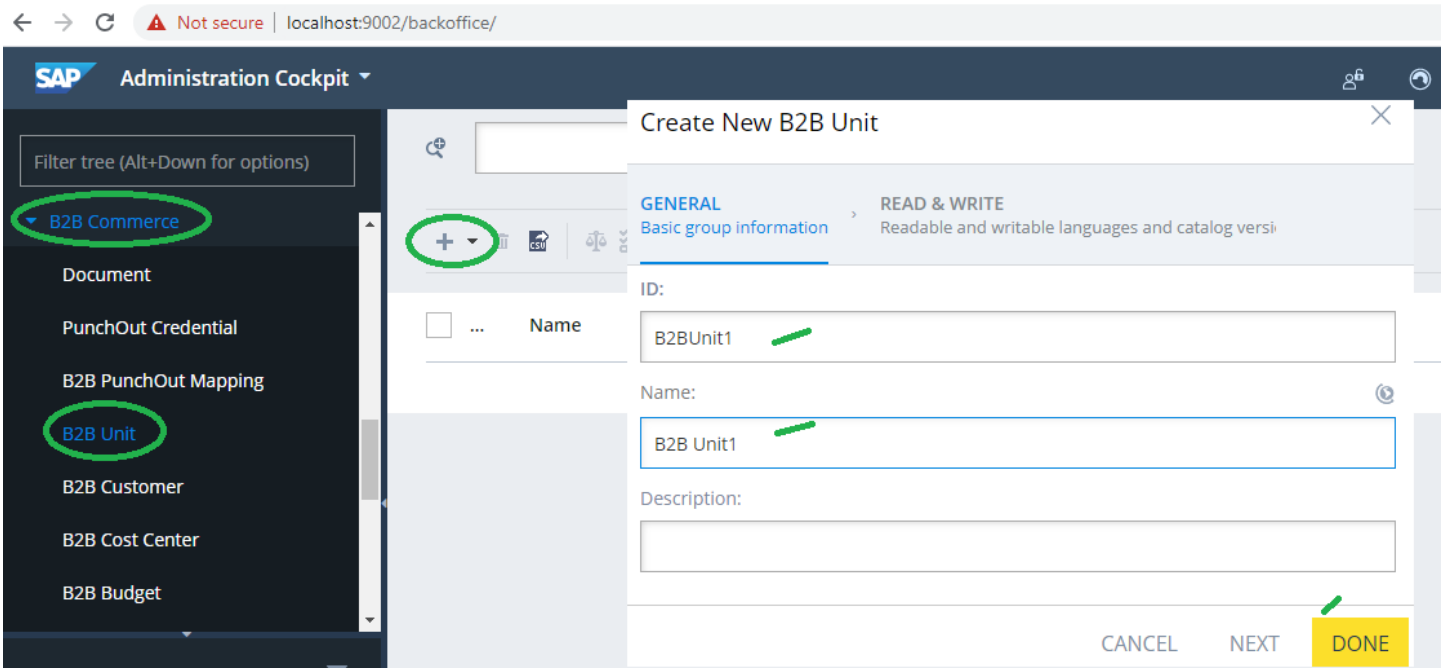


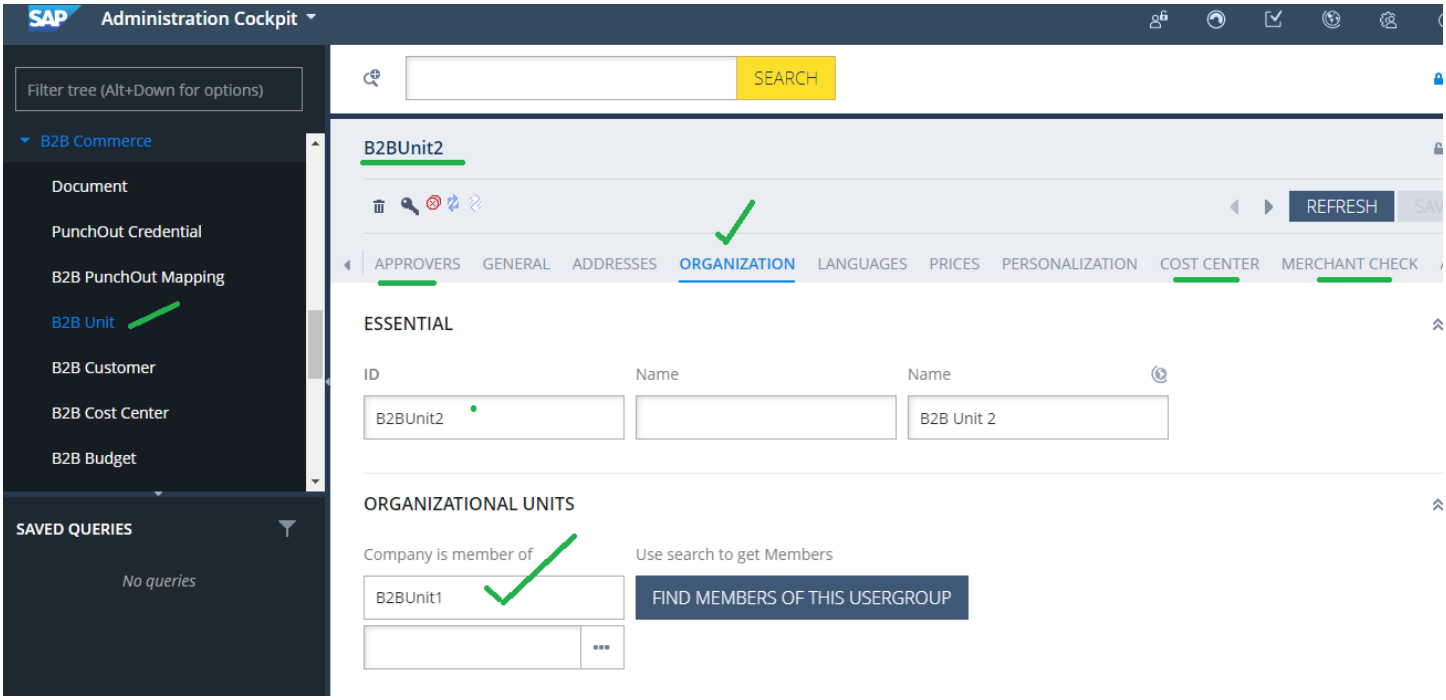
B2B Unit = Organization / Division / Department / Office / ……

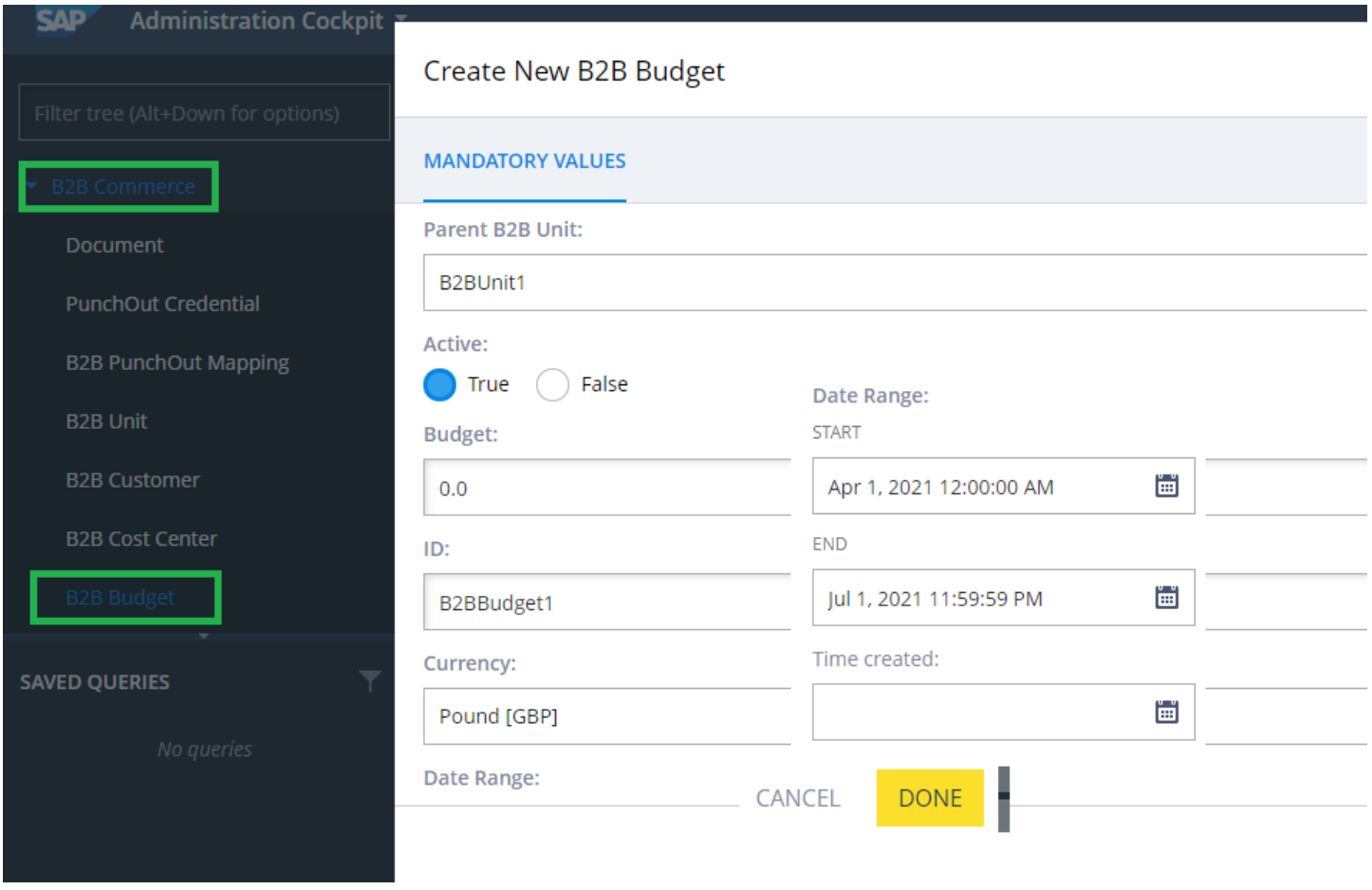


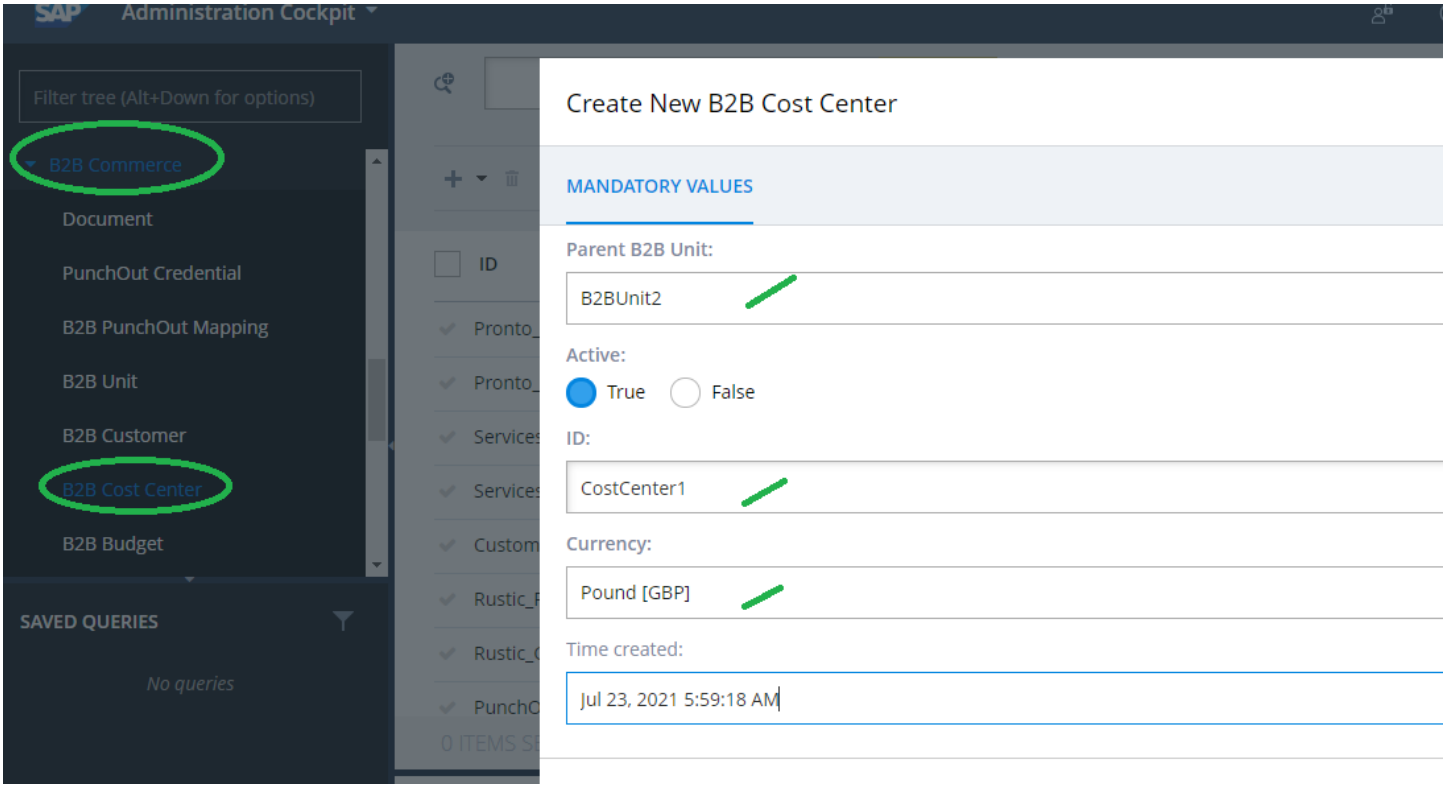


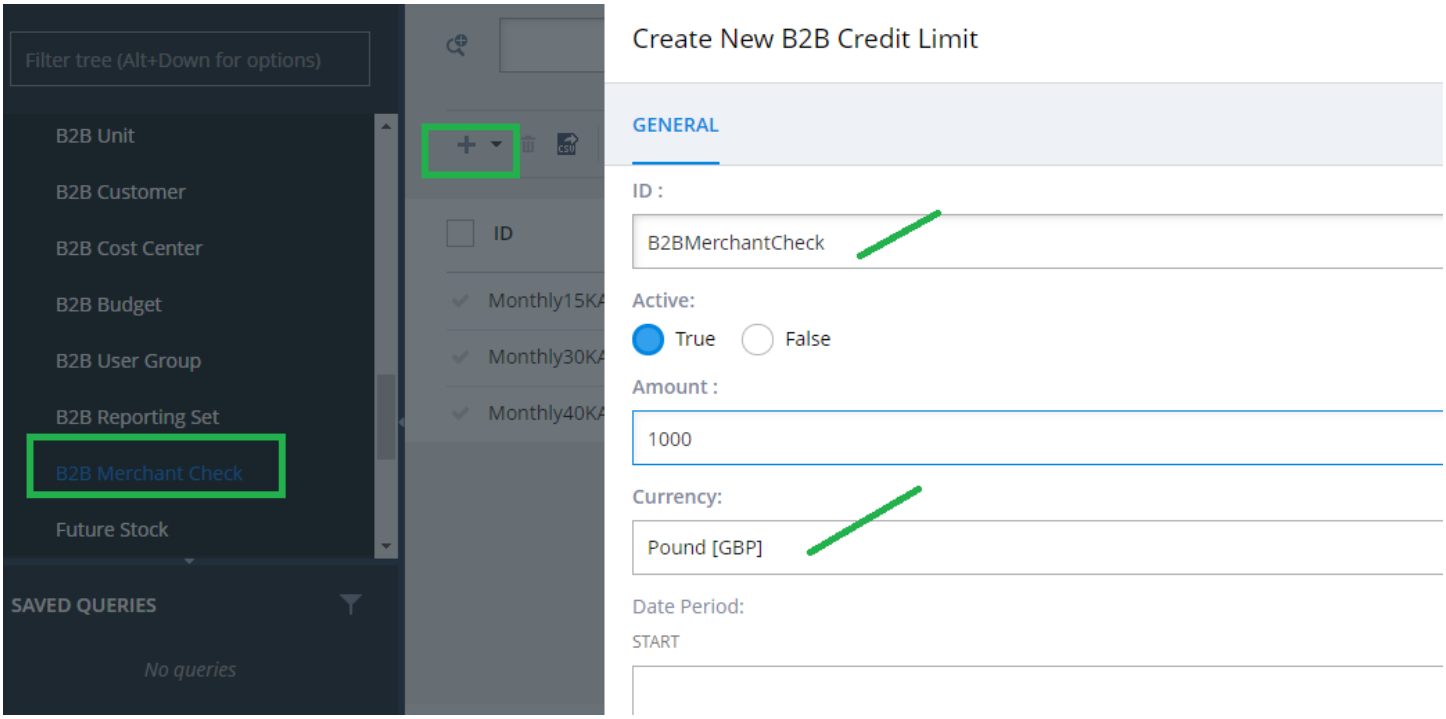
Note = If any B2BUnit is not having Parent B2BUnit then it's that B2BUnit is called "Root Organization". You can activate / deactivate B2BUnits.

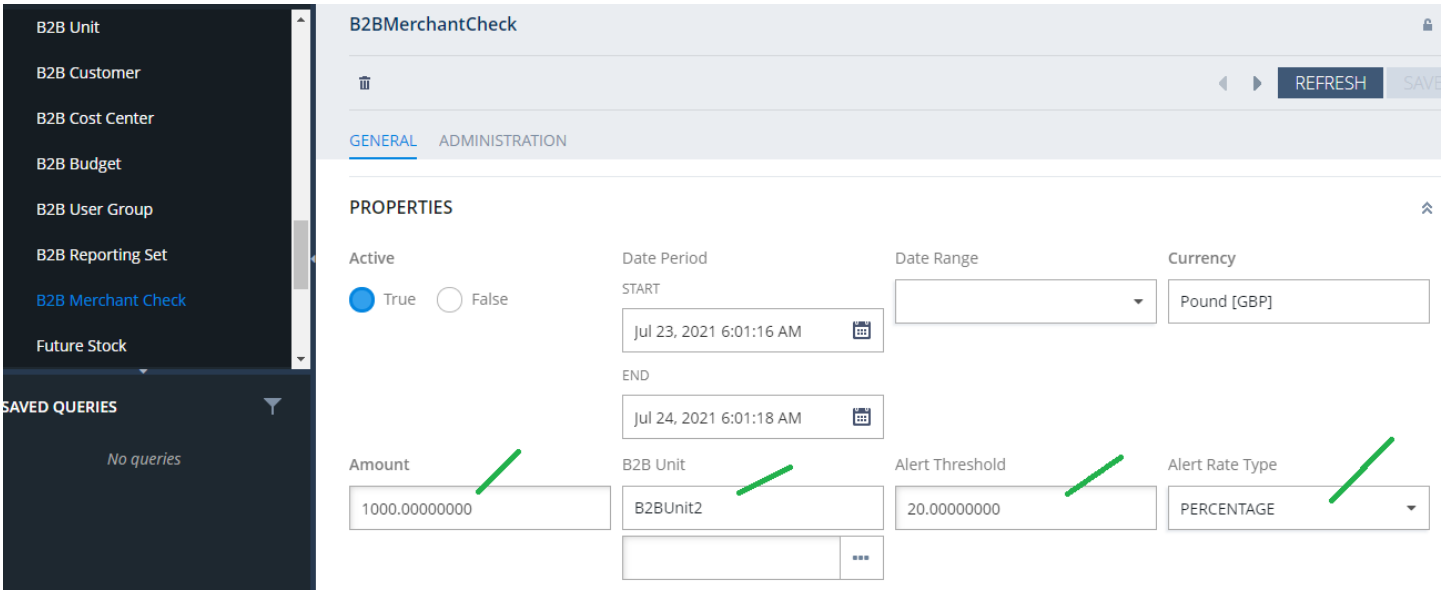


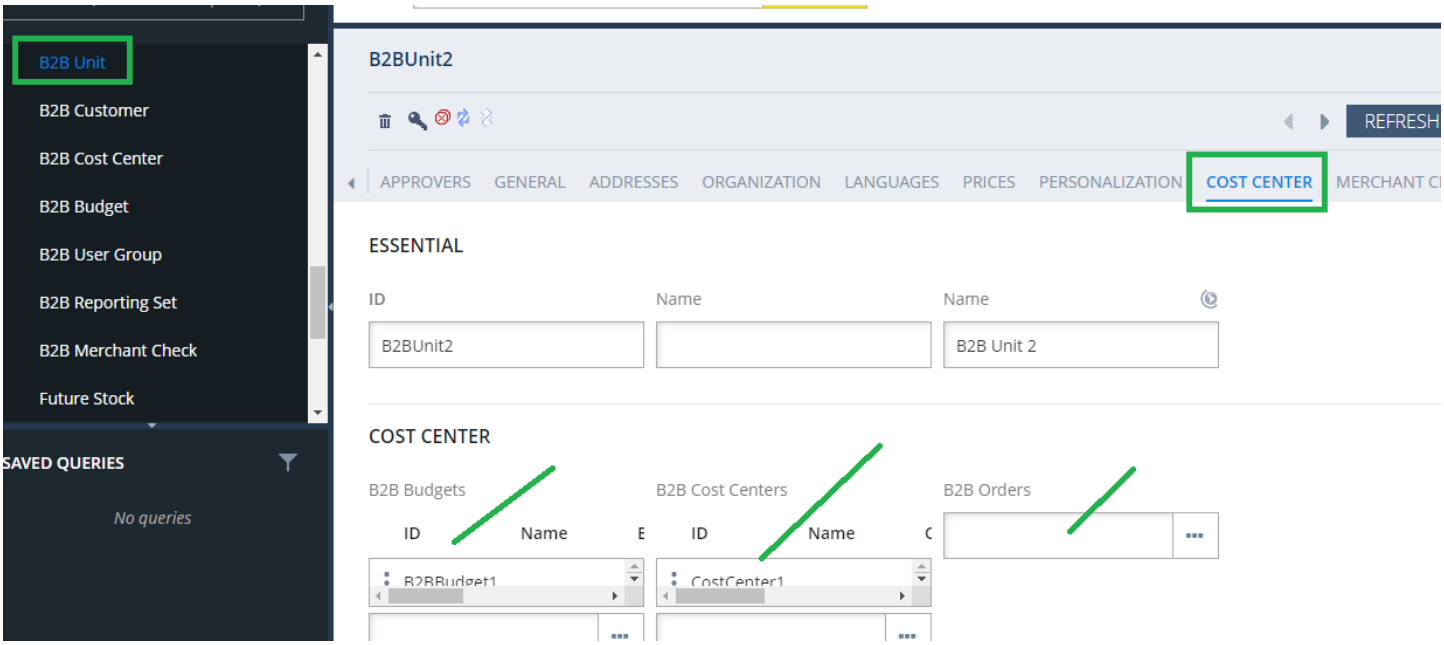

SAP
Administration Cockpit
Filter tree (Alt+Down for options)
B2B Commerce
Document
PunchOut Credential
B2B PunchOut Mapping
B2B Unit
B2B Customer
B2B Cost Center
B2B Budget
SAVED QUERIES
No queries
SEARCH
B2BUnit2
REFRESH
APPROVERS
GENERAL
ADDRESSES
ORGANIZATION
LANGUAGES
PRICES
PERSONALIZATION
COST CENTER
MERCHANT CHECK
ESSENTIAL
ID
Name
Name
B2BUnit2
B2B Unit 2
ORGANIZATIONAL UNITS
Company is member of
Use search to get Members
B2BUnit1
FIND MEMBERS OF THIS USERGROUP

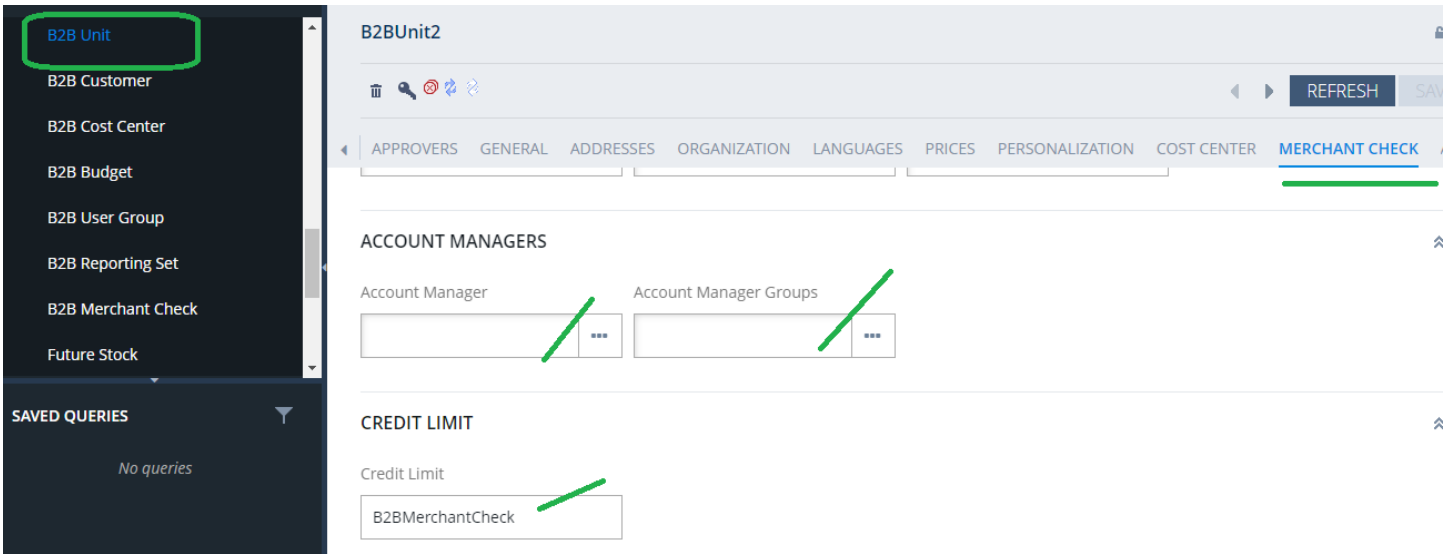

SAP
Administration Cockpit
Filter tree (Alt+Down for options)
B2B Commerce
Document
PunchOut Credential
B2B PunchOut Mapping
B2B Unit
B2B Customer
B2B Cost Center
B2B Budget
SAVED QUERIES
No queries
Create New B2B Budget
MANDATORY VALUES
Parent B2B Unit:
B2BUnit1
Active:
True
False
Budget:
0.0
ID:
B2BBudget1
Currency:
Pound [GBP]
Date Range:
START
Apr 1, 2021 12:00:00 AM
END
Jul 1, 2021 11:59:59 PM
Time created:
Date Range:
CANCEL
DONE


Administration Cockpit
Filter tree (Alt+Down for options)
B2B Commerce
Document
PunchOut Credential
B2B PunchOut Mapping
B2B Unit
B2B Customer
B2B Cost Center
B2B Budget
SAVED QUERIES
No queries
ID
Create New B2B Cost Center
MANDATORY VALUES
Parent B2B Unit:
B2BUnit2
Active:
True
False
ID:
CostCenter1
Currency:
Pound [GBP]
Time created:
Jul 23, 2021 5:59:18 AM


Filter tree (Alt+Down for options)
B2B Unit
B2B Customer
B2B Cost Center
B2B Budget
B2B User Group
B2B Reporting Set
B2B Merchant Check
Future Stock
SAVED QUERIES
No queries
ID
Create New B2B Credit Limit
GENERAL
ID :
B2BMerchantCheck
Active:
True
False
Amount :
1000
Currency:
Pound [GBP]
Date Period:
START







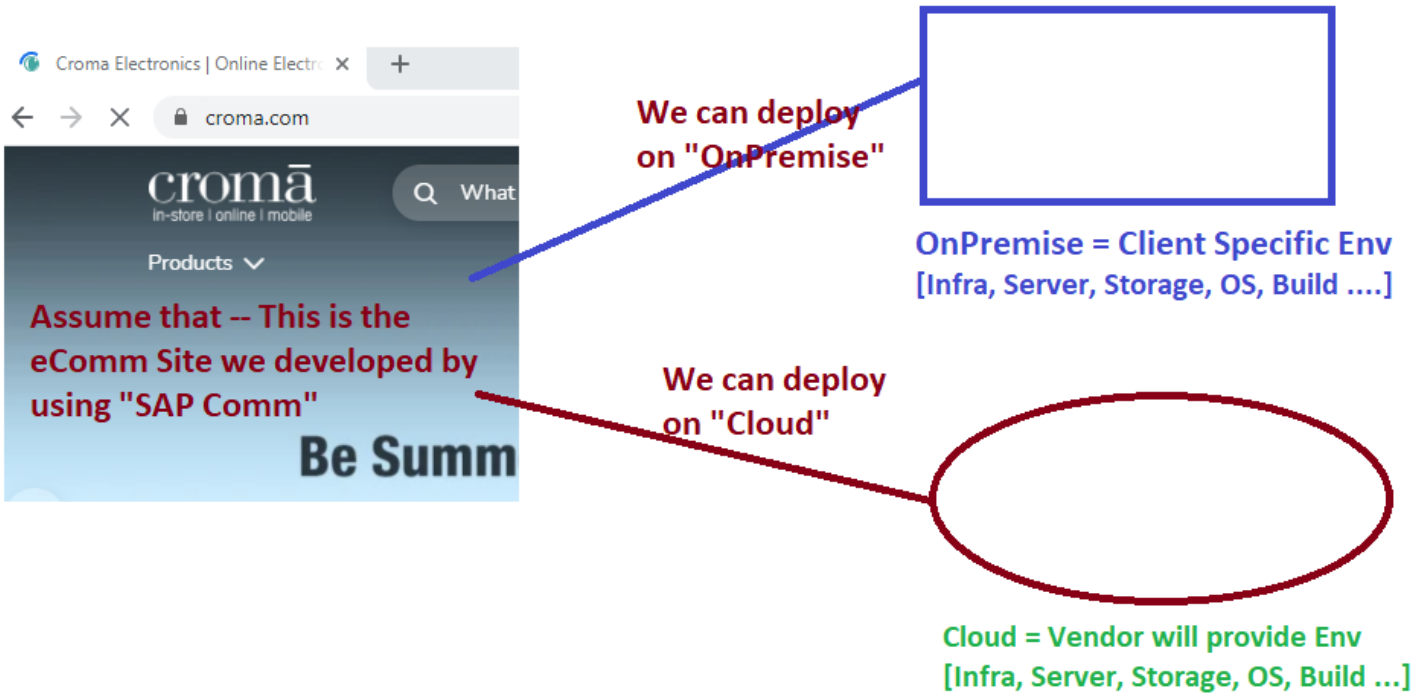
Contact Us = ChennaReddyTraining@RRRS.CO.IN

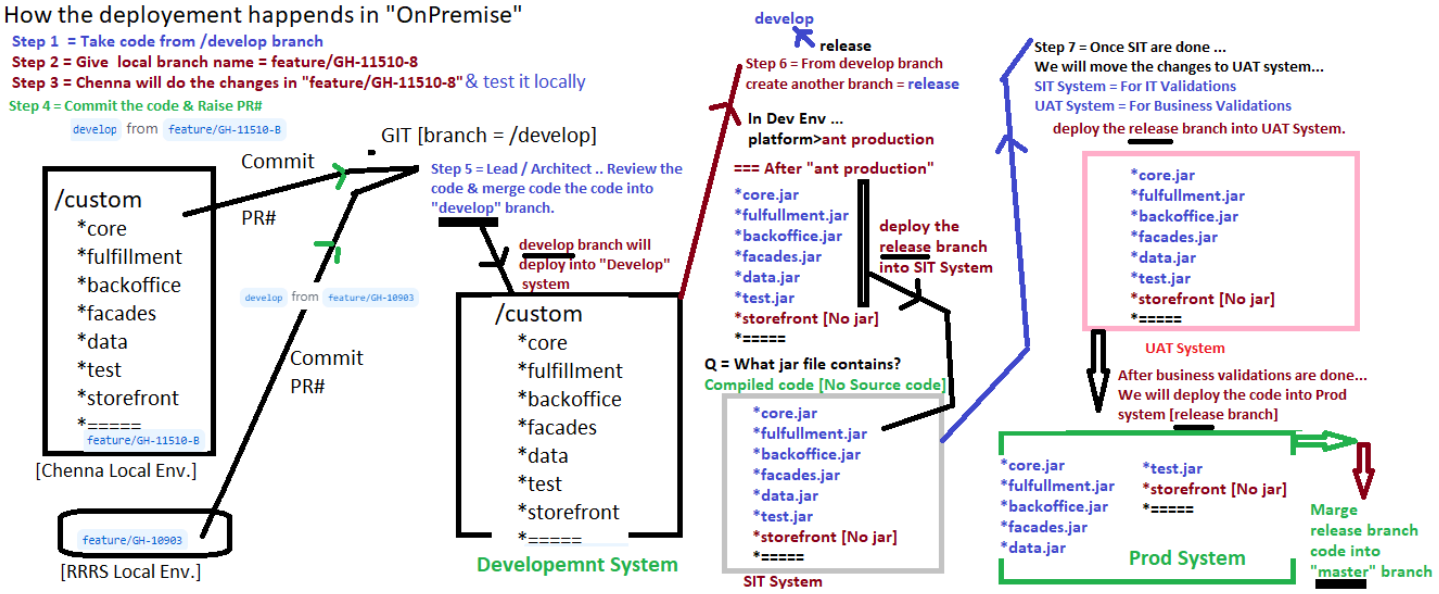
Scenario = Deployment

After developing the eComm / Shopping Site, we need deploy / host it : -

1) We can use client specific Environment / Infra = **OnPremise**

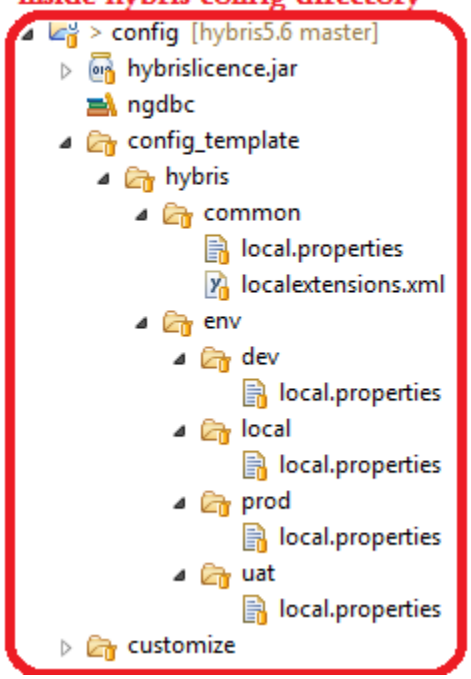
2) We can use Vendor specific Environment / Infra = **Cloud**





Q = Explain How to Manage the configuration properties for multiple environments [Dev … QA … UAT … Prod]?

I. Create below folder structure inside hybris config directory



Step 2 =

Place your properties in each one of local files as per environment related values. The property which have same values across environment will go to common local property file.

Step 3 =

Create a build.xml file inside config project and copy below content.

Step 4 =

Now we have a ANT build xml, which will create a environment specific property file. You can run like below pass the environment name as required.

c:\hybris\config>
ant -Denv=uat

Now you have a flexible solution to configure your hybris system more efficiently.

Q = Production Infrastructure for SAP Commerce (or) What is ideal Production server setup? (or) How many servers, cores / nodes do customers need?