**Wiki link:** [**https://wiki.hybris.com/pages/viewpage.action?spaceKey=accdoc&title=Email+WCMS+and+Process+Engine+Integration**](https://wiki.hybris.com/pages/viewpage.action?spaceKey=accdoc&title=Email+WCMS+and+Process+Engine+Integration)

###create an email page Template.

1. # Email page Template
   1. INSERT\_UPDATE EmailPageTemplate;$contentCV[unique=true];uid[unique=true];name;active;frontendTemplateName;subject(code);htmlTemplate(code);restrictedPageTypes(code)
   2. ;;CustomerEmailUpdateTemplate,Customer email update Template;true;customerEmailUpdate;vodafone\_Email\_Customer\_EmailUpdate\_Subject;vodafone\_Email\_Customer\_EmailUpdate\_Body;EmailPage
2. ###Update the emailPageTemplate by adding the velocitytemplate to customize.

# Templates for CMS Cockpit Page Edit

* 1. UPDATE EmailPageTemplate;$contentCV[unique=true];uid[unique=true];velocityTemplate[translator=de.hybris.platform.commerceservices.impex.impl.FileLoaderValueTranslator]
  2. ;;CustomerEmailUpdateTemplate;$jarResourceCms/structure-view/structure-customerEmailUpdateTemplate.vm

1. ###Adding the components to our pageTemplate
   1. INSERT\_UPDATE ContentSlotName;name[unique=true];template(uid,$contentCV)[unique=true][default='CustomerEmailUpdateTemplate'];validComponentTypes(code)
   2. ;SiteLogo;;;logo
   3. ;TopContent;;$wideContent;
   4. ;BottomContent;;$wideContent;
2. ###Adding the renderer templates by giving the Context file path name

# Email velocity templates

* 1. INSERT\_UPDATE RendererTemplate;code[unique=true];contextClass;rendererType(code)[default='velocity']
  2. ;vodafone\_Email\_Customer\_EmailUpdate\_Body;$emailPackageName.CustomerEmailUpdateContext
  3. ;vodafone\_Email\_Customer\_EmailUpdate\_Subject;$emailPackageName.CustomerEmailUpdateContext

1. **Create a class with the name CustomerEmailUpdateContext under the package which extends AbstractEmailContext<StoreFrontCustomerProcessModel>**
2. **override the init method to customize the renderer of the template.**

**For referral purpose follow** CustomerEmailUpdateContext

1. # Email velocity templates
   1. UPDATE RendererTemplate;code[unique=true];description[lang=$lang];templateScript[lang=$lang,translator=de.hybris.platform.yacceleratorcore.setup.FileLoaderValueTranslator]
   2. ;vodafone\_Email\_Customer\_EmailUpdate\_Body;Customer Email update Body;$emailResource/email-customerEmailUpdateBody.vm
   3. ;vodafone\_Email\_Customer\_EmailUpdate\_Subject;Customer Email update Subject;$emailResource/email-customerEmailUpdateSubject.vm
2. #define the email page
3. # Email Pages
   1. INSERT\_UPDATE EmailPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);defaultPage;approvalStatus(code)[default='approved']
   2. ;;customerEmailUpdate;Customer Registration Email;CustomerEmailUpdateTemplate;true;
   3. UPDATE EmailPage;$contentCV[unique=true];uid[unique=true];fromEmail[lang=$lang];fromName[lang=$lang]
   4. ;;CustomerEmailUpdate;"noreplywebshop.hu@vodafone.com";"Customer Services Team"
4. Bind reusable content slots to the EmailPageTemplate using ContentSlotForTemplate:
   1. INSERT\_UPDATE ContentSlotForTemplate;$contentCV[unique=true];uid[unique=true];position[unique=true];pageTemplate(uid,$contentCV)[unique=true][default='CustomerEmailUpdateTemplate'];contentSlot(uid,$contentCV)[unique=true];allowOverwrite
   2. ;;SiteLogo-customerEmailUpdate;SiteLogo;;EmailSiteLogoSlot;true
   3. ;;TopContent-customerEmailUpdate;TopContent;;EmailTopSlot;true
   4. ;;BottomContent-customerEmailUpdate;BottomContent;;EmailBottomSlot;true
5. **Create an item type of typegroup name="EmailProcess" in vodafonecore-items.xml**
   1. <itemtype code=" *CustomerEmailUpdateProcess*" extends="StoreFrontCustomerProcess"

autocreate="true" generate="true"

* + 1. jaloclass="hu.vodafone.core.jalo.process. *CustomerEmailUpdateProcessModel* ">
       - 1. </itemtype>

1. **create a bean in the vodafonecore-spring .xml**
   1. <bean id=*"generateCustomerEmailUpdate"* parent=*"abstractGenerateEmailAction"*>
      * 1. <property name=*"frontendTemplateName"* value=*"CustomerEmailUpdateTemplate"*/>
      1. </bean>
   2. *Here CustomerEmailUpdateTemplate is the name of the emailPageTemplate created earlier in the impex*
2. **create a process.xml as follows**
   1. <?xml version=*"1.0"* encoding=*"utf-8"*?>
   2. <process xmlns=*"http://www.hybris.de/xsd/processdefinition"* start=*"generateCustomerEmailUpdate"* name=*"customerEmailUpdateProcess"*
      1. processClass=*"hu.vodafone.core.model.process.CustomerEmailUpdateProcessModel"* onError=*"error"*>
      2. <action id=*"generateCustomerEmailUpdate"* bean=*"generateCustomerEmailUpdate"*>
         1. <transition name=*"OK"* to=*"sendEmail"*/>
         2. <transition name=*"NOK"* to=*"error"*/>
      3. </action>
      4. <action id=*"sendEmail"* bean=*"sendEmail"*>
         1. <transition name=*"OK"* to=*"removeSentEmail"*/>
         2. <transition name=*"NOK"* to=*"failed"*/>
      5. </action>
      6. <action id=*"removeSentEmail"* bean=*"removeSentEmail"*>
         1. <transition name=*"OK"* to=*"success"*/>
         2. <transition name=*"NOK"* to=*"error"*/>
      7. </action>
      8. <end id=*"error"* state=*"ERROR"*>Something went wrong.</end>
      9. <end id=*"failed"* state=*"FAILED"*>Could not send customer email update confirmation email..</end>
      10. <end id=*"success"* state=*"SUCCEEDED"*>Sent customer email update confirmation email.</end>
   3. </process>
3. **create an event by extending the AbstractCommerceUserEvent<BaseSiteModel>**

**For referral purpose follow** CustomerEmailUpdateEvent

1. **create an listener once the above event triggers**

**For referral purpose follow** CustomerEmailUpdateEventListener

1. **override the method of onSiteEvent of event listener class as below**

@Override

**protected** **void** onSiteEvent(CustomerEmailUpdateEvent emailUpdate) {

**final** CustomerEmailUpdateProcessModel customerEmailUpdateProcessModel = (CustomerEmailUpdateProcessModel) getBusinessProcessService()

* + - * 1. .createProcess("customerEmailUpdateProcess-" + emailUpdate.getCustomer().getUid() + "-" + System.*currentTimeMillis*(),

"customerEmailUpdateProcess");

customerEmailUpdateProcessModel.setCustomer(emailUpdate.getCustomer());

customerEmailUpdateProcessModel.setSite(emailUpdate.getSite());

customerEmailUpdateProcessModel.setStore(emailUpdate.getBaseStore());

customerEmailUpdateProcessModel.setLanguage(emailUpdate.getLanguage());

customerEmailUpdateProcessModel.setCurrency(emailUpdate.getCurrency());

getModelService().save(customerEmailUpdateProcessModel);

getBusinessProcessService().startProcess(customerEmailUpdateProcessModel);

* 1. }