**Backoffice Cockpit NG**

1. **Create custom item types for backoffice**

<items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="items.xsd">

<collectiontypes>

<collectiontype code="TeamList" elementtype="Team"

autocreate="true" generate="false" type="set" />

</collectiontypes>

<enumtypes>

<enumtype generate="true" code="PlayerType" autocreate="true">

<value code="BATSMEN" />

<value code="BOWLER" />

<value code="ALLROUNDER" />

</enumtype>

<enumtype generate="true" code="MatchType" autocreate="true">

<value code="TEST" />

<value code="ODI" />

<value code="T20" />

</enumtype>

<enumtype generate="true" code="PlayerStatus" autocreate="true">

<value code="PLAYING" />

<value code="RETIRED" />

</enumtype>

</enumtypes>

<relations>

<relation code="Stadium2Player" localized="false" generate="true"

autocreate="true">

<sourceElement type="Stadium" qualifier="homeGround"

cardinality="one">

<description>Home Ground</description>

<modifiers read="true" write="true" search="true" />

</sourceElement>

<targetElement type="Player" qualifier="players"

cardinality="many" collectiontype="set">

<description>Players</description>

<modifiers read="true" write="true" search="true"

optional="true" />

</targetElement>

</relation>

</relations>

<itemtypes>

<!-- Add your item definitions here -->

<typegroup name="Cricket">

<itemtype code="Team" extends="GenericItem" autocreate="true"

generate="true">

<description>Cricket Team</description>

<deployment table="team" typecode="11001" />

<attributes>

<attribute qualifier="name" type="java.lang.String">

<description>Name of team</description>

<modifiers read="true" write="true" search="true"

initial="true" unique="true" optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="description" type="java.lang.String">

<description>Description of the team</description>

<modifiers read="true" write="true" search="true"

optional="true" />

<persistence type="property" />

</attribute>

</attributes>

</itemtype>

<itemtype code="Stadium" extends="GenericItem" autocreate="true"

generate="true">

<description>Cricket Stadium</description>

<deployment table="stadium" typecode="11002" />

<attributes>

<attribute qualifier="name" type="java.lang.String">

<description>Name of Stadium</description>

<modifiers read="true" write="true" search="true"

initial="true" unique="true" optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="stadiumType" type="MatchType">

<description>Stadium type</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="capacity" type="java.lang.Integer">

<description>Stadium capacity</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="city" type="java.lang.String">

<description>city</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

</attributes>

</itemtype>

<itemtype code="Player" extends="GenericItem" autocreate="true"

generate="true">

<description>Cricket Stadium</description>

<deployment table="player" typecode="11003" />

<attributes>

<attribute qualifier="capNumber" type="java.lang.Integer">

<description>Player number</description>

<modifiers read="true" write="true" search="true"

initial="true" unique="true" optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="name" type="java.lang.String">

<description>Name of player</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="playerType" type="PlayerType">

<description>Player type</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="preferMatchType" type="MatchType">

<description>Match type</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="debutMatch" type="java.util.Date">

<description>Debut match</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="playerStatus" type="PlayerStatus">

<description>status</description>

<modifiers read="true" write="true" search="true"

optional="false" />

<persistence type="property" />

</attribute>

<attribute qualifier="bestScored" type="java.lang.String">

<description>Best betting score of player</description>

<modifiers read="true" write="true" search="true"

optional="true" />

<persistence type="property" />

</attribute>

<attribute qualifier="bestBowling" type="java.lang.String">

<description>Best bowling score of player</description>

<modifiers read="true" write="true" search="true"

optional="true" />

<persistence type="property" />

</attribute>

<attribute qualifier="measureTeam" type="TeamList">

<description>measure team</description>

<modifiers read="true" write="true" search="true"

optional="true" />

<persistence type="property" />

</attribute>

<attribute qualifier="isCaptain" type="java.lang.Boolean">

<description>Is Captain</description>

<defaultvalue>Boolean.FALSE</defaultvalue>

<persistence type="property" />

<modifiers read="true" write="true" search="true"

optional="false" />

</attribute>

</attributes>

</itemtype>

</typegroup>

</itemtypes>

</items>

1. **Customizing Backoffice**

**Content tag attributes**

In the Backoffice Application, you group cockpit configuration elements inside context tags. Context tag could have the following attributes:

* component: It specifies which component this configuration should apply in. When the widget loads the configuration, it specifies the component's name. To check a default component value for a given widget and how it is overridden go to the widget's documents.
* principal: It specifies a user or a group of users that this configuration applies for. For example, the Explorer Tree widget can have different nodes for different users.
* type: If the widget can handle objects of many types, you can specify the configuration for each type. For example, for the Editor Area widget you can specify distinct tab configurations for two different types.
* parent: It specifies a parent configuration which is supposed to be merged.
* merge-by: Together with the parent attribute it declares by which properties the configuration should be merged, see the two examples below:

1. **explorer-tree**

<context component=*"explorer-tree"* merge-by=*"module"*>

<n:explorer-tree xmlns:n=*"http://www.hybris.com/cockpitng/config/explorertree"*>

<n:navigation-node id=*"Training"*>

<n:type-node id=*"Team"* code=*"Team"* />

<n:type-node id=*"Stadium"* code=*"Stadium"* />

<n:type-node id=*"Player"* code=*"Player"* />

</n:navigation-node>

</n:explorer-tree>

</context>

1. **create-wizard**

* Single step create wizard:

<context component=*"create-wizard"* type=*"Team"*>

<wz:flow xmlns:wz=*"http://www.hybris.com/cockpitng/config/wizard-config"*

id=*"TeamWizard"* title=*"create.team.title"*>

<wz:prepare id=*"teamPrepare"*>

<wz:initialize property=*"newTeam"* type=*"Team"* />

</wz:prepare>

<wz:step id=*"step1"* label=*"create.team.general.label"*

sublabel=*"create.team.general.sublabel"*>

<wz:content id=*"step1.content"*>

<wz:property-list root=*"newTeam"*>

<wz:property qualifier=*"name"* />

<wz:property qualifier=*"description"* />

</wz:property-list>

</wz:content>

<wz:navigation id=*"step1.navigation"*>

<wz:cancel />

<wz:done>

<wz:save property=*"newTeam"* />

</wz:done>

</wz:navigation>

</wz:step>

</wz:flow>

</context>

* Multi step create wizard:

<context component=*"create-wizard"* type=*"Player"*>

<wz:flow xmlns:wz=*"http://www.hybris.com/cockpitng/config/wizard-config"*

id=*"PlayerWizard"* title=*"create.player.title"*>

<wz:prepare id=*"playerPrepare"*>

<wz:initialize property=*"newPlayer"* type=*"Player"* />

</wz:prepare>

<wz:step id=*"step1"* label=*"create.player.general.label"*

sublabel=*"create.player.general.sublabel"*>

<wz:content id=*"step1.content"*>

<wz:property-list root=*"newPlayer"*>

<wz:property qualifier=*"name"* />

<wz:property qualifier=*"capNumber"* />

<wz:property qualifier=*"preferMatchType"* />

<wz:property qualifier=*"homeGround" /*>

<wz:property qualifier=*"debutMatch"* />

<wz:property qualifier=*"playerStatus"* />

</wz:property-list>

</wz:content>

<wz:navigation id=*"step1.navigation"*>

<wz:cancel />

<wz:next />

</wz:navigation>

</wz:step>

<wz:step id=*"step2"* label=*"create.stadium.general.label"*

sublabel=*"create.team.general.sublabel"*>

<wz:content id=*"step1.content"*>

<wz:property-list root=*"newPlayer"*>

<wz:property qualifier=*"playerType"* />

<wz:property qualifier=*"bestScored"* />

<wz:property qualifier=*"bestBowling"* />

<wz:property qualifier=*"measureTeam"* />

</wz:property-list>

</wz:content>

<wz:navigation id=*"step1.navigation"*>

<wz:cancel />

<wz:back />

<wz:done>

<wz:save property=*"newPlayer"* />

</wz:done>

</wz:navigation>

</wz:step>

</wz:flow>

</context>

* Next/Done button validation

<wz:done visible=*"!#empty(newStadium.name) AND newStadium.stadiumType !=null"*>

<wz:save property=*"newStadium"* />

</wz:done>

* Reference search condition filter
  1. Reference parameter:

<wz:property qualifier=*"homeGround"*>

<wz:editor-parameter>

<wz:name>referenceSearchCondition\_stadiumType</wz:name>

<wz:value>{parentObject.preferMatchType}</wz:value>

</wz:editor-parameter>

<wz:editor-parameter>

<wz:name>availableValuesProvider</wz:name>

<wz:value>preferMatchTypeReferenceSearchFacade</wz:value>

</wz:editor-parameter>

</wz:property>

* 1. Declare bean:

<alias alias=*"preferMatchTypeReferenceSearchFacade "* name=*" defaultPreferMatchTypeReferenceSearchFacade "* />

<bean id=*"defaultPreferMatchTypeReferenceSearchFacade"*

class=*"de.hybris.taining.search.PreferMatchTypeReferenceSearchFacade"*>

<property name=*"fieldSearchFacade"* ref=*"fieldSearchFacade"* />

<property name=*"typeService"* ref=*"typeService"* />

</bean>

* 1. Define ReferenceEditorSearchFacade.
* Available values provider filter

<wz:property qualifier=*"measureTeam"* >

<wz:editor-parameter> <wz:name>referenceSearchCondition\_name\_startsWith</wz:name>

<wz:value>India</wz:value>

</wz:editor-parameter>

</wz:property>

* Show or hide optional field for Enum

<wz:property qualifier=*"* *playerStatus"* type=*"java.lang.Enum(PlayerStatus)"*

editor=*"com.hybris.cockpitng.editor.nonoptionalenum"* />

* Show or hide optional field for Boolean

<wz:property qualifier=*"isCaptain"*>

<wz:editor-parameter>

<wz:name>showOptionalField</wz:name>

<wz:value>false</wz:value>

</wz:editor-parameter>

</wz:property>

1. **Base**

<context type=*"Team"* component=*"base"*>

<y:base xmlns:y=*"http://www.hybris.com/cockpit/config/hybris"*>

<y:labels>

<y:label>'Team - ' + name + ' ' +

(description != null ? description :' ' )

</y:label>

</y:labels>

</y:base>

</context>

1. **simple-search**
   1. With default sort.

<context type=*"Tender"* component=*"simple-search"*>

<yss:simple-search

xmlns:yss=*"http://www.hybris.com/cockpitng/config/simplesearch"*>

<yss:field name=*"referenceNo"* />

</yss:simple-search>

</context>

* 1. With custom sort.

<yss:sort-field name=*"referenceNo"* asc=*"true"* />

1. **advanced-search**
   1. AND/OR operator

<context type=*"Team"* component=*"advanced-search"*>

<as:advanced-search

xmlns:as=*"http://www.hybris.com/cockpitng/config/advancedsearch"*

connection-operator=*"AND"*>

<as:field-list>

<as:field name=*"name"* operator=*"contains"* selected=*"true"* />

<as:field name=*"description"* operator=*"contains"* selected=*"true"* />

</as:field-list>

<as:sort-field name=*"name"* asc=*"true"* />

</as:advanced-search>

</context>

1. **list-view**

<context type=*"Team"* component=*"listview"*>

<list:list-view

xmlns:list=*"http://www.hybris.com/cockpitng/component/listView"*>

<list:column qualifier=*"name"* />

<list:column qualifier=*"description"* />

</list:list-view>

</context>

1. **editor-area**
   1. Single Tab

<context merge-by=*"module"* component=*"editor-area"* type=*"Team"*>

<editorArea:editorArea

xmlns:editorArea=*"http://www.hybris.com/cockpitng/component/editorArea"*>

<editorArea:tab name=*"team.editor.tab.name"*>

<editorArea:section name=*"team.editor.section.name"*>

<editorArea:attribute qualifier=*"name"*

readonly=*"true"* />

<editorArea:attribute qualifier=*"description"* />

</editorArea:section>

</editorArea:tab>

</editorArea:editorArea>

</context>

* 1. Multi Tabs
  2. With editor-parameter for date format

<editorArea:attribute xmlns=*"http://www.hybris.com/cockpitng/component/editorArea"*

qualifier=*"closeTime"*>

<editorArea:editor-parameter>

<name>dateFormat</name>

<value>MMM dd, yyyy h:mm a</value>

</editorArea:editor-parameter> </editorArea:attribute>

* 1. With editor-parameter

<editorArea:editor-parameter>

<editorArea:name>referenceSearchCondition\_businessUnit </editorArea:name>

<editorArea:value>{parentObject.businessUnit}</editorArea:value>

</editorArea:editor-parameter>

<editorArea:editor-parameter> <editorArea:name>referenceSearchCondition\_uid\_endsWith</editorArea:name>

<editorArea:value>MarketingListGroup</editorArea:value>

</editorArea:editor-parameter>

1. **Editor area dynamic forms**

<context type=*"Tender"* component=*"editorAreaDynamicForms"* merge-by=*"module"*>

<df:dynamicForms xmlns:df=*"http://www.hybris.com/cockpitng/component/dynamicForms"*>

<df:attribute id=*"disableBestScored"*

triggeredOn=*"playerType"* qualifier=*"bestScored"*

computedValue=*"playerType==T(de.hybris.training.enums.PlayerType).BOWLER ? null : bestScored"* disabledIf=*"playerType==null OR playerType==T(de.hybris.training.enums.PlayerType).BOWLER"* />

<df:attribute id=*"disableBestBowling"* triggeredOn=*"playerType"* qualifier=*"bestBowling"* computedValue=*"playerType==T(de.hybris.training.enums.PlayerType).BATSMEN ? null : bestBowling"* disabledIf=*"playerType==null OR playerType==T(de.hybris.training.enums.PlayerType).BATSMEN"* />

</df:dynamicForms>

</context>

1. **Configurable flow dynamic forms**

<context type=*"Player"* component=*"configurableFlowDynamicForms"*merge-by=*"module"*>

<df:dynamicForms modelProperty=*"newPlayer"*>

<df:attribute id=*"disableBestScored"*

triggeredOn=*"playerType"* qualifier=*"bestScored"*

computedValue=*"playerType==T(de.hybris.training.enums.PlayerType).BOWLER ? null : bestScored"* disabledIf=*"playerType==null OR playerType==T(de.hybris.training.enums.PlayerType).BOWLER"* />

<df:attribute id=*"disableBestBowling"* triggeredOn=*"playerType"* qualifier=*"bestBowling"* computedValue=*"playerType==T(de.hybris.training.enums.PlayerType).BATSMEN ? null : bestBowling"* disabledIf=*"playerType==null OR playerType==T(de.hybris.training.enums.PlayerType).BATSMEN"* />

</df:dynamicForms>

</context>

1. **notification-area**

<context component=*"notification-area"*

source=*"de.hybris.training.backoffice.actions.playerstatusaction"*

merge-by=*"source"*>

<na:notification-area>

<na:notifications eventType=*"PlayerRetired"* level=*"SUCCESS"*>

<na:message>player.retired.status.action.success.message</na:message>

</na:notifications>

<na:notifications eventType=*"PlayerRetired"* level=*"FAILURE"*>

<na:message>player.retired.status.action.error.message</na:message>

</na:notifications>

</na:notification-area>

</context>

NotificationUtils.*notifyUser*("de.hybris.training.backoffice.actions.playerstatusaction ", "PlayerRetired", Level.***SUCCESS***, **new** Object[] { Collections.*singletonList*(playerModel) });

1. **Back Office Look & Feel (labels)**

com.hybris.cockpitng.widgets.configurableflow.create.team.title=Create Team

com.hybris.cockpitng.widgets.configurableflow.create.team.general.label=Provide team details...

com.hybris.cockpitng.widgets.configurableflow.create.team.general.sublabel=Provide team details...

team.editor.tab.name=Team

team.editor.section.name=Team Details

type.playertype.name=Player Type

type.playertype.description=Player Type [BATSMEN, BOWLER, ALLROUNDER]

type.playertype.BATSMEN.name=Batsmen

type.playertype.description=Batsmen

type.team.name=Team

type.team.description=Team

type.team.name.name=Team name

type.team.name.description=Team name

type.team.description.name=Team description

type.team.description.description=Team description