**Delo s Generics – xml**

Definiranje entitet

Namenjeno definiranju vseh entitet, ki se bodo uporabljale v eni xml datoteki. Tem entitetam se kasneje definira bazne objekte in atribute.

<DefinedEntities>

<Entity type=*"RPGV\_VINO\_OCENE\_LOV\_VW"* />

<Entity type=*"RPGV\_VINO\_OCENE"* />

<Entity type=*"VINO\_PODATKI\_WEB\_VW"* />

<Entity type=*"ORG\_PODATKI\_WEB\_VW"* />

<Entity type=*"STR\_STR\_CODELIST"* />

</DefinedEntities>

Definiranje DB entitet

Namenjeno definiranju baznih objektov entiteti. Bazni objekt je lahko view ali tabela. Če se uporabi definiran objekt tudi za vnos novih podatkov je potrebno paziti, da je view updateable.

Entity vsebuje:

1. type – tip entitete (imena, ki so definirana v node-u DefinedEntities)
2. dbTable– ime podatkovne tabele ali view-ja

<DBEntities>

<Entity type=*"RPGV\_VINO\_OCENE\_LOV\_VW"* dbTable=*"RPGV\_VINO\_OCENE\_LOV\_VW"*/>

<Entity type=*"RPGV\_VINO\_OCENE"* dbTable=*"RPGV\_VINO\_OCENE"*/>

<Entity type=*"VINO\_PODATKI\_WEB\_VW"* dbTable=*"VINO\_PODATKI\_WEB\_VW"*/>

<Entity type=*"ORG\_PODATKI\_WEB\_VW"* dbTable=*"ORG\_PODATKI\_WEB\_VW"*/>

</DBEntities>

Definiranje XML entitet

Namenjeno za definiranje iskalnih atributov ali combo box-ov.

Entity vsebuje:

1. type – tip entitete (imena, ki so definirana v node-u DefinedEntities)
2. name– ime sklopa, ki vsebuje iskalne atribute ali atribute combo box-a
   1. EntityAtrribute – definira atribut
      1. name – ime atributa, ki je v primeru, da se nanaša na bazno tabelo ali view enak imenu v bazi.
      2. Label – ime, ki naj se prikaže na ekranu. V primeru, da label ni definiran se prikaže kar ime atributa.

Iskalni atributi

<Entity type=*"RPGV\_VINO\_OCENE\_LOV\_VW"* name=*"RPGV\_VINO\_OCENE\_LOV\_VW"*>

<EntityAttribute name=*"OCENA\_ID"* label=*"Ocena ID"*/>

<EntityAttribute name=*"VINO\_ID"* label=*"Vino ID"*/>

...

</Entity>

Combo box

<Entity type=*"STR\_STR\_CODELIST"* name=*"STR\_STR\_CODELIST"*>

<EntityAttribute name=*"CODE"* type=*"0"*/>

<EntityAttribute name=*"VALUE"* type=*"0"*/>

</Entity>

Definiranje data sources

Določi se povezavo med posameznim widget-om in izvorom podatkov. Izvor podatkov je lahko oracle baza , xml datoteka, ...

Entity vsebuje:

1. type – tip entitete (imena, ki so definirana v node-u DefinedEntities)
2. name – ime entitete
3. datasourceType – tip podatkovnega vira (oracle, XML):
   1. Oracle
      1. dbTable –se uporabi v primeru, da je datasourceType=«oracle« in vsebuje ime tabele ali view-a
      2. primarykey – se uporabi v primeru, da je datasourceType=«oracle« in vsebuje ime atributa, ki je ključ na tabeli ali view-ju.
   2. XML
      1. XMLFile - se uporabi v primeru, da je datasourceType=«XML« in vsebuje ime xml datoteke s podaki

<EntityDatasources>

<Entity type=*"RPGV\_VINO\_OCENE\_LOV\_VW"* name=*"vino\_ocene\_lov\_vw"* datasourceType=*"oracle"* dbTable=*"RPGV\_VINO\_OCENE\_LOV\_VW"* primarykey=*"OCENA\_ID"*/>

<Entity type=*"RPGV\_VINO\_OCENE"* name=*"vino\_ocene"* datasourceType=*"oracle"* dbTable=*"RPGV\_VINO\_OCENE"* primarykey=*"OCENA\_ID"*/>

...

<Entity type=*"STR\_STR\_CODELIST"* name=*"tip\_vrh\_vina"* datasourceType=*"XML"* XMLFile=*"codelists.xml"* />

<Entity type=*"STR\_STR\_CODELIST"* name=*"tip\_vina"* datasourceType=*"XML"* XMLFile=*"codelists.xml"* />

...

XMLFile=*"codelists.xml"* />

</EntityDatasources>

Definiranje ekrana – Widgets

Ekran se lahko sestavi s posameznimi sklopi, grupami podatkov. Mora obstaja vsaj ena groupa podatkov. Grupo se označi z Entity

Primer:

<Widgets>

<Entity type=*"RPGV\_VINO\_OCENE"* name=*"vino\_ocene-tabela1"* tableMaxRows=*"4"*>

<EntityAttributeDefaults hidden=*"true"*/>

<EntityAttribute name=*"OCENA\_ID"* hidden=*"false"* position=*"10"*/>

<EntityAttribute name=*"VINO\_ID"* hidden=*"false"* position=*"20"*/>

<EntityAttribute name=*"ST\_VZORCA"* hidden=*"false"* position=*"60"*/>

<EntityAttribute name=*"DAT\_VZORCA"* hidden=*"false"* valueFormat=*"dd.MM.yyyy"* position=*"70"*/>

<EntityAttribute name=*"DAT\_OCENE"* hidden=*"false"* valueFormat=*"dd.MM.yyyy"* position=*"80"*/>

<EntityAttribute name=*"DAT\_DEGUSTACIJE"* hidden=*"false"* valueFormat=*"dd.MM.yyyy"* position=*"90"*/>

<EntityAttribute name=*"DAT\_ODLOCBE"* hidden=*"false"* valueFormat=*"dd.MM.yyyy"* position=*"100"*/>

<EntityAttribute name=*"OZNAKA\_ODLOCBE"* hidden=*"false"* position=*"110"*/>

<EntityAttribute name=*"NAZIV"* hidden=*"false"* position=*"120"*/>

</Entity>

</Widgets>

**Definiranje groupe podatkov – Widgets -> Entity**

**Grupa ima naslednje atribute:**

1. type – Na katere podatke (tabelo, view) se navezuje.
2. name – ime tabele ali view-a.
3. tableMaxRows – maksimalno število kolon v katerih se bodo razvrstili objekti.

**Znotraj grupe lahko definiraš:**

1. EntityAttributeDefaults – Namenjeno je definiranju lastnosti objektov, ki bodo veljale za vse definirane objekte.
2. EntityAttribute – Definicija posameznega objekta.
3. WidgetMason – Definiranje kompleksnih form, uporaba wizard-a
4. Stub – Definira praznino v tabeli, kii se jo lahko zapolni s čimerkoli, tabelo, poljem, ...
5. Div – Prostor kjer se le definira objekte, ki jih potrebuješ. Oblikovno jih lahko urediš le s CSS-jem.
6. FlexTable – Definira / uredi objekte v tabelo

**Lastnosti, ki se jih lahko določi objektom:**

1. name – Ime objekta.
2. hidden – Ali je objekt skrit. True / False
3. label – Opis vnosnega polja, če ne obstaja je to ime polja v tabeli ali view-ju
4. valueFormat – Definira se tip polja.
5. position – Pozicija polja na ekranu znotraj neke tabele. Vpliva samo na tabeli, ne na MasonWidget-u
6. readonly – Polje je samo za branje. True / False
7. valueMinimum – Minimalna vrednost, ki jo lahko uporabnik vnese
8. valueMaximum – Maximalna vrednost, ki jo lahko uporabnik vnese
9. FlexTableWidget-NextRow – Naslednje polje je v novi vrstici. True / False
10. FlexTableWidget-Colspan – Velikost polja, raztegneš polje v tabeli na max kolon v tabeli
11. FlexTableWidget-group – Označi kateri grupi polj polje pripada.
12. FlexTableWidget-groupLabel – Ime grupe polj.
13. FlexTableWidget-MaxColumns – Maksimalno število kolon v tabeli
14. Lookup – Definira tip LOV-a
15. lookupSource – Definira se izvor podatkov za LOV
16. lookupKeys – Definira ključ LOV-a, ki se ga prenese v glavno tabelo
17. lookupLabels – Imena vrednosti LOV-a.
18. lookupHasEmptyChoice – Ali lahko izbere uporabnik iz LOV tudi prazno vrednost. True / False
19. lookupTableButton – Doda gumb poleg polja za odpiranje LOV. Če gumba ni se LOV odpre z entrom. True / False
20. lookupWidget – Definira se widget, kjer je LOV definiran
21. lookupEntityType – Definira se tip entitete oz. entiteto samo (tabela ali view)
22. required – Polje obvezno.
23. large – Pomeni, da namesto text box naredi text area
24. type – Se avtomatsko nastavi iz baze. Lahko se ga ročno ponastavi. Možne vrednosti:
    1. 0 – string; tekstno polje
    2. 1 – int; intiger polje
    3. 2 – float; decimalno polje
    4. 3 – date; datumsko polje
    5. 4 – boolean; izbirno polje
    6. 99 – stub; za namen implementacije widget-a.
25. Ignore – Komplet ignorira polje, kar pomeni da niti ne prebere vrednosti iz iz baze. Na primer hidden samo skrije polje na ekranu vrednost iz baze pa prebere. True / False

EntityAttributeDefaults – Namenjeno je definiranju lastnosti atributov, ki bodo veljale za vse definirane atribute.

Widgets -> Entity -> WidgetMason -> FlexTable

Widgets -> Entity -> WidgetMason -> Div