



In deze rubriek worden specifieke onderdelen van het Oracle Application Development Framework besproken. De bedoeling is dat de lezer daarna zelf met het onderwerp aan de slag kan.



Gebruik van View Criteria

Er zijn verschillende manieren om de dataset van een viewobject te beperken. De mogelijkheid bestaat om dit met behulp van een standaard where-clause te doen. Deze where-clause schrijf je op de gebruikelijke manier in de query van het viewobject. Aan deze where-clause kun je eventueel nog bindvariabelen toevoegen zodat je iets meer flexibiliteit hebt. Nog meer flexibiliteit bereik je door gebruik te maken van view criteria.

View criteria kun je zien als filters die je kunt gebruiken om de rijen in een viewobject collectie te beperken. Ze zijn een aanvulling op de bestaande where-clause van een viewobject en worden als het ware toegevoegd aan deze where-clause. Een van de voordelen van het gebruik van view criteria naast een where-clause is het feit dat er runtime gewisseld kan worden tussen de te gebruiken view criteria.

View criteria definieer je designtime en zijn onderdeel van de definitie van een viewobject. Bij ieder willekeurig viewobject kun je view criteria definiëren. Dit kan in JDeveloper op de Query page van de viewobject editor. Omdat de view criteria die je definieert een naam moeten hebben, worden ze ook wel 'named' view criteria genoemd. Op het tabblad met UI hints kun je ook aangeven op welke manier de view criteria in de ADF query component kunnen worden gebruikt. Deze functionaliteit is erg krachtig, maar leidt er toe dat een viewobject een combinatie van twee dingen wordt. Aan de ene kant is het een business façade om data te ontsluiten, terwijl het aan de andere kant ook UI georiënteerde logica bevat, zoals de default setting voor de query component. Het is ook mogelijk om view criteria programmatisch te definiëren en te koppelen aan een view object. Er is een uitgebreide API die dit mogelijk maakt. Dit kun je gebruiken op het moment dat de viewobject editor tekort schiet. Dit zijn geen *named* view criteria.

```
.....  
ViewCriteria vc = vo.createViewCriteria();  
ViewCriteriaRow vcr = vc.createViewCriteriaRow();  
  
//Zet attribute waarden om te filteren  
vcr.setAttribute("Salaris", "> 20000");  
vcr.setAttribute("PersCode", "MNGMNT");  
  
// Voeg view criteria rij toe aan de view criteria row set  
vc.add(vcr);  
  
// Apply het view criteria  
vo.applyViewCriteria(vc);
```

In voorgaand codefragment wordt programmatisch één view criterium gemaakt en aan het view object gekoppeld. Je kunt meerdere view criteria koppelen. Default worden deze door middel van 'OR' aan elkaar gekoppeld, maar de API biedt de mogelijkheid van 'AND', 'NOT' en 'UNION' te gebruiken. Hierdoor ontstaat

een krachtig mechanisme om data te filteren op basis van programmatische view criteria.

Designtime gedefinieerde named view criteria kun je op verschillende manieren gebruiken. De meest voor de hand liggende manier is de toepassing in de <af:query/> component. Deze component kun je gebruiken om de eindgebruiker de mogelijkheid te geven gegevens te filteren. Via drag en drop vanuit de datacontrol palet kun je ze als <af:query/> component, of als 'adf filtered table' toevoegen aan je pagina's.

View criteria worden ook vaak gebruikt in UI List of Values. In deze List of Values laat je de eindgebruiker switchen tussen de vooraf gedefinieerde named view criteria, in plaats van willekeurige zoekwaarden in te geven. Een minder voor de hand liggende toepassing is het programmatisch gebruik van view criteria. Door gebruik te maken van de API kun je de designtime declaratief gedefinieerde named view criteria programmatisch koppelen aan een view object. In het volgende codefragment wordt een named view criteria opgezocht ("Functie") en wordt de waarde van een variabele gezet. De setApplyViewCriteriaNames() methode kan meerdere view criteria aan, maar in dit voorbeeld wordt er maar één gebruikt. Vervolgens kun je de gegevens via executeQuery() ophalen.

```
.....  
ViewCriteriaManager vcm = getViewCriteriaManager();  
ViewCriteria vc_functie = vcm.getViewCriteria("Functie");  
VariableValueManager vvm_functie = vc_functie.ensureVariableManager();  
vvm_functie.setVariableValue("b_functie", "MNGMNT");  
setApplyViewCriteriaNames(new String[] {"Functie"});  
executeQuery();
```

De view criteria kun je met setApplyViewCriteriaNames(null) weer verwijderen, waarna de executeQuery() weer de ongefilterde gegevens zal opleveren.

Tot slot

Met view criteria heb je als ontwikkelaar de mogelijkheid om meer controle te krijgen over de manier waarop eindgebruikers gegevens kunnen filteren. Je zult meestal de kortste weg kiezen door de named view criteria te maken in de viewobject editor en deze te gebruiken in een <af:query/> component. In de meeste gevallen is dit voldoende. Maar er zijn dus meer manieren om view criteria te maken en je kunt ze ook via de API gebruiken om data te filteren.

Luc Bors (luc.bors@amis.nl) werkt als technisch specialist/architect en is ADF Expertise Lead bij AMIS Services.