

Fleksible søk i JGrape-generert kode

v.1.0 - omb - 24/11/2005
v.1.1 - omb - 25/11/2005 (endret fra Expression til Restrictions i
criteria-eks.)
(beskrevet targetMainKey)
v.1.2 - omb - 26/01/2006 (støtte for alias i CRITERIA)

Innhold:

- 1. Angivelse av søkekriterier
2. Begrensning av resultatsett
3. Sortering av resultatsett

1. Angivelse av søkekriterier

Ved default bruk av action Find og FindAll gjøres følgende:

Find:

Det gjøres oppslag på "main key" for target-rollens klasse, og det returneres funnet forekomst evt. null dersom ingen ble funnet.

FindAll:

Alle forekomster av target-rollens klasse returneres i en liste.

Det er nå (fom. G-810Alpha9) mulig å angi andre utvalgs-kriterier. Det er støtte for 4 typer oppslag:

DEFAULT:

For FindAll benyttes ingen utvalgs-kriterier. For action Find gjøres det oppslag på main key. Default oppførsel er å benytte target-rollens objekt som nøkkelverdier (fra obtain() i klienten), men dette kan overstyres ved at man sender med et eget nøkkelobjekt. Main key angis som en property på ObjectSelection, og kan overstyres i hooks dersom man ønsker å benytte en annen nøkkel enn den som generatoren velger. Generatoren velger main key ut ifra hva som er inkludert av attributter i dialogen etter følgende prioritering:

1. Primærnøkkel
2. Første unike attributt
3. Første unike gruppe

Det er et krav at det skal eksistere en main key i dialogen for at generert kode skal virke uten overstyring.

EXAMPLE:

Oppslag via et eksempelobjekt, "query by example". Default oppførsel er å benytte target-rollens objekt til dette (fra obtain() i klienten), men dette kan overstyres ved at man sender med et eget eksempelobjekt.

QUERY:

Oppslag ved hjelp av en søkestreng. Syntaks for søkestrengen er avhengig av valgt persistensløsning, mot Hibernate benyttes HQL. Søkestreng og tilhørende parametre sendes med til service-koden. Parameterrekkefølgen er signifikant ("positional parameters").

CRITERIA:

Oppslag ved hjelp av et tre av søkeuttrykk. Innhold i treet er avhengig av valgt persistensløsning, mot Hibernate benyttes det org.hibernate.criterion.Criterion-forekomster.

Det er opprettet en ny klasse `no.genova.support.FindData` som benyttes som databærer i forbindelse med fleksible søk. `ObjectSelection`-klassen har nå en referanse til `FindData`.

Kort oppsummering av grensesnittet på `FindData`:

```
setFindMethod(EFindMethod findMethod)
- setter valgt type søk. Default type er EFindMethod.DEFAULT.

setKey(Serializable key)
- setter alternativ nøkkel for bruk ved "main key"-oppslag
  (EFindMethod.DEFAULT).

setExample(Object example)
- setter eksempelobjekt for EFindMethod.EXAMPLE.

setCriterion(Object criterion)
- setter rotnoden i treet av søkekriterier for EFindMethod.CRITERIA.

addCriterionAlias(String associationPath, String alias)
- legger til et nytt alias til bruk i EFindMethod.CRITERIA.

setQuery(String query)
- setter søkestrengen for EFindMethod.QUERY.

addQueryParameter(Object parameter)
- legger til en parameter i lista for søkestrengen.
```

Ved action `Find` vil det returneres 1 forekomst ved treff eller null hvis ingen treff. Hvis det finnes flere forekomster som tilfredsstillter søkekriteriene vil den første forekomsten returneres.

Ved action `FindAll` vil det returneres en liste med alle forekomster som tilfredsstillter søkekriteriene.

Eksempel på bruk av `FindData` fra `FindAll`-hook på klienten:

```
For alle typer:
-----
FindData findData = new FindData();
objectSelection.setFindData(findData);

QUERY:
-----
findData.setFindMethod(EFindMethod.QUERY);
findData.setQuery("from Customer as c where c.id > ?");
findData.addQueryParameter(new Integer(4));

EXAMPLE:
-----
findData.setFindMethod(EFindMethod.EXAMPLE);
Customer cust = new Customer();
findData.setExample(cust);
cust.setInformation(true);
EPay pay = new EPay();
pay.setCurrentValue(1);
cust.setPay(pay);

CRITERIA:
-----
```

```

findData.setFindMethod(EFindMethod.CRITERIA);
Criterion crit = Restrictions.gt("id", new Integer(5));
findData.setCriteria(crit);

findData.setFindMethod(EFindMethod.CRITERIA);
findData.addCriterionAlias("company", "comp");
Criterion crit = Restrictions.and(
    Restrictions.gt("id", new Integer(5),
    Restrictions.like("comp.name", "Software%"));
findData.setCriteria(crit);

```

Eksempel på overstyring av default-oppslag fra Find-hook på klienten:

```

DEFAULT:
-----
FindData findData = new FindData();
findData.setKey(new Integer(4));
objectSelection.setFindData(findData);
objectSelection.setTargetMainKey("orderLineNo");

```

2. Begrensning av resultatsett

En action FindAll vil i utgangspunktet returnere alle forekomster som tilfredsstillter søkekriteriene. Det er mulig å begrense dette ved hjelp av FindData:

```

setMaxResults(int maxResults)
- setter maksimum antall rader som returneres av et søk.

```

Klassen FindData har en static-attributt som angir applikasjonens defaultverdi for maxResults:

```

setDefaultMaxResults(int defaultMaxResults)
- setter defaultverdi for maxResults i nye FindData-objekt. Denne har
  i utgangspunktet verdien Integer.MAX_VALUE.

```

Det er mulig å få til applikasjonsstyrt "blaing" ved å hoppe over et gitt antall rader i resultatsettet:

```

setFirstResult(int firstResult)
- setter radnummer for første rad fra søket som ønskes
  returnert. Første rad i søket har nummer 0.

```

Eksempel på bruk:

```

FindData findData = new FindData();
objectSelection.setFindData(findData);
findData.setMaxResults(350);
findData.setFirstResult(50);

```

3. Sortering av resultatsett

Databasen kan settes til å sortere resultatsettet ved hjelp av sorteringsangivelser i FindData:

```

addOrder(FindOrder order)
- legger til et nytt attributt som skal inngå i sortering fra
  databasen.

```

no.genova.support.FindOrder holder sorteringsinformasjon for ett attributt:

FindOrder(String propertyName, boolean ascending)
- oppretter en ny FindOrder for angitt attributt med "ascending" eller "descending" sortering.

FindOrder.asc(String propertyName)
- oppretter en ny "ascending" sortering for angitt attributt.

FindOrder.desc(String propertyName)
- oppretter en ny "descending" sortering for angitt attributt.

setIgnoreCase(boolean ignoreCase)
- angir om denne sorteringen skal være case-sensitiv eller ei. Default er case-sensitiv.

Eksempel på bruk:

```
FindData findData = new FindData();  
objectSelection.setFindData(findData);  
findData.addOrder(FindOrder.desc("name"));  
findData.addOrder(FindOrder.asc("pay"));
```