**LLBLGen KTS**

**Basic Data Service Operations**

Gasim in clasa generica din infrastructura DataService urmatoarele metode (operatii CRUD)

* **GetById** -> select unique entity (+related graph if needed) - **READ**
  + *Read entity with Id from database, in prefetch specificam entitatile asociate (graful de obiecte)*
* *Unde credeti ca sunt folosite Save si Delete?*
* **Save** -> Add new entity & Update existing entity (+related graph if recurrent) – **INSERT + UPDATE**
  + *Insert/Update de entitati*
  + *Exemplu SaveCommand: ModelController - ExecuteSaveCommand*
* **Delete** -> logical delete! - **DELETE**
  + *La delete se apeleaza o procedura stocata care face delete-ul logic*
  + *Exemplu DeleteCommand: ModelController - ExecuteDeleteCommand*
* **GetDbCount** -> does not return select, but instead count only (more efficient)
  + *Ne returneaza un int, util cand vrem sa vedem cate date avem si nu alte informatii, este mai eficient*
* **GetEntityCollection** -> selects multiple entries (similar to GetById)
  + *Similar cu GetById, doar ca ne returneaza o colectie de entitati*

In continuare e important sa vedem la ce se refera parametrii metodelor mentionate (**PrefetchPath si RelationPredicateBucket**) mai sus:

Inainte sa facem asta, as vrea sa discutam cateva chestiuni:

* Daca va amintiti de la primul KTS, LL-ul este un ORM care isi mapeaza tabelele din db sub forma unor grafuri de obiecte bazate pe foreign key-uri.
* Acum ne punem intrebarea: daca vrem sa aducem o informatie din baza de date, de ce avem nevoie? Daca avem nevoie sa aducem studentii si persoanele asociate lor de exemplu?
  + Un punct de pornire, o **radacina**: Studentul
  + **Entitatile asociate**, in cazul asta persoana
  + Si atunci, aici intervine Prefetch Path, ne spune entitatile care trebuie aduse in graf si punctul de pornire al grafului, adica radacina -> **Deci Prefetch Path = cum arata graful**
  + Ce e important de retinut vizavi de prefetch este ca fiecare bucatica se traduce intr-un select -> LL-ul functioneaza cu select-uri individuale
  + Aici ar merita mentionat ca la laborator o sa aveti nevoie de Profiler ca sa puteti sa urmariti query-urile generate de LL
  + Sa va dau un exemplu:

|  |  |
| --- | --- |
| LL | Code |
| SELECT \* FROM Sync\_CourseTemplates  WHERE CourseTemplateID = 50 | courseTemplateDataService.GetById(50); |

Hai sa vedem elementele **PrefetchPath**-ului!

IPrefetchPath2 prefetch = new PrefetchPath2(EntityType.CourseTemplateEntity);

prefetch.Add(prefetchElement, maxNrOfElements, predicate, relationCollection, ...)

* **prefetchElement** - static property on entity = entitatea related de care avem nevoie
* **maxNrOfElements** - 0 = infinity
  + relatie 1-1 punem **1**
  + relatie 1-n punem **0** daca vrem sa luam toata colectia
* **predicate** ⬄ WHERE condition
* **relationCollection** ⬄ JOIN

1. Simple predicates

* Equality, Inequality, >, <, >= etc.
  + Exemplu cod: **GetPrefetch\_Example1**
* [Entity]Fields.Field == value (DBValue.Null example and explanation)
  + Exemplu cod: **GetPrefetch\_Example2**

Note: For deletion check – always use CommonEntityBase.GetNotDeletedPredicate<T>();

* Custom predicates – see options from VS after writing new Predicate: **GetPrefetch\_Example3**
* Ne uitam in cod la CommonEntityBase

1. Complex predicates – use predicate expressions

* AND and OR between conditions IPredicateExpression AddWithAnd(IPredicate predicateToAdd);
  + Exemplu cod: **GetPrefetch\_Example4**
* Expressions derive from predicates (classes)

**Relations** – setup joins in expressions for more complex filtering

* This enables adding fields to predicates that are on related entities specified in relations
* IRelationCollections – holds multiple relations
  + EntityRelation (pk, fk, type etc.)
  + .Add(relation, alias (optional), join hint (optional) – INNER, LEFT, RIGHT, CROSS)
  + Relation obtained from Entity.Relations property
* Exemplu cod: **GetPrefetch\_Example5**

**Sumarizare**

**How are complete queries generated?** (start from an example: **GetPrefetch\_Example2**)

* 1 prefetch node = 1 SELECT
* JOINS = Relations
* WHERE = Predicate
* TOP X = maxNrOfItems
* FROM = prefetch element

**RelationPredicateBucket** – on multiple select/count (GetEntityCollection, GetDbCount)

* Combines a *PredicateExpression* with *Relations*

Exemplu simplu cu ceva relations: **GetPrefetch\_Example6 (**Uitati selectul generat in sql query!**), RelationBucket\_Example7**

**!!Difference between prefetch and relation+predicate. E clara?**

**Complex prefetch with branching**

CT -> PG -> CI -> SA (1) -> Student -> Person

SA (2) -> SAI -> MI (with Event)

Desenez pe tabla impreuna cu ei si scriem codul (sa aiba si nod).

**Exemplu** (cu nod) aici:

var cTprefetch = new PrefetchPath2(EntityType.CourseTemplateEntity);

IPrefetchPathElement2 saNode = CourseInstanceEntity.PrefetchPathStudentAssignments;

saNode

.SubPath.Add(StudentAssignmentEntity.PrefetchPathStudent, 1, CommonEntityBase.GetNotDeletedPredicate<StudentEntity>())

.SubPath.Add(StudentEntity.PrefetchPathPersons, 1, CommonEntityBase.GetNotDeletedPredicate<PersonEntity>());

saNode.SubPath.Add(StudentAssignmentEntity.PrefetchPathStudentAssignmentInstances, 0, CommonEntityBase.GetNotDeletedPredicate<StudentAssignmentInstanceEntity>())

.SubPath.Add(StudentAssignmentInstanceEntity.PrefetchPathFirstModuleInstance, 1, CommonEntityBase.GetNotDeletedPredicate<ModuleInstanceEntity>())

.SubPath.Add(ModuleInstanceEntity.PrefetchPathEvents, 0, CommonEntityBase.GetNotDeletedPredicate<EventEntity>());

cTprefetch.Add(CourseTemplateEntity.PrefetchPathProgramGroups, 0, CommonEntityBase.GetNotDeletedPredicate<ProgramGroupEntity>())

.SubPath.Add(ProgramGroupEntity.PrefetchPathCourseInstances, 0, CommonEntityBase.GetNotDeletedPredicate<CourseInstanceEntity>())

.SubPath.Add(saNode);

return cTprefetch;

**Note**: For update & Add keep in mind that the entire graph will be persisted!

* Connections are automatically handled both ways (FK vs. PK)
* Unit of Work: poti sa salvezi si daca nu e in graful de obiecte al entitatii tale

**Aliases** – used for circular references in joins or prefetches. Arate-le parametru?

**How to run scripts in Olympus for Lab2?** Button or dump in OnAfterViewCreated()

* Faceti regiuni separate
* Faceti metode separate ca sa fie mai clean