

ORM, NHibernate и Dapper

ORM

- ◆ ORM – object-relational mapping – объектно-реляционное отображение.
- ◆ Чаще всего данные хранятся в базе данных.
- ◆ Но в коде хочется оперировать объектами.
- ◆ ORM берет на себя заботы об отображении таблиц в объекты и наоборот.
- ◆ ORM следит за объектами, умеет сохранять их изменения, создавать новые, читать из базы старые, удалять записи (то есть выполняет полный CRUD)

А нельзя обойтись без этого?

- ◆ Можно. Но наличие ORM в проекте существенно облегчает операции CRUD (про Read – чуть позже).
- ◆ Само собой, дополнительные накладные расходы => жертвуем скоростью.
- ◆ Много рефлексии => ещё жертвуем скоростью.
- ◆ Но пользы от ORM довольно много: некоторые из них умеют неплохо работать с каскадами и сами могут создавать (а некоторые и обновлять) схемы БД.

Почему NH, а не EF?

- ◆ В DDD нужно разделять различные уровни. Уровень работы с БД, само собой, должен быть выше уровня домена.
- ◆ NH позволяет естественно отделять домен.
- ◆ При использовании EF приходится учитывать его особенности при разработке домена.
- ◆ NH хорошо работает и естественно работает с каскадами.
- ◆ В EF требуется каскады подтягивать (и удалять) самому.

NHibernate

- ◆ Изначально был портирован с Hibernate'a от Java.
- ◆ Изначально все маппинги хранились в страшных XML-файлах.
- ◆ При его использовании можно испытывать сильнейшие БОЛИ.

FluentNHibernate

- ◆ Конфигуратор для NHibernate.
- ◆ Позволяет довольно простым кодом регистрировать маппинги (без XML'а!).
- ◆ Поддерживает так называемые конвенции, по которым будут именоваться таблицы и поля схемы БД.
- ◆ Рассмотрим использование NH с FluentNH в рамках какого-нибудь простенького проекта.

Сравнительная таблица NH и массажистки

Фича	NHibernate	Массажистка
Правильно умеет считать сгруппированные объекты	-	+
Умеет брать различные объекты из отсортированного множества	-	+
Умеет объединять объекты одинаковой структуры (UNION)	-	+
LEFT JOIN с ограничениями	-	+
Сосёт	+	+

Слой чтения — болезненное место NH

- ◆ NHibernate ОЧЕНЬ плохо подходит для задач составления сложных запросов через LINQ. В таких случаях проще писать plain-sql.
- ◆ Но тут возникает еще одна проблема: а во что (а еще очень часто — как?) маппить результат?

Лечим боли с помощью Dapper

- ◆ Dapper – micro ORM от ребят, которые сделали StackOverflow (и на котором он собственно и работает).
- ◆ Предоставляет несколько расширений для IDbConnection, которые позволяют мапить результаты SQL-запроса в классы.
- ◆ IL-код мапперов формируется в рантайме и кэшируется – хороший прирост к скорости!