



Метод AsNoTracking

Последнее обновление: 31.10.2015



Когда контекст данных извлекает данные из базы данных, Entity Framework помещает извлеченные объекты в кэш и отслеживает изменения, которые происходят с этими объектами вплоть до использования метода `SaveChanges()`, который фиксирует все изменения в базе данных. Но нам не всегда необходимо отслеживать изменения. Например, нам надо просто вывести данные для просмотра.

Чтобы данные не помещались в кэш, применяется метод **AsNoTracking()**. При его применении возвращаемые из запроса данные не кэшируются. А это означает, что Entity Framework не производит какую-то дополнительную обработку и не выделяет дополнительное место для хранения извлеченных из БД объектов.

Метод `AsNoTracking()` применяется к набору `IQueryable`:

```
using(BookContext db = new BookContext())
{
    IEnumerable<Book> books1 = db.Books.AsNoTracking().ToList();
    IEnumerable<Book> books2 = db.Books
        .Where(b => b.Price > 200)
        .AsNoTracking().ToList();
    IEnumerable<Book> books3 = db.Books
        .Include(b=>b.Author)
        .AsNoTracking().ToList();
}
```

Небольшой пример. У нас в базе данных есть несколько моделей `Phones`:

```
db.Phones.Add(new Phone { Name = "Samsung Galaxy Note" });
db.Phones.Add(new Phone { Name = "iPhone 6" });
db.SaveChanges();
```

При обычном выполнении:

```
Phone firstPhone = db.Phones.FirstOrDefault();
firstPhone.Name = "Samsung Galaxy Ace 2";
db.SaveChanges();

List<Phone> phones = db.Phones.ToList();
```

Мы увидим, что в наборе `phones` первый элемент имеет название "Samsung Galaxy Ace 2".

Но если бы мы использовали `AsNoTracking`, то результат был бы другой:

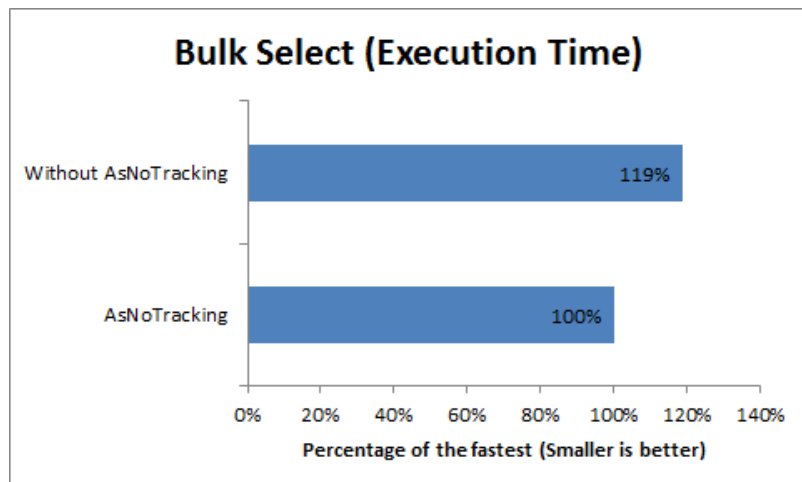
```
Phone firstPhone = db.Phones.AsNoTracking().FirstOrDefault();
firstPhone.Name = "Samsung Galaxy Ace 2";
db.SaveChanges();

List<Phone> phones = db.Phones.AsNoTracking().ToList();
```

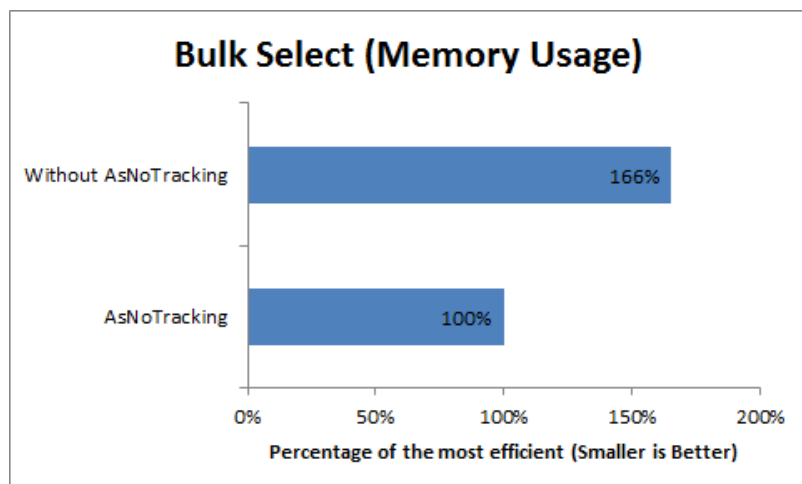
Так как при получении первого элемента используется AsNoTracking, он не будет отслеживаться, и поэтому вызов `db.SaveChanges()` никак не повлияет на базу данных, а первый элемент сохранит свое первоначальное значение - "Samsung Galaxy Note".

Если мы обратимся к тестам сравнения простого запроса и запроса с использованием AsNoTracking, то мы увидим преимущество как по скорости выполнения, так и по использованной памяти.

Сравнение по скорости выполнения



Сравнение по использованию памяти



Правда, различия будут заметны на больших объемах данных в сотни и тысячи объектов.

Когда следует использовать AsNoTracking? Если нам надо просто вывести данные для отображения без необходимости их дальнейшего обновления, тогда как раз тот случай, когда мы можем использовать AsNoTracking.

[Назад](#) [Содержание](#) [Вперед](#)



Стилизатор изображений
Ашманова. ✓

ashmanov.net

Яндекс.Директ

[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.