

## 第47期-实现仓储批操作与导航属性加载

2021年11月3日 21:10

### 仓储实现批量操作

`IRepository<TEntity>` 新增:

```
Task InsertManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default);
```

```
Task UpdateManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default);
```

```
Task DeleteManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default);
```

`EfCoreRepository<TDbContext, TEntity>` 新增:

```
public async Task InsertManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default)
{
    await DbSet.AddRangeAsync(entities, cancellationTokens);

    if (autoSave)
    {
        await _dbContext.SaveChangesAsync(cancellationTokens);
    }
}
```

```
public async Task UpdateManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default)
{
    DbSet.UpdateRange(entities);

    if (autoSave)
    {
        await _dbContext.SaveChangesAsync(cancellationTokens);
    }
}
```

```
public async Task DeleteManyAsync(IEnumerable<TEntity>entities, bool autoSave
=false, CancellationTokens cancellationTokens =default)
{
    DbSet.RemoveRange(entities);

    if (autoSave)
    {
```

```

        await _dbContext.SaveChangesAsync(cancellationToken);
    }
}

```

**IRepository<TEntity, TKey> 新增:**

```

Task DeleteManyAsync(IEnumerable<TKey>ids, bool autoSave = false, CancellationToken
cancellationToken = default);

```

**EfCoreRepository<TDbContext, TEntity, TKey> 新增:**

```

public virtual async Task DeleteManyAsync(IEnumerable<TKey>ids, bool autoSave
=false, CancellationToken cancellationToken = default)
{
    var entities = await base.DbSet.Where(x =>
ids.Contains(x.Id)).ToListAsync(cancellationToken);

    await DeleteManyAsync(entities, autoSave, cancellationToken);
}

```

## EF Core 批量操作性能优化

### Entity Framework Core 中实现批量更新和删除

#### 使用第三方扩展库实现批量更新和删除

<https://github.com/dotnet/efcore/issues/795#issuecomment-303995659>

## 灵活加载关联的导航属性数据

```

Task<IQueryable<TEntity>> IncludeRelatedAsync(params Expression<Func<TEntity, object>>
[] propertySelectors);

```

```

Task LoadRelatedAsync<TProperty>(TEntity
entity, Expression<Func<TEntity, IEnumerable<TProperty>>>
propertyExpression, CancellationToken cancellationToken = default) where
TProperty : class;

```

```

Task LoadRelatedAsync<TProperty>(TEntity entity, Expression<Func<TEntity, TProperty>>
propertyExpression, CancellationToken cancellationToken = default) where
TProperty : class;

```

### EF Core 懒加载与延迟加载

## 扩展技术点

- 使用部分类和扩展方法分割文件,
- 常见命名规范说明。

驼峰命名法: camelCase

帕斯卡写法: PascalCase

蛇形命名法: snake\_case