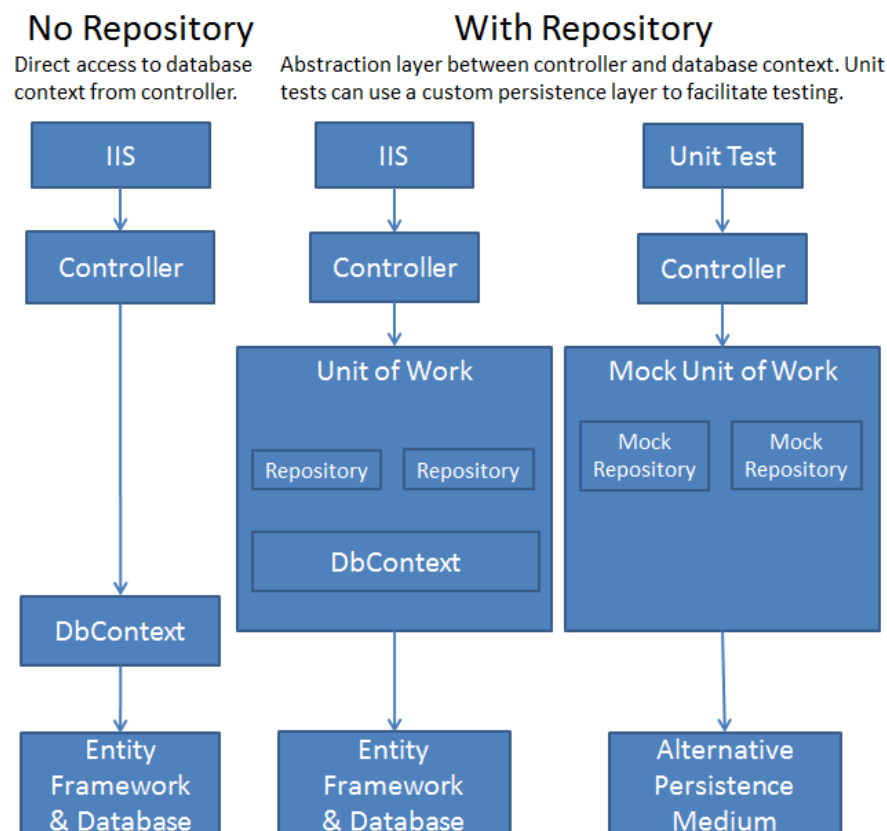


# 第49期-仓储与工作单元模式

2019年7月16日 15:45

## 基本概念

仓储和工作单元模式旨在创建数据访问层和应用程序的业务逻辑层之间的抽象层，实现这些模式可让你的应用程序对数据存储介质的更改不敏感，而且很容易进行自动化单元测试和进行测试驱动开发 (TDD)。



有许多方法来实现存储库和工作单元模式，接下来我们基于 EF Core 实现最通用的仓储和工作单元模式，也可以说是仓储和工作单元模式的最佳实践。

## 开源框架扩展

### CatFactory

CatFactory 是一种面向 .NET Core 的基架引擎，它可自动基于 SQL Server 数据库生成 DbContext 类、实体、映射配置和存储库类。 [GitHub 存储库](#)

### Threenine.Data

基于 Entity Framework Core 框架的通用存储库与工作单元模式实现，最初的灵感来自撰写博客文章 [Generic Repository Pattern .NET Core](#)，该项目已经开始具有依赖注入功能的可重用工作包单元。 [GitHub 存储库](#)

### Microsoft.EntityFrameworkCore.UnitOfWork

Microsoft.EntityFrameworkCore 的一个插件，它支持存储库、工作模式单元，并支持多个具有有所支持分布式事务的数据库。[GitHub 存储库](#)

## 开源项目中的应用

[ASP.NET Boilerplate Unit Of Work](#)

[nopCommerce Repository Pattern](#)

## 相关阅读

[Implementing the Repository and Unit of Work Patterns](#)

[Using the Repository and Unit Of Work Pattern in .net core](#)

[The Repository Pattern](#)

[仓储\(Repository\)和工作单元模式\(UnitOfWork\)](#)