

Лабораторная работа №4

Основы реляционных СУБД. MS SQL Server. MS SQL Server Management Studio. ADO.NET.

EntityFramework (OPTIONAL)

Введение:

В предыдущей лабораторной работе вы разработали менеджер поставки конфигурации для сервиса. Ваш сервис выполнял манипуляции с файлами, но файлы не содержали данных. Теперь этот момент мы поправим!

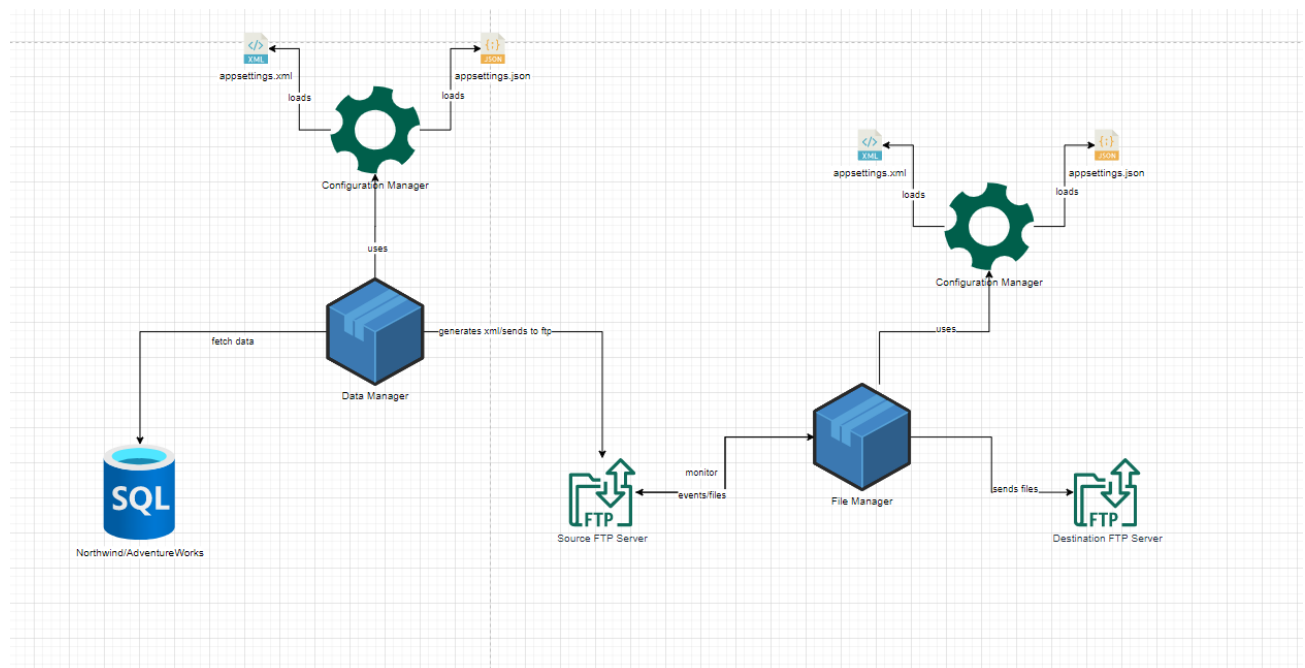
Постановка задачи:

Необходимо реализовать новую службу, для простоты назовём её **Data Manager**.

Основные возможности Data Manager:

1. Извлекать данные из базы данных MS SQL Server.
2. Сгенерировать на основе имеющихся данных XML файл/файлы необходимой структуры.
3. Отправить его на FTP сервер.

Максимально упрощённо схема выглядит так:



Требования:

Стек технологий/библиотек для решения задачи.

База данных - MS SQL Server 2012+.

Технология для работы из .NET – ADO.NET

Поставщик конфигурации – пакет ConfigurationManager.

Конфигурационные файлы – XML/JSON.

А теперь непосредственно к требованиям.....

1. Необходимо развернуть сервер базы данных MS SQL. Восстановить БД с помощью одного из backup-ов (<https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15> , <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/linq/downloading-sample-databases>)
2. Выбрать набор из **минимум 5** отношений (таблиц/сущностей), на основании данных которых будет формироваться XML файл.
3. Разработать необходимые **хранимые процедуры** для работы с отношениями (таблицами).
4. Разработать слой работы с базой данных (Data Access Layer). API этого слоя представляет собой конечный набор интерфейсов с набором публичных методов для чтения данных.
5. Разработать слой сервисов (Service Layer) для работы с DAL. API этого слоя представляет собой конечный набор интерфейсов с набором публичных методов.
6. Выставить DataAccess и ServiceLayer приминительно к используемой БД как отдельные подключаемые пакеты (зависимости).
7. Все используемые модели выставить как отдельный пакет (зависимость), шарящийся между всеми нуждающимися проектами (**исключая модели конфигурации служб**).
8. Разработать сервис для генерации XML документа. (IXmlGeneratorService). Так же сервис должен генерировать XSD схему и отсылать вместе с файлом.
9. Разработать сервис для отправки файлов на FTP сервер. (IFileTransferService).

10. Конфигурацию **Data Manager** представить в виде модели .NET, и получать конфигурацию с использованием уже разработанного пакета **ConfigurationManager**.
11. Конфигурацию **Data Manager** предоставлять в виде JSON и XML файла (предполагая, что если не передан путь к конфигурационному файлу, то файл находится рядом с исполняемым файлом **Data Manager**).
12. В вашем решении должно присутствовать МИНИМУМ 6 проектов.
 1. AdventureWorks/Northwind.DataManager
 2. AdventureWorks/Northwind.FileManager
 3. AdventureWorks/Northwind.ConfigurationManager
 4. AdventureWorks/Northwind.DataAccess.
 5. AdventureWorks/Northwind.ServiceLayer.
 6. AdventureWorks/Northwind.Models
13. Разработать отдельную базу данных для логирования AdventureWorks/Northwind.ApplicationInsights, которая будет содержать в себе разработанную ВАМИ (или слезанную из существующих логгеров) структуру логирования. Уровни логов, сообщения, исключения, время и т.д. (т.е. ВСЯ необходимая информация, чтобы понять на каком этапе и где произошла ошибка).
14. Разработать уровень DAL, SL для работы с AdventureWorks/Northwind.ApplicationInsights наподобии имеющегося решения.
15. Все операции с базами данных выполняются в рамках **TransactionScope** с использованием **хранимых процедур** (никаких голых запросов в строках).
16. **ОБЯЗАТЕЛЬНО!!!!!!!!!!!!!!!** Разработать АДЕКВАТНУЮ структуру папок в каждом проекте. Не пихайте всё в один namespace. Чётко разделите интерфейсы и реализации, модели и конфигурационные файлы.
17. **ОБЯЗАТЕЛЬНО!!!!!!!!!!!!!!!** Никакой СТАТИКИ ни в каком ВИДЕ (за исключением методов расширения, но только там, где они действительно нужны и вы сможете пояснить). Все классы должны быть экземплярные, как и методы (судя по вашим лабораторным работам вы слишком полюбили статику и забыли, что работаете в среде ООП, а не в процедурной).

ВАЖНО:

Невыполнение пунктов 16, 17 означает следующее – останавливается защита и вы уходите переделывать.

ЗАПРЕЩЕНО использование ORM-библиотек.

ЗАПРЕЩЕНО использование автомапперов в любом виде (только если самописный).

ЗАПРЕЩЕНО использование готовых библиотек логирования Serilog, Log4Net и т.д.

Литература:

1. <https://metanit.com/sharp/adonet/>
2. <https://metanit.com/sharp/mvc5/23.5.php>
3. Ваши любимые Рихтеры, Албахари. (источники у вас есть).

Удачи!