

SqlRepositoryImpl 소개

2010년 8월 11일 수요일

오후 1:50

SqlRepositoryImpl 는 MS SQL Server 전용의 DAAB를 이용한 AdoRepositoryImplBase 의 Concrete class 입니다.

SQL Server 전용이므로, 다른 RDBMS에서는 사용이 불가능하지만, 비동기 IO 작업을 수행하므로, 기존 동기 방식의 작업보다 훨씬 확장성이 좋아졌습니다.

일례로 기존 AdoRepositoryImpl 사용 시 100개의 스레드로 테스트 시에 Timeout이 걸려버렸습시다만, SqlRepositoryImpl을 이용한 비동기 IO 작업을 하게되면 200개의 스레드를 동시에 실행시켜도, 아무런 문제없이 작업을 수행합니다.^.^

적용 방법도 쉽습니다. IoC의 장점이랄까요...

기존 AdoRepository를 사용하는 제품에서는 IoC 환경 설정에서 다음과 같이 AdoRepositoryImpl 을 SqlRepositoryImpl로 변경만 하면 됩니다.

```
<component id="NorthwindAdoRepository"
  service="RCL.Data.Ado.IAdoRepository, RCL.Data"
  type="RCL.Data.Ado.AdoRepositoryImpl, RCL.Data">
  <parameters>
    <!-- dbName을 설정하지 않으면 DAAB의 기본 DB 사용 -->
    <!--<dbName>Northwind</dbName>-->
    <QueryStringProvider>${IniAdoQueryStringProvider.Northwind}</QueryStringProvider>
  </parameters>
</component>
```

RCL.Data.Ado.Sql.SqlRepositoryImpl 로 변경하시면 됩니다.

참 쉽죠 잉^^

아래 코드는 SqlRepositoryImpl class의 본체입니다. 비동기 IO 작업은 SqlCommand에서 ExecuteNonQuery, ExecuteReader, ExecuteXmlReader 만 지원합니다만, ExecuteReader를 통해 ExecuteScalar를 구현할 수 있으니 총 3개의 함수만 재정의하고, XmlReader 관련만 추가하면 됩니다. (이게 상속의 힘)

```

/// <summary>
/// Microsoft SQL Server 전용의 sAdoRepository입니다.
/// 비동기 IO 작업을 통해 더 빠른 작업을 수행할 수 있도록 해 줍니다.
/// </summary>
public class SqlRepositoryImpl : AdoRepositoryImplBase, IAdoRepository
{
    public SqlRepositoryImpl() { }
    public SqlRepositoryImpl(string dbName) : base(dbName) { }

    public new SqlDatabase Db
    {
        get { return (SqlDatabase)base.Db; }
    }

    protected override object ExecuteScalarInternal(DbCommand cmd)
    {
        CheckIsSqlCommand(cmd);

        using (var task = Db.ExecuteScalarAsync(cmd as SqlCommand))
        {
            return task.Result;
        }
    }

    protected override int ExecuteNonQueryInternal(DbCommand cmd)
    {
        CheckIsSqlCommand(cmd);

        using (var task = Db.ExecuteNonQueryAsync(cmd as SqlCommand))
        {
            return task.Result;
        }
    }

    protected override IDataReader ExecuteReaderInternal(DbCommand cmd)
    {
        CheckIsSqlCommand(cmd);

        using (var task = Db.ExecuteReaderAsync(cmd as SqlCommand))
        {
            return task.Result;
        }
    }
}

```

화면 캡처: 2010-08-11 오후 2:03