namespace Sample.Repository.ADONET

{

public class EmployeeRepository : IEmployeeRepository

{

private string \_connectionString;

public string ConnectionString

{

get { return \_connectionString; }

set { \_connectionString = value; }

}

public EmployeeRepository(string connectionString)

{

if (!string.IsNullOrWhiteSpace(connectionString))

{

this.\_connectionString = connectionString;

}

}

/// <summary>

/// Gets the one.

/// </summary>

/// <param name="id">The id.</param>

/// <returns></returns>

/// <exception cref="System.NotImplementedException"></exception>

public Employee GetOne(int id)

{

string sqlStatement = "select \* from Employees where EmployeeID = @EmployeeID";

Employee item = new Employee();

using (SqlConnection conn = new SqlConnection(this.ConnectionString))

using (SqlCommand comm = new SqlCommand(sqlStatement, conn))

{

comm.Parameters.Add(new SqlParameter("EmployeeID", id));

if (conn.State != ConnectionState.Open) conn.Open();

using (IDataReader reader = comm.ExecuteReader())

{

if (reader.Read())

{

for (int i = 0; i < reader.FieldCount; i++)

{

PropertyInfo property = item.GetType().GetProperty(reader.GetName(i));

if (property != null && !reader.GetValue(i).Equals(DBNull.Value))

{

ReflectionHelper.SetValue(property.Name, reader.GetValue(i), item);

}

}

}

}

}

return item;

}

/// <summary>

/// Gets the employees.

/// </summary>

/// <returns></returns>

/// <exception cref="System.NotImplementedException"></exception>

public IEnumerable<Employee> GetEmployees()

{

List<Employee> employees = new List<Employee>();

string sqlStatement = "select \* from Employees order by EmployeeID";

using (SqlConnection conn = new SqlConnection(this.ConnectionString))

using (SqlCommand command = new SqlCommand(sqlStatement, conn))

{

command.CommandType = CommandType.Text;

command.CommandTimeout = 180;

if (conn.State != ConnectionState.Open) conn.Open();

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

Employee item = new Employee();

for (int i = 0; i < reader.FieldCount; i++)

{

PropertyInfo property = item.GetType().GetProperty(reader.GetName(i));

if (property != null && !reader.GetValue(i).Equals(DBNull.Value))

{

ReflectionHelper.SetValue(property.Name, reader.GetValue(i), item);

}

}

employees.Add(item);

}

}

}

return employees;

}

}

}

在 GetOne 與 GetEmployees 這兩個方法都是向資料庫執行 SQL Statement 後取得資料，取得資料後再去對應到我所定義的 Employee 類別的屬性，這邊並不是使用一個個欄位對應屬性