**Evolent Contacts Service**

Goal

Design and implement a production ready application for maintaining Evolent contact information.

Technologies & Frameworks

* C#
* Web API
* Entity Framework (code first approach)
* LINQ
* SQL Server
* MOQ

Patterns

* Dependency Injection
* Repository Pattern

Instructions to run the service

* Pull code base using below Git repository
* Run below SQL scripts from App\_Data folder to create database and seed test data
* Either create a virtual directory to run application for local IIS or choose IIS Express from solution properties. Below are the URL’s for GET operations of the service (running from Local IIS)
* Run all test methods within the solution to verify functionality using TDD approach

Solution structure and explanation

* Solution Name: EvolentContacts
* Projects: contains below two projects representing code base and test cases of service operations
  + EvolentContacts – Web API project contains the code base to build the functionality
    - App\_Data: contains supportive SQL files
      * CreateDatabase.sql – SQL to create the database
      * SeedData.sql – SQL to insert test data into the database table
    - App\_Start: contains configuration files
      * WebApiConfig.cs: contains Route and DI Unity Resolvers
      * UnityResolver.cs: class implements IDependencyResolver interface
      * BrowserJsonFormatter.cs: JSON formatter support class
    - Controllers: controller class(es) exposing service operations
      * EvolentContactController.cs: contains service operations
    - Helpers: contains custom helper classes
      * RequiredEitherValidator.cs: validates either value of properties
    - Migrations: contains EF code first migration files
    - Models: contains POCO objects
      * EntityBase.cs: abstract base class with common properties
      * EvolentContact.cs: Entity representing EvolentContact table in database
      * EvolentContactContext.cs: DbContext class with Connection and DbSet definitions
    - Repositories: contains required repository interfaces and classes
      * Interfaces/IEvolentContactRepository.cs: interface
      * EvolentContactRepository.cs: repository with LINQ queries to interact with data
    - Web.config: contains all configuration settings for the solution
  + EvolentContactTests – Test project containing the test cases for the functionality
    - EvolentContactTests.cs: contains TestMethods to test service operations
    - App.config: contains configuration setting definitions