**Data Access Layer Centralization API using ADO.net**

**Instruction Manual**

**Author:** Sonny R. Recio

**Description:** This API simplifies the process of using ADO.net in a more organized way, allowing developers to create a Data-driven applications more rapidly than ever before.

**Instructions:**

The following are the set of steps on how to use the API:

* Call **InitializeDataAccess()** method to make the API work. The method has three parameters:
  + - * ProviderType – Specifies which provider your database uses. In the meantime, only 3 providers were currently supported. ***Required***
      * ConnectionString – Specifies Connection String for the database. ***Required***
      * Query – Parameter for your query. ***Optional***
  + If you want to display the query you’ve provided, you can use **getDataTable()**. The method has 3 parameters:
    - * tableIndex – Specifies which table you want to retrieve. If you have 2 tables, it contains an array of tables starting from index 0. Otherwise you don’t need to worry about it. ***Optional***
      * SelectQuery – Specifies a query which retrieves data if you haven’t specified a query during InitializeDataAccess(). ***Optional if InitializeDataAccess() method has query***
      * CommandType – Specifies if you’re using a query directly at the code or using a Stored Procedure. This gives a support for Stored Procedures if you are using one. It’s set in CommandType.Text by default. ***Optional***

*\*\*note: you can also use* ***getDataSet()*** *method but* ***getDataTable()*** *is much recommended to use.*

* If you want to **Add**, **Edit**, **Delete** records, make use of SaveChanges() method. It has 2 parameters:
  + - Query – Specifies your query. ***Optional if InitializeDataAccess() method has query***
    - CommandType - \*\*refer to 2nd Step\*\*. ***Optional***

*\*\*note: to avoid SQL injection within your query, make use of CreateCommandParameters() method.*

**Updates and features:**

***Version 1.0***

* Support for multiple providers(MS Access, SQL Server, ODBC Connections)
* Support for Stored Procedures
* Automatically handled Connections and garbage collections
* Added support for DataSet as data retrieval
* Added support for DataTable as data retrieval
* Automatically processed parameters for SQL Commands
* Auto-execute commands for CRUD(Create, read, update, delete) operations
* Support for Scalar commands that returns single value