**Intelligent Reach API by Emmanouil Gimissis MSc ICSD MCP**

**Bournemouth 26/9/2021**

**Project is located at https://github.com/elrod68/DataImporter**

1. **Components/libraries used**

Visual Studio 2019 and NET 5.0 were used along with Newtonsoft JSON, Swagger. SQL Server 2016 hosts the database.

xUnit is used for tests.

1. **General structure and operation**

Chose to use SQL Server Bulk Copy to insert the records. BCP is the most performant library for this task, although it required a tiny amount of extra work to add records to a temp table (FeedData) and then move them to Products.

StreamReader is used for this process in order to conserve memory.

Contrary to what is specified in the requirements, added companyID in Feeds table, so no need to include companyID in Products, feedID includes companyID.

Retained companyID parameter in ProductService GetProduct method, for compatibility purposes, could be removed and use only feedID.

ADO.NET queries are used for speed and simplicity reasons, a stored procedure and view (sps\_Products, VW\_ProductsGeneral) are used to get products for the same reasons. Entity Framework is several times slower than ADO and for a high performing simple service as a this, the overhead is not justified by EF advantages.

Import is activated from Console project, does not run automatically every time the API requests data. Running import every time the API is queried is not optimal and usually import is a scheduled process.

Import deletes all data present before importing again, this could be modified if needed.

1. **Setup**

Please restore the provided SQL Server DB to a local SQL Server 2016 or later and modify connection strings in application.json and basePath constant in Console app and test project.

1. **Possible improvements**

Add logging info in the database, for example an import log table and date/time imported in Product records.

Obviously, SQL server DB and Web API/service could be moved to Azure.