# Read Me Doc

## Shop Bridge

#### Version 1.0

**Table of Contents**

[Read Me Doc 1](#_Toc73033001)

[Shop Bridge 1](#_Toc73033002)

[1. Purpose 3](#_Toc73033003)

[2. Summary of Business Requirements 3](#_Toc73033004)

[Exception handlings need to be present. 3](#_Toc73033005)

[3. Project Architecture 4](#_Toc73033006)

[4. Installation Instructions 5](#_Toc73033007)

# Purpose

The purpose of this document is to summarise the installation instruction for the system.

# Summary of Business Requirements

**Create a backend solution for Product functionality to be used by Product**

**Admin, and perform below actions:**

1. Add a new item to the inventory (The item should require a name, description, and price

as basic fileds, think of additional information that would be useful)

2. Modify an item in the inventory.

3. Delete an item from the inventory.

4. Lists the items in the inventory.

 Use a relational (SQL Server / MSSQL / Postgres) database and any ORM you feel

comfortable with for any persistent data storage needs. All of the functions of

the backend need to be served by an API call and the store’s inventory should be persisted

across restarts of the backend process.

 All API calls should be asynchronous.

 Include data validations as and when needed.

##  Exception handlings need to be present.

# 

# Project Architecture

Technologies

1. Visual Studio 2019
2. Sql Server Express 2019
3. Entity Framework

**Logging and Exception Handling**

Exception Handling – Global exception handling has been provided and on error Log will be created at application “Logs” folder.

Logging has been implemented using **Log4Net**

Database –

Database Context has been provided through Entity Framework and entities has been generated using Poco generator and extended to be used between different layers.

Database Context has been implemented using tradition SqlClient components like SqlDataReader and SqlCommand

UI Application (Not part of the requirement)

ShopBridge.Service

DAL(Entity Framework) and Repository

# Installation Instructions

Source code consists of 2 Major Project Files

1. **UI –** As per the requirement this is not included but we can have ASP.net / MVC Appliation/ Console/Windows/WPF application. It will call ShopBridge.Service APi to perform Crud Operation.
2. **ShopBridge.Service** – Rest API Service with CRUD DAL Operations

**Controller** -> **Model(DAL Operations)**

**Controllers – API Calls**

* InventoryController –> Controller which calls InventroyService which performs traditional DAL Operation using DataReader and SqlCommand
* InventoryControllerUsingEFF –> Controller which calls InventroyServiceUsingEff which perfroams DAL using Entity Framework

**Model – DAL Operations**

* IInventoryService – Interface for CRUD Operation
* InventoryService: **IInventoryService** – Service Class for Traditional DAL Operation using DataReader and SqlCommand
* InventoryServiceUsingEFF: **IInventoryService** – Controller for DAL using Entity Framework

**Currently DAL and Entities are included in same project but we can have a separate project for these two.**

1. **ShopBridgeTests** – Unit Testing Framework

**Controllers**

* InventoryControllerTests – Unit tests for functions defined in **ShopBridge.Service\Controller->** InventoryController
* InventoryControllerUsingEffTests - Unit tests for functions defined in **ShopBridge.Service\Controller->** InventoryControllerUsingEFF

1. **Config file changes in both the projects**

**ShopBridge.Service** -> Web.config “ShopBridgeEntities” connectionString for providing database information marked in green.

<appSettings>

<add key="ShopBridgeConnString" value="Server=xxx;Database=xxx;Trusted\_Connection=True;" />

</appSettings>

<connectionStrings>

<add name=" ShopBridgeEntities" connectionString="metadata=res://\*/ShopBridgeDAL.csdl|res://\*/ShopBridgeDAL.ssdl|res://\*/ShopBridgeDAL.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=xxx;initial catalog=ShopBridge;user id=xxx;password=xxx;multipleactiveresultsets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

**ShopBridge.Tests** -> App.config “ShopBridgeEntities” connectionString for providing database information marked in green.

<appSettings>

<add key="ShopBridgeConnString" value="Server=xxx;Database=xxx;Trusted\_Connection=True;" />

</appSettings>

<connectionStrings>

<add name=" ShopBridgeEntities" connectionString="metadata=res://\*/ShopBridgeDAL.csdl|res://\*/ShopBridgeDAL.ssdl|res://\*/ShopBridgeDAL.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=xxx;initial catalog=ShopBridge;user id=xxx;password=xxx;multipleactiveresultsets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

1. Generate Database scripts located at “ShopBridge\ShopBridgeDBScripts\ ShopBridgeDB.sql” to create the database. If required change the database path mentioned in green below in “DatabaseScript.sql” file as CREATE DATABASE [ShopBridge] ON PRIMARY

( NAME = N'ShopBridge', FILENAME = N'c:\Program Files\Microsoft SQL Server\MSSQL10\_50.SQLSERVER\MSSQL\DATA\ShopBridge.mdf' , SIZE = 2304KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'ShopBridge\_log', FILENAME = N'c:\Program Files\Microsoft SQL Server\MSSQL10\_50.SQLSERVER\MSSQL\DATA\ShopBridge\_log.LDF' , SIZE = 576KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)