SIF Framework (.NET)

Version 6.0.0

Setup Guide

**Author:** , SIF Solutions Architect

**Revision:** 2.0

**Published:** May 2022

Copyright © 2022, Systemic Pty Ltd

Table of Contents

1. Introduction 3

2. Download the SIF Framework 3

3. Set up the SIF Framework 3

3.1. Open the Solutions in Visual Studio 3

3.2. Create an Environment 3

3.3. Start the Environment Provider 4

3.4. Start the demo student Provider 4

3.5. Run the demo student Consumer 5

4. Review expected behaviour 5

# Introduction

This document outlines the steps necessary to download and set up the SIF Framework. Upon the successful completion of the instructions outlined, the SIF Framework will be correctly set up for use.

# Download the SIF Framework

The .NET version of the SIF Framework can be downloaded from GitHub at the following URL.

<https://github.com/nsip/Sif3Framework-dotNet>

Download or Clone to an appropriate directory.

# Set up the SIF Framework

The SIF Framework is made up of multiple Solutions that can be found under the *Code* directory. For the demo projects to be run, the Sif3Framework and Sif3FrameworkDemo Solutions need to be loaded into Visual Studio and built. When loading for the first time, Visual Studio may require time to restore referenced NuGet packages.

## Open the Solutions in Visual Studio

Open and build both the Sif3Framework and Sif3FrameworkDemo Solutions in Visual Studio (separate instances). No additional configuration should be required to run the demo projects.

## Create an Environment

Before a Service Consumer and Object Service Provider can interact, an Environment must be created to manage the session information between them. Once an Environment has been defined, then the Service Consumer and Object Service Provider are able to register to that Environment. This registration is managed by an Environment Provider.

Environment definition generally falls under the domain of a SIF Administrator. However, creation of an initial Environment for these demos is performed by running the *Scripts\BAT\Demo execution\NetCore\DemoSetup.bat* script. This script runs the Sif.Framework.Demo.NetCore.Setup Project to create and populate a demo database with an appropriate Environment definition.

An SQLite database is used (with a relative path configured for the setup project) to enable this demo to run out of the box. All mandatory data required by the Consumer and Provider (e.g., applicationKey, sharedSecret) has been predefined for the demo projects.

On successful completion of the script, the following output should be displayed. Once run, it can be shut down.

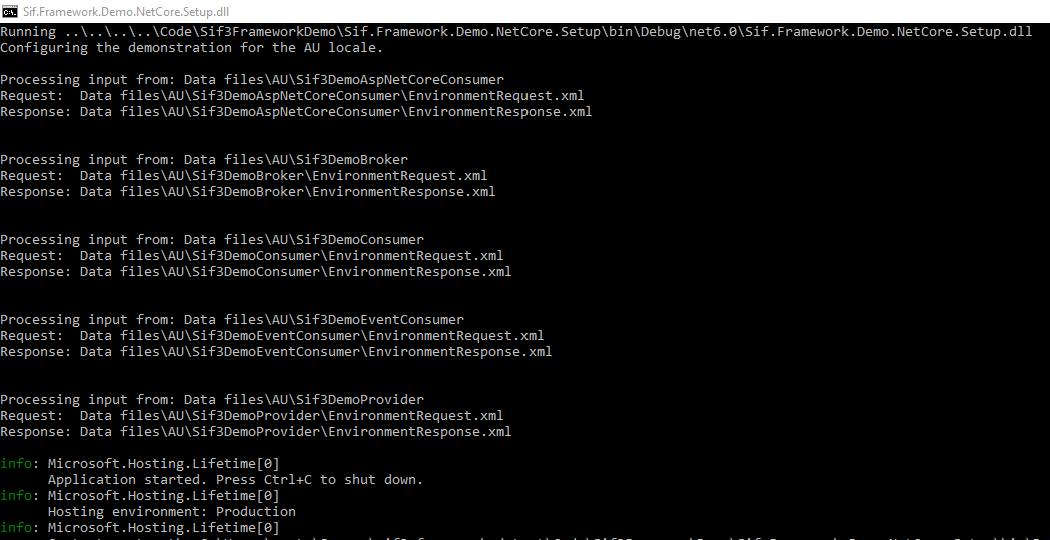


Figure 1: Demo set up

## Start the Environment Provider

In the Sif3Framework Visual Studio instance, ensure that the Sif.Framework.AspNetCore.EnvironmentProvider Project is set as the single start-up project and run it. If successful, a web browser page will open as follows.

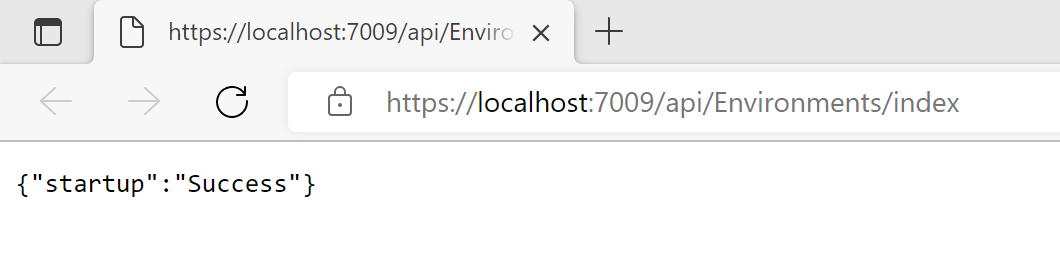


Figure 2: Successful Environment Provider start up

The Environment Provider project has been configured to run on port 7009 for HTTPS (as specified in launchSettings.json). This port is referenced in the Environment definition configured from section 3.2 Create an Environment.

## Start the demo student Provider

In the Sif3FrameworkDemo Visual Studio instance, ensure that the Sif.Framework.Demo.Provider Project is set as the single start-up project and run it. If successful, a web browser page will open as follows.

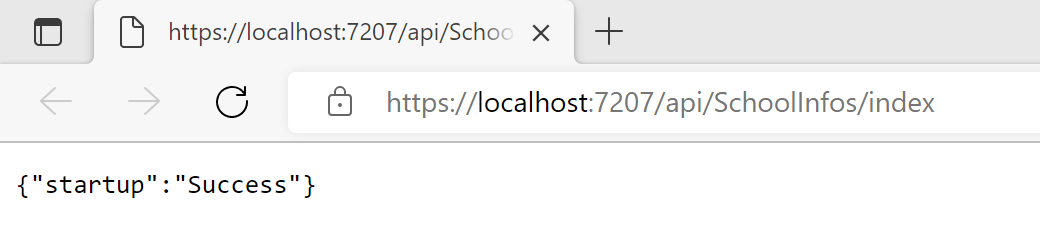


Figure 3: Successful demo student Provider start up

The demo student Provider project has been configured to run on port 7207 for HTTPS (as specified in launchSettings.json). This port is referenced in the Environment definition configured from section 3.2 Create an Environment.

## Run the demo student Consumer

Once the Environment Provider and demo student Provider have been successfully started, run the *Scripts\BAT\Demo execution\NetCore\DemoConsumer.bat* script to start the demo student Consumer. This script runs the Sif.Framework.Demo.Consumer Project of the Sif3FrameworkDemo Solution.

On successful completion of the script, the following output should be displayed.

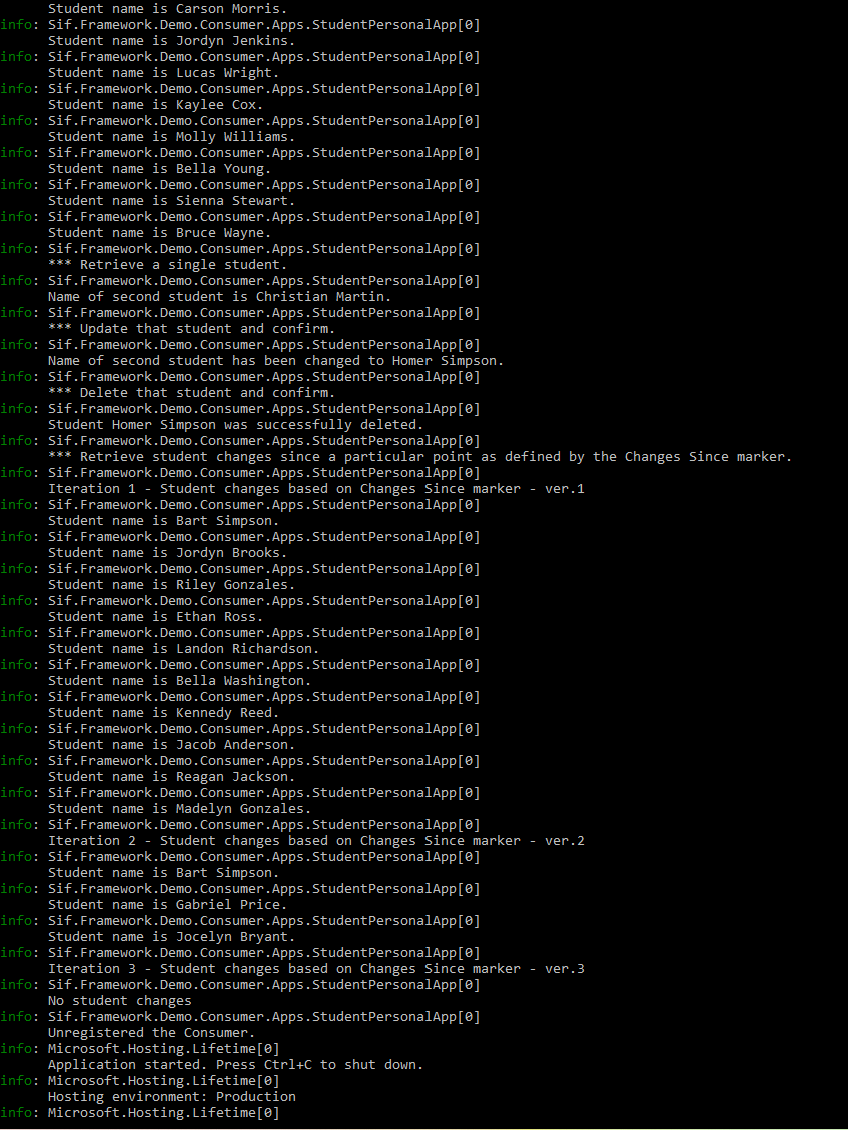


Figure 4: Demo student Consumer runtime output

# Review expected behaviour

The scenario illustrated in this set up is one where a Consumer is making a call for student data in a Direct Environment. As such, once this data has been consumed, the demo has been completed.

In this case, the student Consumer simply prints student details to the console based upon the Create, Retrieve, Update and Delete (CRUD) operations executed. Once the details have been printed, the Environment Provider and student Provider instances can be stopped.

The architecture of this Direct Environment is outlined below.

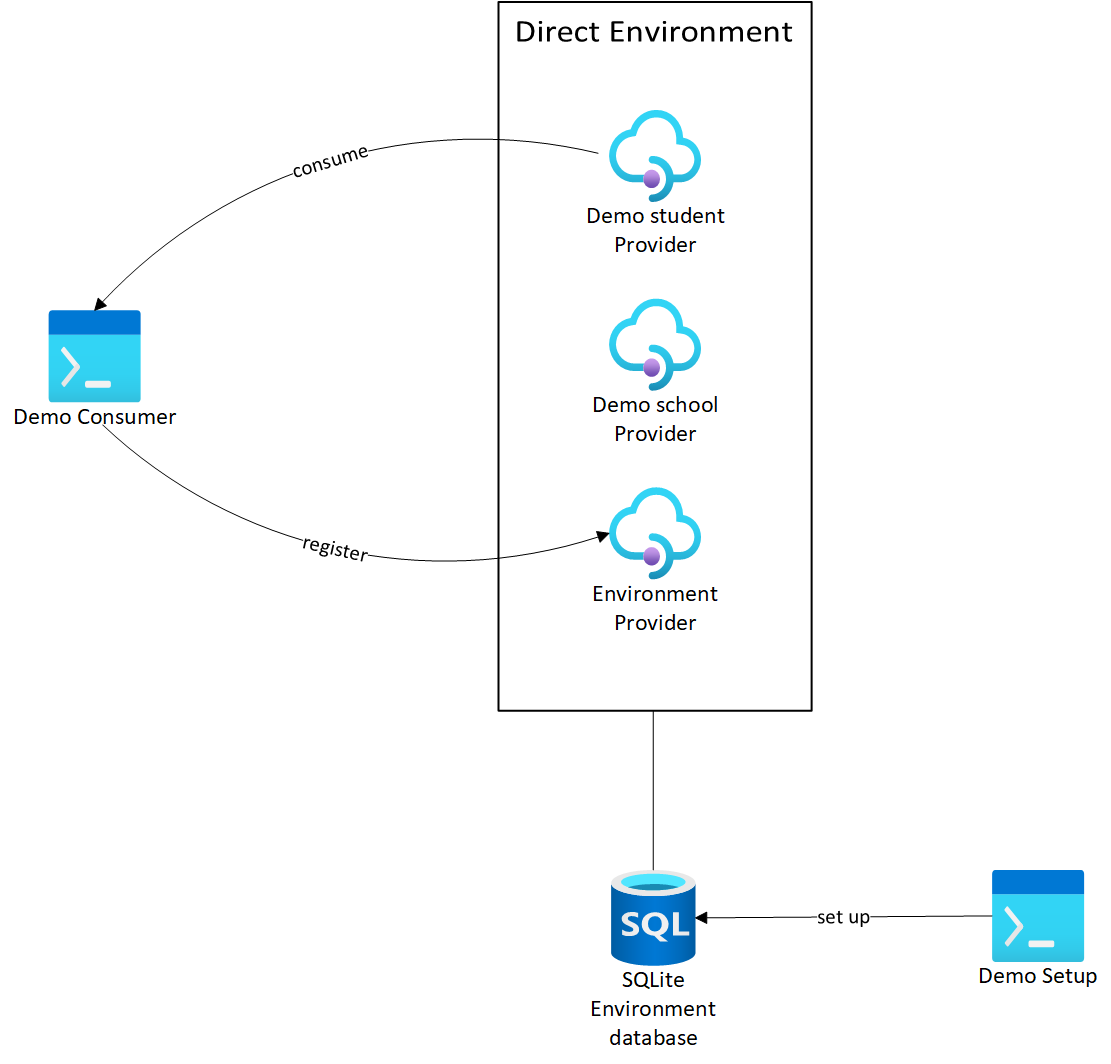


Figure 5: Direct Environment architecture