RMIT Counter Open Source Sample Project

Contents

[Getting Started 2](#_Toc407114413)

[Download and Run 2](#_Toc407114414)

[View CSV, TSV and HTML Counter Reports 2](#_Toc407114415)

[Retrieve Sushi reports with sushi client 3](#_Toc407114416)

[Project Structure 4](#_Toc407114417)

[Applications Folder 4](#_Toc407114418)

[Data folder 4](#_Toc407114419)

[Libraries folder 4](#_Toc407114420)

[Tests folder 4](#_Toc407114421)

[Add custom authentication and Authorization 4](#_Toc407114422)

[Add your Data to produce your Counter reports 4](#_Toc407114423)

# Getting Started

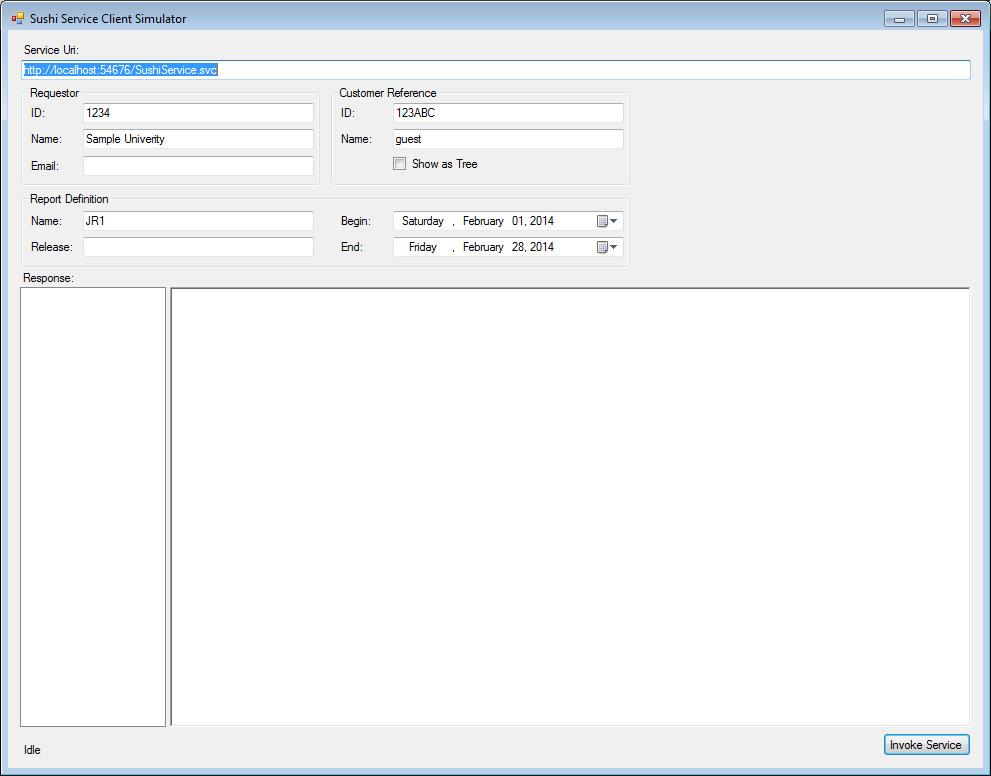
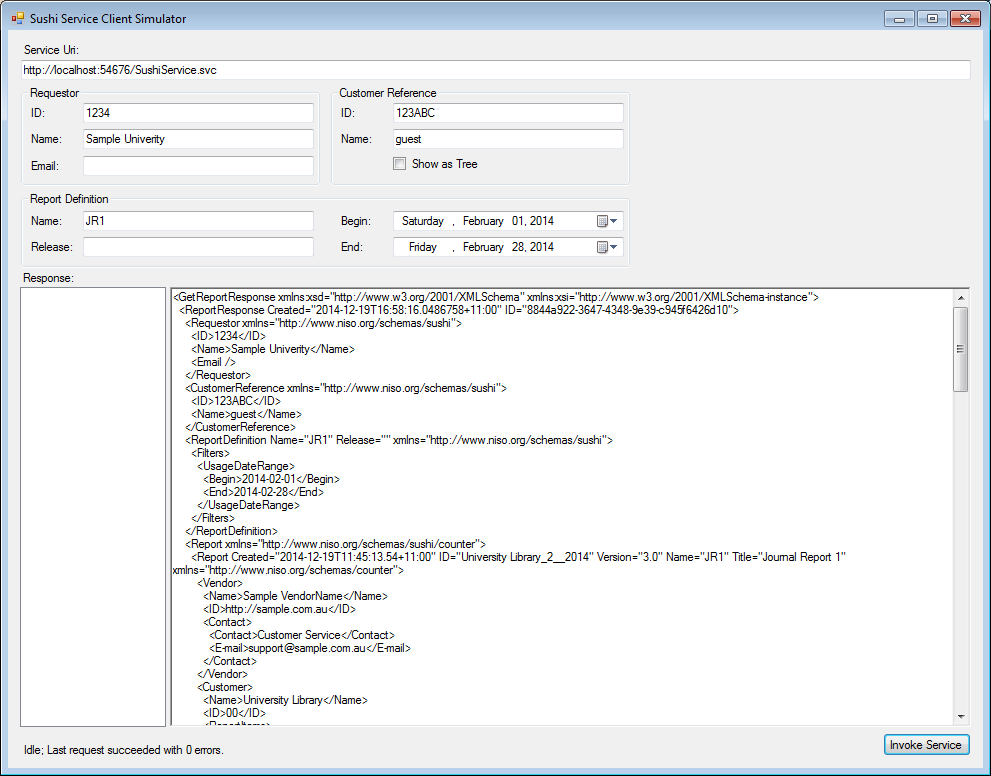
## Download and Run

1. Download project from github to “C:\Temp\”
   1. If you choose other location then “C:\Temp\” to download and save project then web.config file of **CounterReports** and **SushiService** project must be change accordingly <add key="Sushi.CounterXsltFolder" value="C:\Temp\RMIT.Counter\Libraries\Reporting\Reports\Xslt" />
2. Open solution in visual studio and compile it.
3. Make sure it compiles without any errors.

## View CSV, TSV and HTML Counter Reports

1. Click and run **CounterReports** web project (Ctrl + F5)
2. Select Report\_Format to HTML
3. Select Report\_Template to JR1
4. Click **Generate Report**
5. It will display the report as HTML.
   1. Please note as this is sample data, we are serving same report data always.
   2. User Authentication is authorized if ID is **123ABC** and customer name is **guest**.
   3. We encourage you to add your data and authentication logic.
   4. Therefore changing date range or username will not produce different report.
6. Change the Report\_Format to CSV or TSV and also change the Report\_Template to other report type and try to submit the form again and wait until download dialog pops up.
   1. You can add additional formats by adding new XSLT files

## Retrieve Sushi reports with sushi client

1. Run **Sushi Service** project (Ctrl + F5)
2. Navigate to <http://localhost:54676/SushiService.svc> and make sure service is running
3. Now run **sushi client** (Ctrl + F5). It would open following windows form
4. Click on **invoke service** button and it would display sushi report for JR1
5. Change the report definition name from JR1 to JR5 or MR1 or BR1 or BR2 and invoke service again.
6. Report Response window will display appropriate result.

# Project Structure

## Applications Folder

Applications folder Contains following applications

* **CounterReports**: Asp.net 4.5 WebApi/MVC web application. It presents counter reports in html, csv and tsv format.
* **Sushi Client** : Windows Form application that calls **Sushi Service (WCF Service)** and displays counter reports in **sushi xml** format
* **Sushi Service:** it is web applications that host the WCF Service.

## Data folder

* Contains Sample xml data to be used to produce counter reports

## Libraries folder

* **Library :**  class library contains commonly used classes
* **Reporting :** class library contains implementation for counter reports generation
* **Sushi :** class library that contains Sushi Authentication and Sushi Service partial implementation
* **Sushi Core:** class library that contains Sushi Service complete implementation with WSDL

## Tests folder

We use NUnit for testing framework and NSubstitute for Mocking framework.

* Unit: contains testing code for Unit tests to test smallest possible source code
* Integration: contains Integration tests to test more than Units

# Add custom authentication and Authorization

**AuthorizationAuthority.cs** class inside the project **Sushi,** under **Libraries** folder is responsible for providing authorization. Navigate to this class and add Custom Authorization. Currently, its sample data with username: “guest” and Id: “123ABC”.

# Add your Data to produce your Counter reports

This project uses Sample data from **Data** folder to produce Counter and Sushi reports. However, it can be change by updating **ReportDataBase.cs** class GetDataset() method. **ReportDatabase.cs** class is in **Reporting** class library project under **Libraries** folder. Navigate to the GetDataset() Method and start adding different SQL Query/StoredProcedure for different reports (Jr1,Jr5 etc). SQL Query or StoredProcedure added make sure it return appropriate dataset. Dataset’s data must conform to XML Schema file (.xsd) in **Data** folder.

For e.g. jr1 sql query must return dataset same as JR1.xsd in **Data** folder.