# TFS Evolution solution

The solution contains of two projects:

* TFSDataAccessPortable
* TFSEvolution

1. **TFSDataAccessPortable**

The project is a portable class library that can also be used by other applications and is responsible for getting the TFS data.

The important class for getting the data is TFSClient.cs. This class contains all calls to the VSO Rest Api and our own restful web service.

In the project folder “DataContracts” all contracts for mapping with the JSON response can be found. The folder “DataModel” contains all classes used by the app that are returned by the TFSClient.cs.

Some useful extension methods for working with HttpClient and internal classes can be found in the “Extensions.cs” in the “Misc” folder.

1. **TFSEvolution**

This project is the Windows 8.1 App. The app is implemented with some principles of the MVVM pattern (but without using any nugget package). In the folder “Common” you can find some of the classes for MVVM.

All classes needed for authentication are available in the “Auth” folder. As the app uses different types of authentication the app has an “OAuthAuthenticator” used for authentication with OAuth and a “BasicAuthenticator” used for authentication with the alternative credentials.

The Views and Viewmodels can be found in the corresponding folder as well as converters. The special user controls are stored in the “Controls” folder.

The “Data” folder contains the Context.cs class, used to identify the custom web service. Additionally it contains the “DataSource.cs” used for loading the data with the portable class library and for storing the runtime data for the app.