Software Developer

Technical Exercise

# Introduction

You have been tasked with completing a partially finished Address Book web app.

The web app should read data from a file stored on disk and expose a public API which users can use to search for a contact by name. Details on the contact should then be returned.

You are provided with:

* A sample data file
* The file specification
* A half-finished project

You will need:

* Visual Studio 2015 Community  
  Download free from <https://www.visualstudio.com/downloads/>
* The NUnit 3 Test Adapter for Visual Studio (so you can run the Unit Tests in Test Explorer)  
  See ‘Tools’ > ‘Extensions and Updates’ in Visual Studio
* A Hex editor (optional, but useful)  
  HxD is free from <http://mh-nexus.de/en/hxd/>

# Your taskS

Your tasks are:

1. **Complete the AddressBookReader class so that all unit tests (already defined) pass cleanly.**
2. **Complete the Web project so a HTTP-based API is exposed and documentation on how to invoke this API is available from the browser.**

Note:

* There is no time limit for this task - we are more interested in the approach you take and the quality of your code than the time it takes.
* Looking up things you are not sure about on Google, MSDN or StackOverflow is encouraged
* Generating documentation for your API should be largely automated
* Don’t over-engineer your solution – simpler is better.

## Bonus Points (optional)

For bonus points, add a user interface to the project using technologies of your choice.

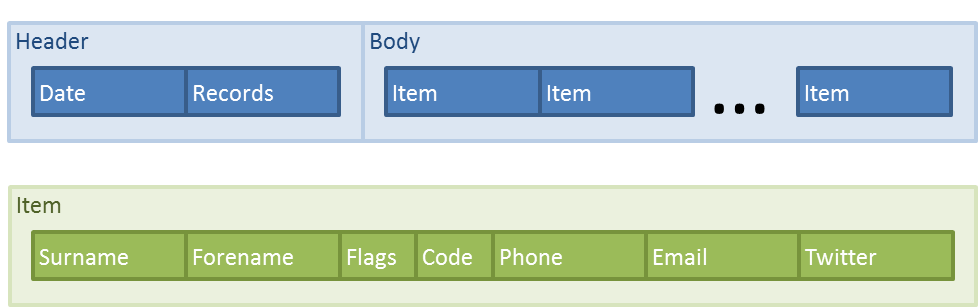
# Existing Code

The following projects and files are already present:

* **AddressBook.Core** – contains the object model and unfinished AddressBookReader class
* **AddressBook.Tests** – contains the NUnit 3 Tests
* **AddressBook.Web** – a bare-bones web project containing just enough code to use your AddressBookReader to load in an AddressBook (see Global.asax and Index.cshtml)
* **data.bin** – the sample file used for the tests and the web app
* **Readme.docx** – this file

# ADDRESS BOOK File Specification

The Address Book File Format is a simple binary data based format for storing address book items. It consists of two main sections:



* The Header section contains information on the file, such as the number of items it contains and the date it was generated
* The Body contains the Items to be read

## Header

The file header starts at an offset of 0x0F and contains the following fields:

|  |  |
| --- | --- |
| Offset | Field |
| 0x00 | Export Date (days since) int, 4 bytes |
| 0x04 | Number of Items in the file int, 4 bytes |

## Body

The Body of the file contains an array of Contact Items. Each Contact Item is 146 bytes long and contains the following fields (Strings are ASCII encoded):

|  |  |
| --- | --- |
| Offset | Field |
| +0x00 | Surname int, 16 bytes |
| +0x10 | Forename int, 16 bytes |
| +0x20 | Item Flags flags, 1 byte   |  |  | | --- | --- | | 0 | Favourite? | | 1 | 0 = Male, 1 =Female | | 2 | 0 = Personal, 1 = Work | | 3 | Recent? | | 4 | Not Used | | 5 | | 6 | | 7 | |
| +0x21 | Phone: Country Code int, 1 byte |
| +0x22 | Phone: Number string, 16 bytes |
| +0x32 | Email Address string, 64 bytes |
| +0x72 | Twitter Username string, 32 bytes |