

Technical Assessment

Create a C# application that reads in book information, looks up additional metadata, and outputs details to a flat file. You can download a free version of Visual Studio from here: <https://www.visualstudio.com/vs/community/>. After you are done, ZIP up and return the whole solution folder.

This application should take in the following command line arguments:

- -Input="<Path to Books.xml>"
- -Output="<Path to Output.txt>"
- -E (indicates whether or not to log errors to file)

For example, the application may be executed as shown below, although the parameters could be in any order:

```
Test.exe -Input="C:\Temp\Books.xml" -Output="C:\Temp\Output.txt" -E
```

The application should parse each of the ISBN's out of Books.xml and call a REST API hosted by openlibrary. A sample call for a book with ISBN 123456789 looks like: <http://openlibrary.org/api/books?bibkeys=ISBN:123456789&jscmd=data&format=json>. If the REST call returns no results for that ISBN, add an attribute to the appropriate <book> node called "found" with a value of "false". If results are found, add a "found" attribute with a value of "true", as well as create the following nodes within the appropriate <book> node:

- <publishDate> node populated by "publish_date" from the REST call
- <subjects> node with one child <subject> node per "name" in the "subjects" results from the REST call.

Save this modified XML to the same folder that holds the Output.txt file specified in the -Output command line argument. Additionally, create a flat file at the location specified in the -Output command line argument that is tab-delimited with the following information for each book:

- ISBN
- Title
- Author
- Price
- Pages

- Publish Date
- The first Subject if there are multiple subjects
- Path to the updated Books.xml file

Finally, if the -E command line argument was passed to the application, log any errors to a file called Error.log. This log file should be written to the same directory your EXE is run from.