The inBloom Accelerator code sample for Word and Excel shows how to integrate inBloom’s educational data service into useable apps for educators. It provides an extendable pattern for consuming student data from inBloom and using it in Apps for Office.

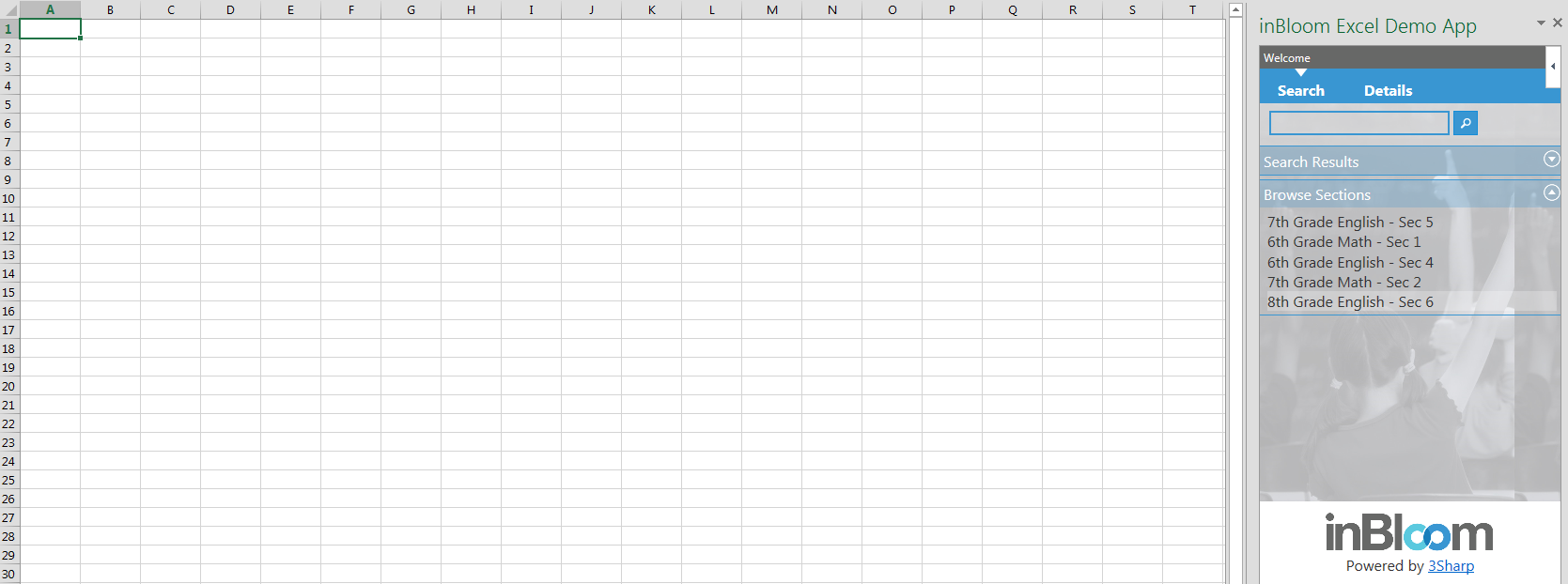
Description of the *inBloom Accelerator sample for Word and Excel*

The inBloom Accelerator apps showcase authenticating to inBloom’s API through OAuth 2.0 utilizing the Microsoft ASP extension for DotNetOpenAuth. The apps also show how to acquire inBloom data entities and integrate them to form a new data structure that is easier to work with. It does this asynchronously and shows how to assemble the entities together in JavaScript where there would otherwise be no guarantee to the order of data returned.

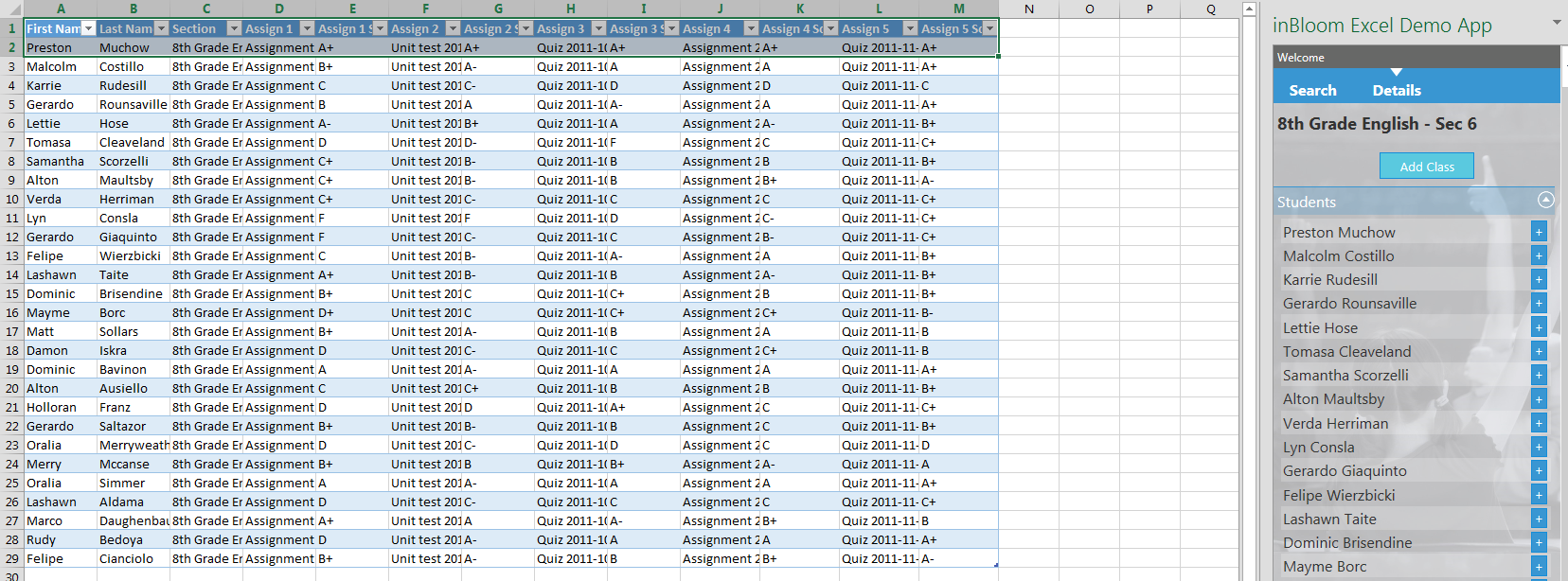
Excel App

The Excel app demonstrates the use case of a teacher browsing or searching for his or her students and inserting assignment grade information into a table. The app focuses on interacting with Excel tables and provides a basic framework that can be reused to quickly facilitate Excel interaction.

**View of Excel App before selecting a student or class**



**View of Excel App after adding a class to the table**



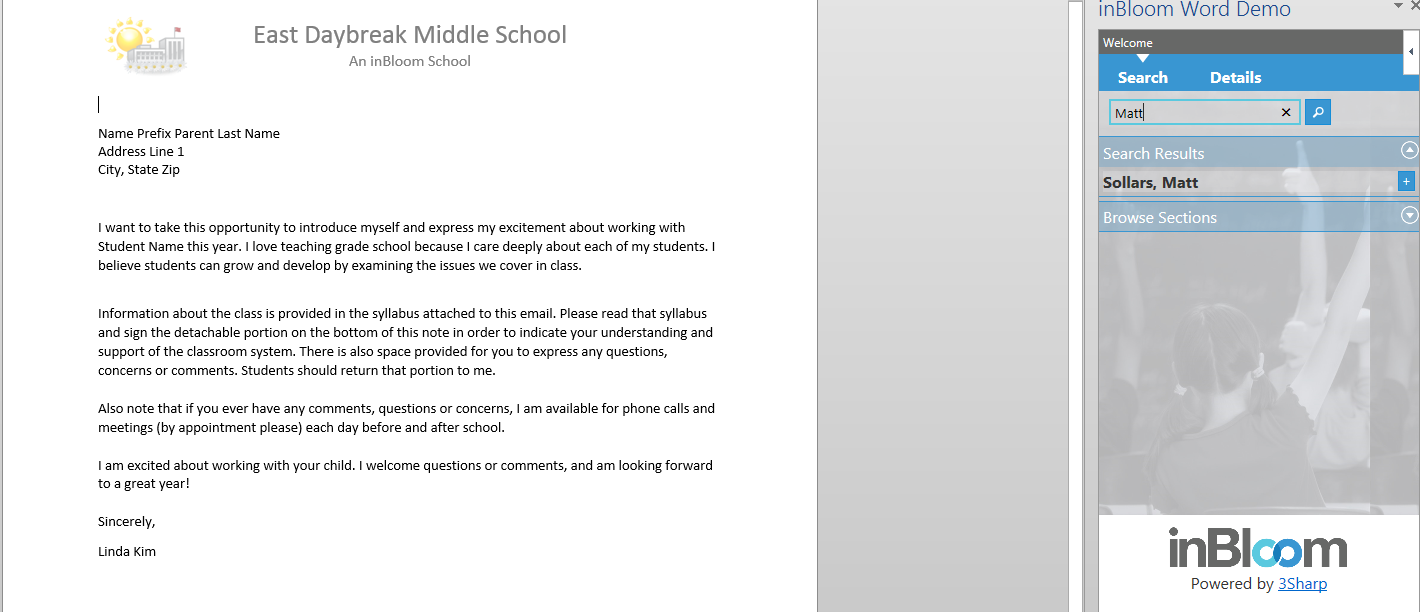
Word App

The Word app demonstrates the use case of a teacher browsing or searching for his or her students and binding the template’s content controls to student information for personalization of a letter.

This is done by adding a custom xml part to the template and binding the content controls on the document surface to the custom xml. The app updates the custom xml, and the changes are reflected on the document surface.

To add additional content controls, use Word’s Developer tab -> XML Mapping and pick the “Student Info” custom xml part.

**View of Word App before adding student information**



**View of Word App after inserting student information**



Prerequisites

This code sample was built using Windows 7 and IIS 7.5 and was also tested on Windows Server 2008 using IIS 7.5.

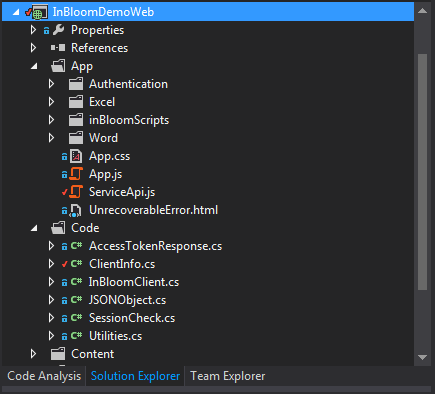
For the best experience, we recommend temporarily installing the self-signed certificate to the root trust store or using a trusted certificate (See [Troubleshooting](#Troubleshooting) and [Related content](#RelatedContent) for details). Other prerequisites are as follows:

* Office 2013 (Word and Excel)
* Visual Studio 2012
* Office Developer Tools for Visual Studio 2012:
  + <http://msdn.microsoft.com/en-us/office/apps/fp123627.aspx>
* IIS with SSL configured
  + HTTPS is required by inBloom and under the OAuth spec
  + Ensure you have registered ASP.NET with IIS if you did not set up IIS first
  + See [Troubleshooting](#Troubleshooting) and [Related content](#RelatedContent)for details
* .NET 4.5:
  + <http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx>
* inBloom Developer account:
  + <https://www.inbloom.org/developer-account-registration>
* Internet connection
* Familiarity with the following
  + JavaScript
  + C#
  + WCF RESTful web services
  + OAuth 2
  + inBloom API

Key components of the sample

The key files are located in the web project under the App, Code, and Service folders.

**Important folders and files**

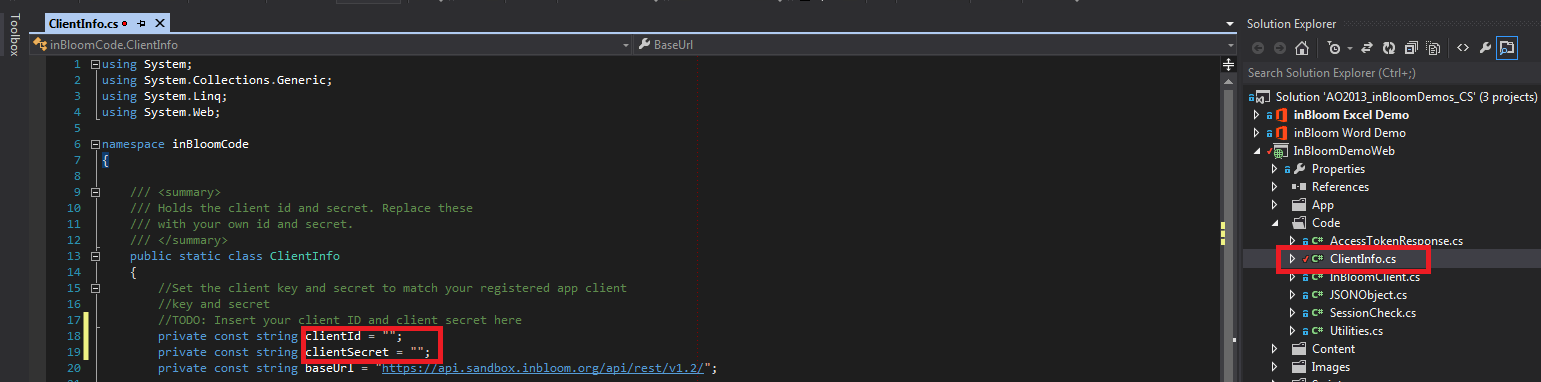


* The App folder contains the folders for the Excel and Word apps, inBloomScripts folder, and the ServiceApi JavaScript file.
  + The page for each app is located in the Excel and Word folders, respectively, along with the app JavaScript file and other JavaScript files that manage data for searching/browsing and drawing the related UI. These files highlight the pattern that can be used to manage the various inBloom entities that will be retrieved.
  + The inBloomScripts folder highlights the pattern for bringing the entities into a more usable fashion by creating default values, checking for undefined properties in one location, and providing functions for each entity including retrieval.
  + ServiceApi.js contains public, static-equivalent functions that call the included web service in order to retrieve data from inBloom.
* The Code folder contains C# classes that facilitate authenticating with inBloom and requests to the inBloom API.
* The Service folder contains the web service that will make the actual calls to inBloom’s API. This is needed because inBloom does not support JSONP or CORS.

Configure the sample

You will need to register the code sample application with inBloom to obtain a client ID and key, and insert them into ClientInfo.cs to enable authentication. The registration should use the inBloom “Small Sample Dataset” as it was regarded by inBloom as the most realistic in terms of data (as of May 2013).

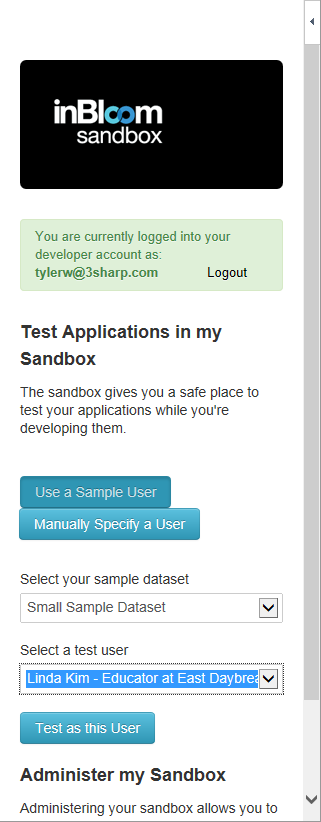
**Code file and lines to insert client ID and client secret**



Once the ID and key are added, you will need to create a virtual directory for IIS from the web project’s web properties (right-click the web project and choose properties, then click the “Web” tab, then the “Virtual Directory” button), and then build the solution.

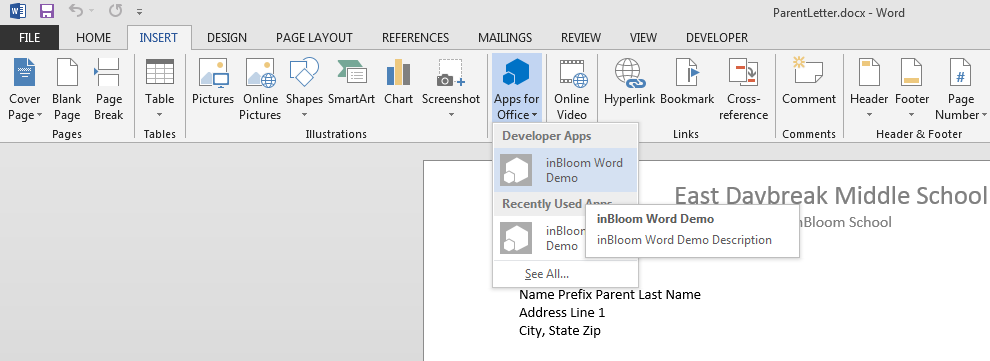
**Excel**: Set the Excel Demo project as the startup project and press F5. You will then need to log into inBloom and select a user to impersonate. Linda Kim is the suggested user to impersonate as the data for her in the “Small Sample Dataset” is the most complete (as of May 2013).

**Select a user to impersonate within inBloom’s authentication**



**Word**: Set the Word Demo project as the startup project and press F5. Ignore (but do not close) the blank Word document that opens. In Visual Studio, under the Word Demo project, open ParentLetter.docx. Navigate to the “Insert” tab of the ribbon, click the “Insert Apps for Office” button, and insert the Word app. From here, log in to inBloom and choose a user to impersonate. Linda Kim is the suggested user to impersonate as the data for her in the “Small Sample Dataset” is the most complete (as of May 2013). We also recommend using the student “Matt Sollars” as he is one of the only students with any parent information (though the parent is missing gender information and an address).

**Insert the app into ParentLetter.docx**



Run and test the sample

Once the app is running in debug mode, breakpoints can be set and the code can be stepped through normally. All debugging must be done inside Visual Studio, as Apps for Office does not have a built in debugger (as of May 2013).

Troubleshooting

The following describe a few issues that have been observed along with recommendations for overcoming the issues:

* Parent name inserted with Word app displays with a ‘-‘ before it.
  + This is expected behavior when the dataset contains a null value for gender (no Mr. or Mrs., etc.). Because a gender indicator is expected, Word cannot omit an insertion here, but the character can be configured to something else.
* inBloom gives a server error upon starting the app (before logging in).
  + Make sure the callback for the URL matches the URL path to authentication.aspx.
* The data inserted into Excel looks incorrect or appears to be missing.
  + inBloom’s “Small Sample Dataset” has significant data gaps as of May 2013. It is continually being improved, however.
* The app was working, but idling errors appear after some time when trying to get data.
  + The OAuth token from inBloom expires 60 minutes from the last API call. Simply reload the app to get a new token.
* Clearing the data from an Excel row and then inserting a new student leaves gaps in the table.
  + Delete the unwanted rows (not just the data in the row) by right-clicking on the row number in Excel and choosing Delete.
* The Excel app seems very slow when getting data.
  + Data retrieval can be slow since it’s done asynchronously (and in phases in order to make the code easier to follow). There can be hundreds of calls outgoing due to the normalization of inBloom’s API. To improve performance, we highly recommend implementing caching to cut down on calls. We did not implement caching in this the code sample in favor of simplicity.
* The web application project fails to load with a “You do not have sufficient privilege to access IIS web sites on your machine” error.
  + Ensure Visual Studio is running as Administrator.
* The app webpages do not load or a MIME error is thrown by IIS.
  + You may need to register ASP.NET with IIS.
    - See [Related content](#RelatedContent): ASP.NET IIS Registration.
* The app fails to load initially and on a reload presents a certificate warning page.
  + For the best experience, either use a trusted certificate or install your certificate to the trust store. The quickest way to do this is as follows:
    - Open Internet Explorer with Administrator privileges
    - Navigate to any part of the website
      * Default is <https://localhost/inBloomWebDemo>) and continue to the site
    - Click the red, broken security shield to the right of the URL
    - Click “View Certificate”
    - Click “Install Certificate”
    - Follow the wizard to the “Certificate Store” step and ensure “Place all certificates in the following store” is selected
    - Choose “Browse” and “Trusted Root Certification Authorities”
    - Finish the wizard by choosing to install the certificate when prompted
* “Web page expired” error.
  + Ensure Internet Explorer Options -> Advanced -> Disable script debugging (Internet Explorer) and Disable script debugging (Other) **are not** checked.
* inBloom error, 403 unauthorized, app pages will not load.
  + When registering the app with inBloom, the redirect URL **must** match the URL to the authentication page.
  + Ex. <https://localhost/inBloomWebDemo/App/Authentication/Authenticate.aspx>

Related content

The following links provide helpful information related to app setup and working with inBloom:

* IIS Install – Windows 7:
  + <http://technet.microsoft.com/en-us/library/cc725762.aspx>
* ASP.NET IIS Registration:
  + <http://msdn.microsoft.com/en-us/library/k6h9cz8h%28v=vs.100%29.aspx>
* Setting up IIS for HTTPS:
  + <http://blogs.msdn.com/b/rakkimk/archive/2007/05/25/iis-7-how-to-configure-a-website-for-https.aspx>
* Installing certificate to root store:
  + <http://www.poweradmin.com/help/sslhints/ie.aspx>
* inBloom developer information:
  + <https://www.inbloom.org/for-developers/sandbox>
* inBloom developer account registration:
  + <https://www.inbloom.org/developer-account-registration>
* inBloom API documentation:
  + <https://www.inbloom.org/developer-documentation>
* Registering an application with inBloom:
  + <https://www.inbloom.org/sites/default/files/docs-developer-1.0.68-20130118/doc-909557d4-cf76-4144-8201-3f643664fb25.html>