Microsoft Office Add-ins

Community call

- AppSource Add-in Validation Improvements
- Blazor in Office Add-ins
- Add-in command changes
- Community ideas

May 11, 2022

Agenda

- **AppSource: Add-in validation Improvements**
- **Blazor in Office Add-ins**
- **Add-in command changes**
- **Community ideas**

Abid Rahman
Product Manager

Linda Cannon
Senior Content Project Manager

Maarten van Stam
MVP Office Add-ins

Juan Balmori
Principal Program Manager

AppSource: Add-in validation improvements

Juan Balmori: Principal Product Manager

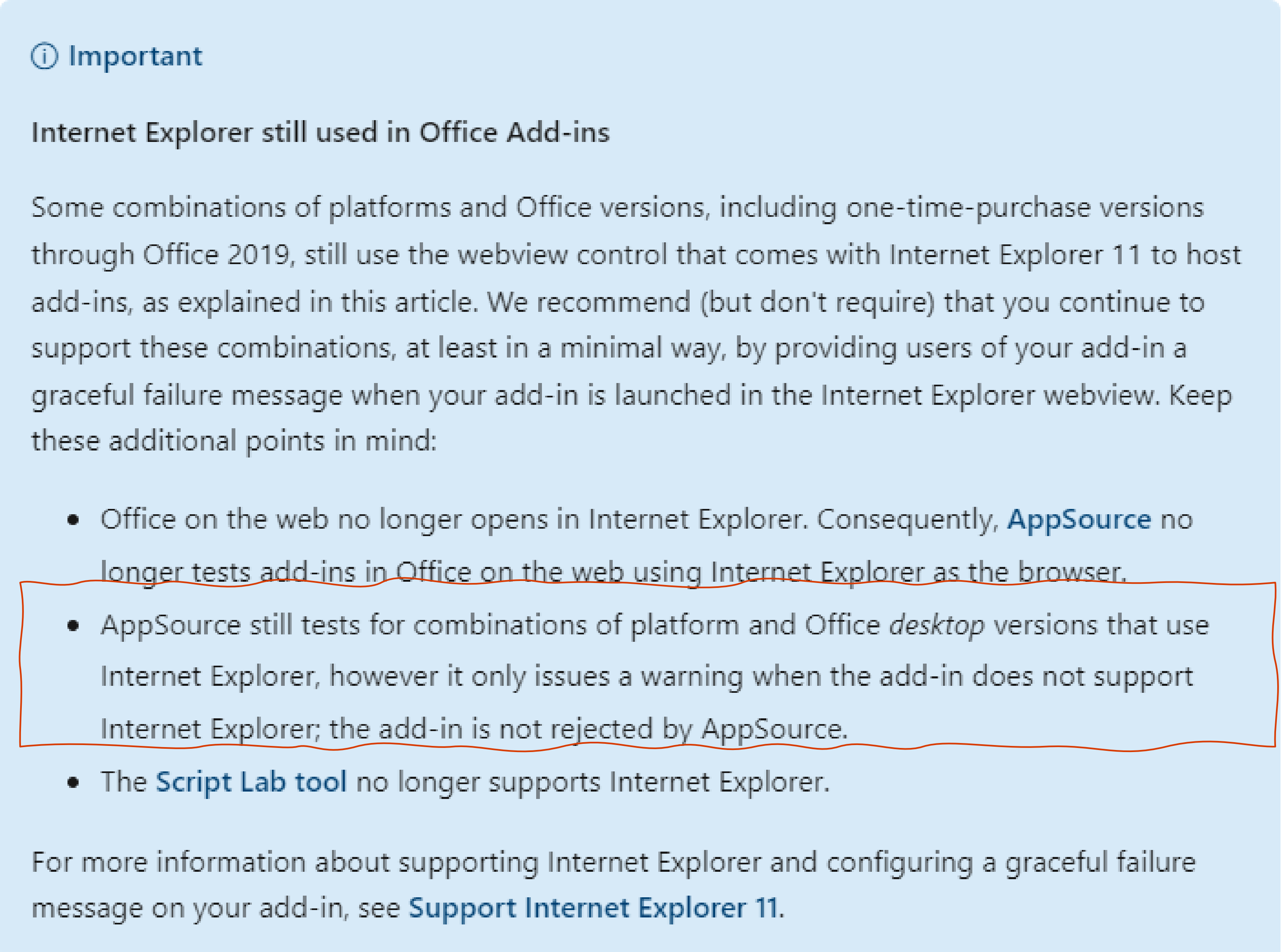
Agenda

- **Browser Support Update**
  - In a nutshell: we will no longer reject add-ins not supporting Trident!
- **Top Validation Failures**
  - Share our top 10 failures so you can help yourself publishing your Add-in faster!

Context

- **Many months ago, we announced IE will not longer be supported (for Office Web Apps)**
- **An Exception to the Above was Web Add-ins running on Win32, we still can activate your add-in in IE (Trident) if you are in:**
  - Office 2013 msi -> Extended support ending on April 11, 2023
  - Office 2016, 2019 msi -> Extended support ending on October 14th, 2025
  - Windows 7, 8, and 10 < 1903
- **Folks in M365 subscriptions (monthly or semi-annual) are with high probability running in WebView2.**
- **Good news !**
  - Telemetry indicates WebView2 has become mainstream with a very significant % of Office Add-ins activations.
  - Hence, we are now “recommending” to support Trident, but its not required to be in the store.
  - If your add-in fails to load in Trident we will Soft-fail it.
  - Effective immediately.

Documentation Updated!



ⓘ Important

Internet Explorer still used in Office Add-ins

Some combinations of platforms and Office versions, including one-time-purchase versions through Office 2019, still use the webview control that comes with Internet Explorer 11 to host add-ins, as explained in this article. We recommend (but don't require) that you continue to support these combinations, at least in a minimal way, by providing users of your add-in a graceful failure message when your add-in is launched in the Internet Explorer webview. Keep these additional points in mind:

- Office on the web no longer opens in Internet Explorer. Consequently, AppSource no longer tests add-ins in Office on the web using Internet Explorer as the browser.
- AppSource still tests for combinations of platform and Office *desktop* versions that use Internet Explorer, however it only issues a warning when the add-in does not support Internet Explorer; the add-in is not rejected by AppSource.
- The Script Lab tool no longer supports Internet Explorer.

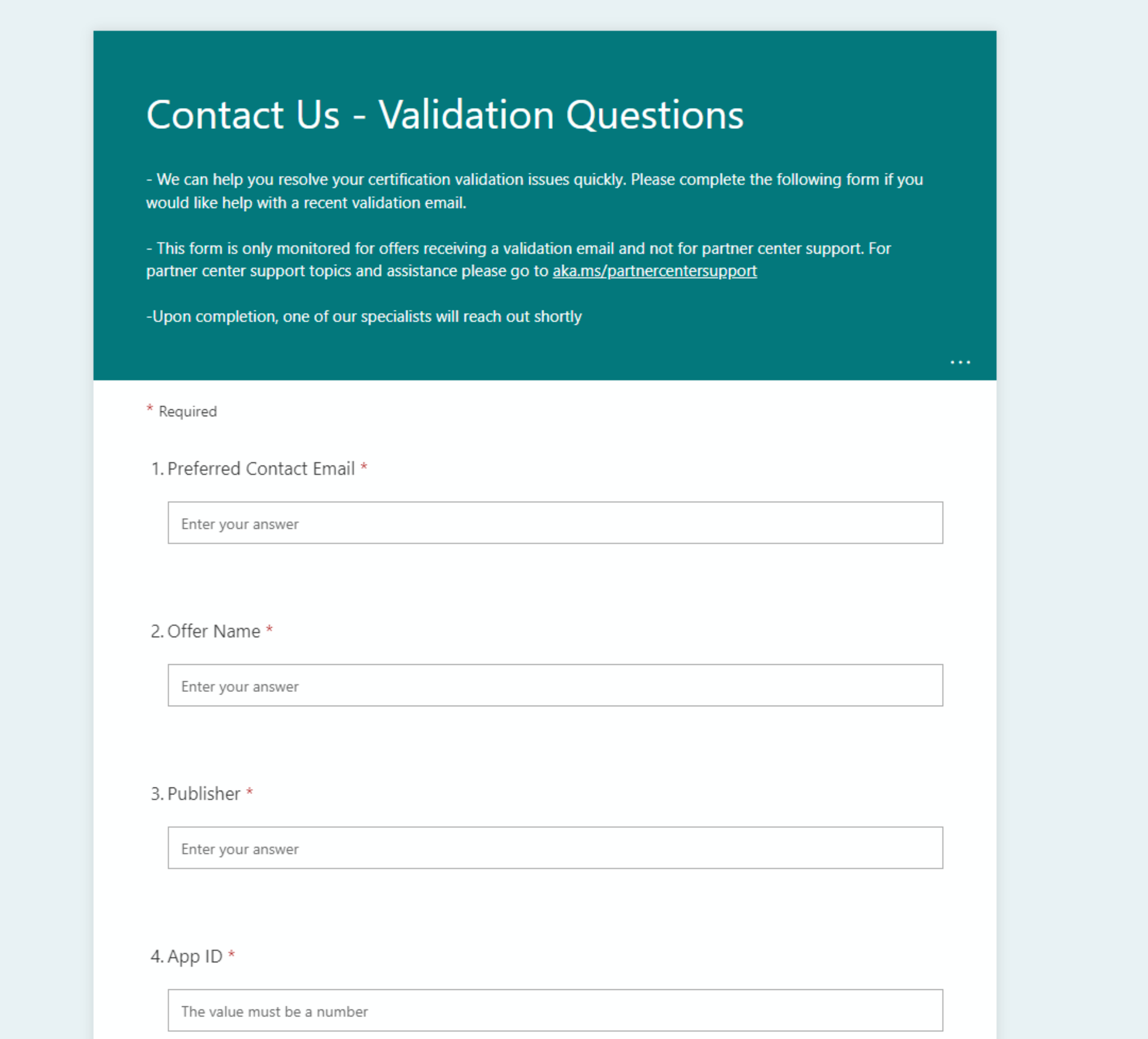
For more information about supporting Internet Explorer and configuring a graceful failure message on your add-in, see Support Internet Explorer 11.

Top Failures when validating Add-ins

1. Testing notes
2. First run experience.
3. External Service experience: Sign In-out, Sign up links
4. Additional charges disclosure
5. Provider Name Match
6. Functionality as described.
7. High Resolution Icon.
8. Environment compatibility

Directly Engage with Validation Team!

- **PAST: Add-in gets rejected -> Challenges to validation were made via CSS support tickets**
- **Announcing: direct interaction with Validation Team.**
  - Find the link on your certification report e-mail to contact our team directly. (via Forms)
  - This will reduce enormously our response times.



Contact Us - Validation Questions

- We can help you resolve your certification validation issues quickly. Please complete the following form if you would like help with a recent validation email.

- This form is only monitored for offers receiving a validation email and not for partner center support. For partner center support topics and assistance please go to aka.ms/partnercentersupport

-Upon completion, one of our specialists will reach out shortly

* Required

1. Preferred Contact Email *

Enter your answer

2. Offer Name *

Enter your answer

3. Publisher *

Enter your answer

4. App ID *

The value must be a number

- # Blazor in Office Add-ins

Maarten van Stam: MVP Office Add-ins

Maarten’s slides here

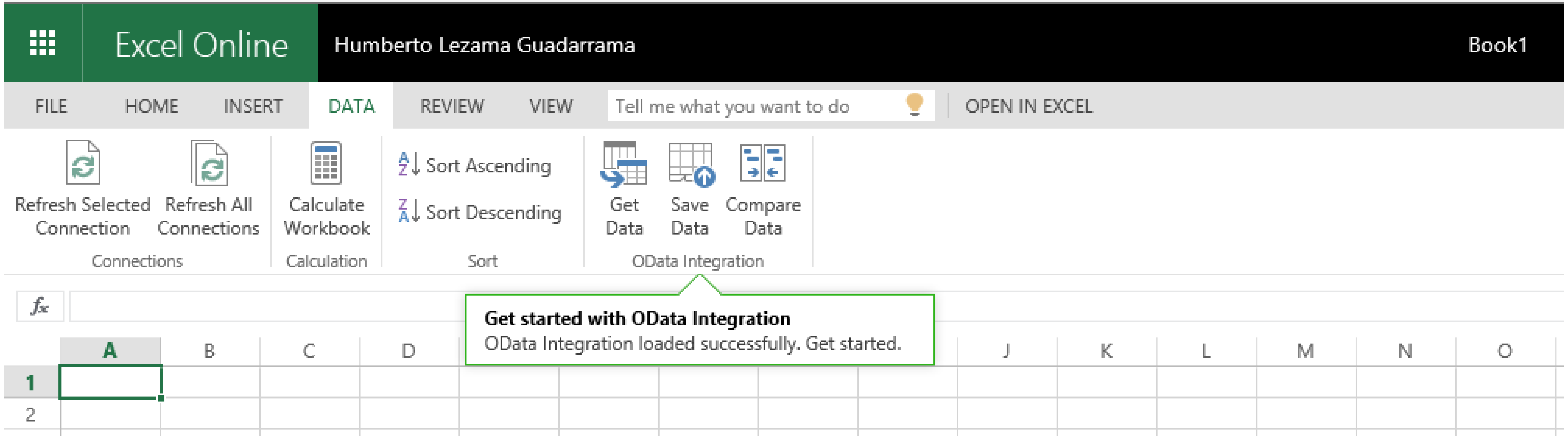
- # Add-in command changes

Abid Rahman: Product Manager

What are add-in commands?

Add-in commands extend native parts of Office applications and can perform one of two actions:

1. ShowTaskpane
2. ExecuteFunction



Get started with OData Integration
OData Integration loaded successfully. Get started.

ExecuteFunction add-in commands are defined in the manifest as:

```
<Action xsi:type="ExecuteFunction">
                <FunctionName>writeText</FunctionName>
</Action>
```

To improve security in Office Add-ins, we are changing how you create ExecuteFunction add-in commands. This change does not require re-submission of your Office Add-in.

In addition to defining the JavaScript function, you also need to register the function with Office by adding a call of `Office.actions.associate("name-of-function", function)`.

```
Office.actions.associate("name-of-function", function)
```

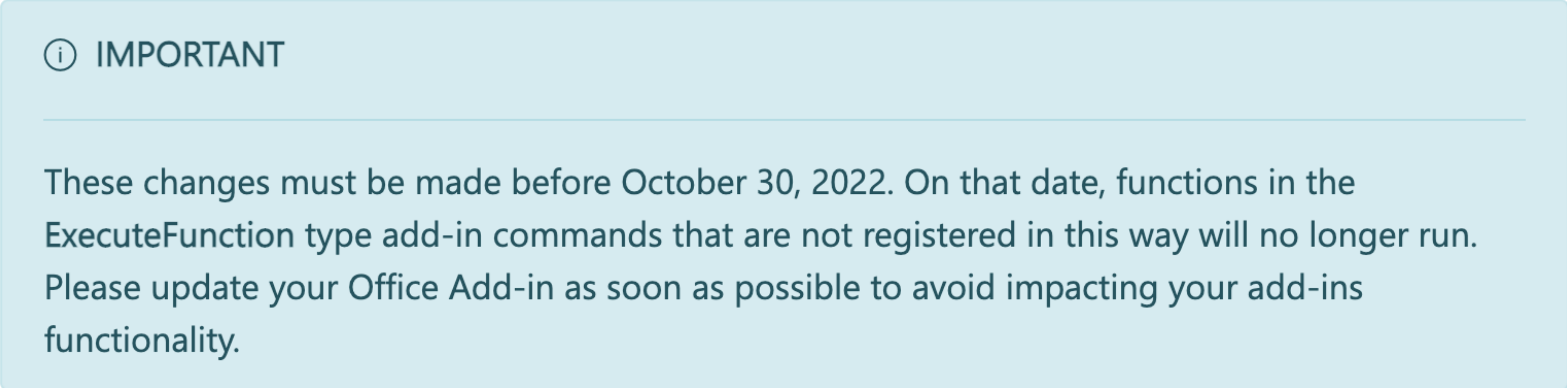
```
// Define the function.
function writeText(event) {

    // Implement your custom code here. The following code is a simple example.
    Office.context.document.setSelectedDataAsync("ExecuteFunction works. Button ID=" + event.source.id,
        function (asyncResult) {
            var error = asyncResult.error;
            if (asyncResult.status === Office.AsyncResultStatus.Failed) {
                // Show error message.
            }
            else {
                // Show success message.
            }
        });

    // Calling event.completed is required. event.completed lets the platform know that processing has completed.
    event.completed();
}

// You must register the function with the following line.
Office.actions.associate("writeText", writeText);
```

Announcing changes to Office Add-ins' ExecuteFunction commands - Microsoft 365 Developer Blog

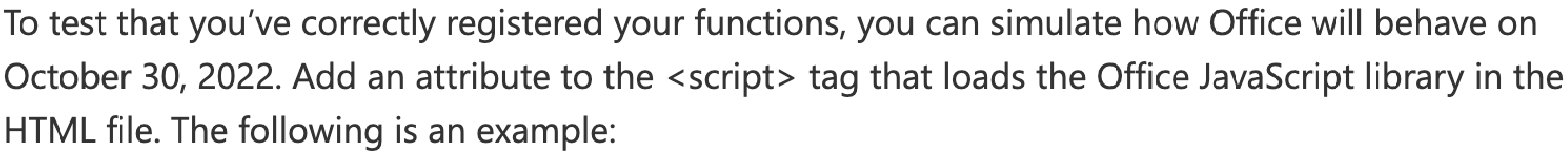


ⓘ IMPORTANT

These changes must be made before October 30, 2022. On that date, functions in the ExecuteFunction type add-in commands that are not registered in this way will no longer run. Please update your Office Add-in as soon as possible to avoid impacting your add-ins functionality.

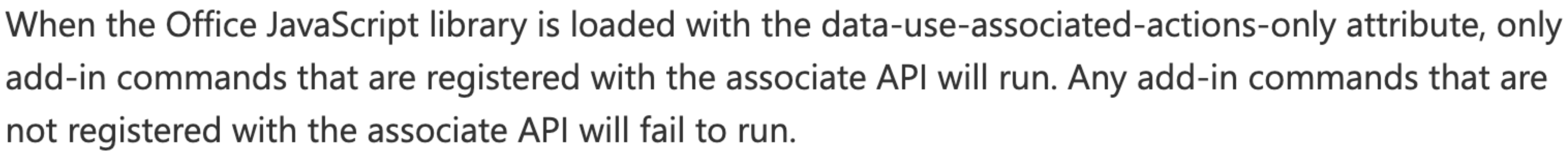
You can find complete add-in samples in our GitHub repo. To learn more about add-in commands, see our documentation.

For questions, we invite you to join our monthly Office Add-ins community call. Lastly, if you run across any issues, please report them in the Github repo.



To test that you've correctly registered your functions, you can simulate how Office will behave on October 30, 2022. Add an attribute to the <script> tag that loads the Office JavaScript library in the HTML file. The following is an example:

```
<script src="https://appsforoffice.microsoft.com/lib/1/hosted/office.js" data-use-associated-actions-only="1" type="text/javascript"></script>
```



When the Office JavaScript library is loaded with the data-use-associated-actions-only attribute, only add-in commands that are registered with the associate API will run. Any add-in commands that are not registered with the associate API will fail to run.

Community ideas

Linda Cannon: Senior Content Product Manager

Help shape the future of Microsoft 365 developer suggestions/ideas

- Recap: current situation
- Options
  - Stay put at https://aka.ms/m365dev-suggestions
  - Open new issue at https://github.com/OfficeDev/office-js
  - Move to new all-up Microsoft Ideas forum at Feedback · Community (microsoft.com)

Samples discussion

We’re looking for your feedback and suggestions for what you need from samples for Office Add-ins.
Please join us June 15th at 8:00 AM Pacific Time.

https://aka.ms/officeaddinsamples-call

Q&A

1. I'd love to follow-up on this feature request: Office JS Save and Close methods for Word, Excel and PowerPoint Apologies for my persistence... but this is literarily the only blocking issue for us to move from the old add-ins to the JS add-ins platform. We would really really appreciate if you could push this into the roadmap (and we represent many clients with many employees, so do not take the 20 upvotes at face value, we represent hundreds of users and the deployments of the old add-ins and updates are not ideal...). Thank you so much.

Answer:

Thanks for your persistence! We're sorry you are being blocked by this. Unfortunately we can't add this to the roadmap at this time, but we will continue to keep an eye on the request.

As a workaround you may want to explore continuing to use your COM add-in for full functionality while adding partial functionality in the web add-in for cross-platform support. We recommend exploring this article for more information. Make your Office Add-in compatible with an existing COM add-in

Q&A

2. Open Outlook mobile API to compose scenarios to insert attachments in emails. Currently, third party cloud storage companies cannot integrate directly into Outlook mobile. With this API, we would be able to click a button and insert files from private cloud storage providers into emails. Thank you

Answer:

We'd like to know more about this scenario. Can you please create a feature request with more information at https://aka.ms/m365dev-suggestions? Thanks!

Q&A

3. How to communicate from a Word add-in to Office JS task pane? For example, OnContentEnter, highlight one HTML control in the task pane.

Answer:
We see this question was also posted to the Q&A site. It seems like you should be able to use the onSelectionChanged event. If not can you please follow up with more information on the Q&A site? Thanks!

4. Are you planning on upgrading the Visual Studio Web Add-in Templates to ASP.NET Core?

Answer:
We are looking into some updates to the Visual Studio Office Add-in templates, but don't have anything on the roadmap at this time. Can you let us know more about what you'd like to see in these templates by posting a request at https://aka.ms/m365dev-suggestions? Thanks!

Q&A

5. Has the PowerPoint API 1.3 been officially released to all platforms?

Answer:
The PowerPoint JavaScript API 1.3 requirement set is supported on Windows, Mac, and the web (but not iOS or mobile at this time.) See [PowerPoint JavaScript API requirement sets]() for complete details.

Q&A

I have two questions both relating to getAccessToken

(1) Can you clarify which namespace we should be using to access the getAccessToken method please? The Microsoft answer to this post says that we should be using the OfficeRuntime.Auth.getAccessToken. But the third sentence on this page says that the "Office.auth.getAccessToken" will be the one to receive updates in the future.

(2) Why does the documentation refer to "Office.auth" when the full namespace is "Office.context.auth"? Could you confirm that the full namespace is actually Office.context.auth.getAccessToken and not Office.auth.getAccessToken

Answer:
You can use either OfficeRuntime.auth.getAccessToken(), or Office.auth.getAccessToken(). Both are equivalent. To be consistent we're in the process of updating docs/samples to always use Office.auth.getAccessToken().
https://docs.microsoft.com/en-us/javascript/api/office-runtime/officeruntime.auth?view=common-js-preview

Q&A

During the last call, one of my questions was if it would be possible to programmatically let the compose window disappear after the user clicked on a custom "Encrypt and Send" button, as the mail already is encrypted, and send as MIME payload via the send message Graph API.

You suggested to use smart alerts to handle the send event, however, this is not ideal for our use-case, as the on send event is triggered on each email, but we want that the user deliberately decides via the custom button if she/he wants to encrypt the email or not…

Answer:

We’ll work to follow up in the blog post with more info.

Q&A

Is there a way for the admin to install an Excel add-in for the user or group that will show only for some Excel files generated by our application and not when the user creates a new Excel file or some other Excel files opened by the user?

Answer:
We’ll follow up in the blog post with more info.

New to Office Add-ins?

Documentation	https://aka.ms/office-add-ins-docs
Quick starts	Outlook, Excel, Word, PowerPoint
Script Lab	https://aka.ms/getscriptlab
Samples	https://aka.ms/officeaddinsamples
Microsoft 365 Developer Program	https://aka.ms/M365devprogram
Office Scripts	aka.ms/office-scripts-docs

Engage with the team

Engagement	Where to go
Technical questions about Office Add-ins	https://aka.ms/office-addins-dev-questions **[office-js-dev] [office-addins-dev]** https://stackoverflow.com **[office-js] [outlook-web-addins] [office-scripts]**
Product bugs	https://github.com/OfficeDev/**office-js**
Feature requests and ideas	https://aka.ms/m365dev-suggestions
Send a Smile or Frown	Inside Office

Microsoft 365 developer community calls

Microsoft 365 platform call (weekly)	https://aka.ms/m365-dev-call

Adaptive Cards (monthly)	https://aka.ms/adaptivecardscommunitycall
Microsoft identity platform (monthly)	https://aka.ms/IDDevCommunityCalendar
Office Add-ins (monthly)	https://aka.ms/officeaddinscommunitycall
PowerApps (monthly)	https://aka.ms/PowerAppsMonthlyCall

Microsoft 365 Community (bi-weekly)	https://aka.ms/m365-dev-sig
SharePoint Framework (bi-weekly)	https://aka.ms/spdev-spfx-call

All calls and assets: **https://aka.ms/m365pnp**

Thank you

Recording will be available in 24 hours on
Microsoft 365 Community (PnP) YouTube channel
https://aka.ms/M365PnP/videos
(subscribe today)

Follow us on Twitter
@Microsoft365Dev | @M365PnP

Next call: **June 8th at 8:00 AM Pacific Time**
https://aka.ms/officeaddinscommunitycall
Agenda: https://aka.ms/officeaddinsagenda
Submit your questions: https://aka.ms/officeaddinsform

Microsoft