

Office Add-ins community call–February 10, 2021

February's call, hosted by **David Chesnut**, featured the following presenters and topics:

- **Juan Balmori** showed how to set up compatibility between an Outlook add-in and its COM/VSTO counterpart.
- **Abid Rahman** announced that the team is looking into building out Word APIs. He created a [Word Extensibility Survey](#) to gather feedback on top scenarios and blockers for developers.
- **Aishwarya Iyer** provided an update on PowerPoint APIs progress, including what APIs are released, in preview, and under development.
- **David Chesnut** showed a new PnP sample that demonstrates how to build a contextual tab that only appears on the Microsoft Excel ribbon when the focus is in a table.

[Watch the call](#)

Q&A

We welcome you to submit questions and topic suggestions prior to each call by using our short [survey form](#).

Outlook add-in compatibility with COM/VSTO

You mentioned a registry entry option to suppress the web add-in if a COM/VSTO add-in is present. Does Excel, Word and PowerPoint also require a registry entry?

That option is only available for Outlook add-ins, for now.

Will VTSO be phased out over the new office add-ins in the future? Or are both planned to be supported indefinitely?

VSTO will continue to be supported. Depending on your scenario, a web add-in can be an awesome alternative for your Microsoft 365 customers, take advantage of cross-platform support, and improved deployment management.

Word APIs discussion

Do you have plans to make the "Show as saved image" content pane feature available from the API? Also, it would be great to have this function automatically triggered in case the required add-in is not available/installed in the system.

If referring to the “show as saved image” capability for content add-ins, we haven’t had an API ask for that specific capability. It would be great if you could log a request on [Office Developer Platform UserVoice](#). The second idea is a great suggestion, as it can make the experience better for users on the downstream version. Can you log a UserVoice request for that as well?

How long will the survey form for Word APIs be open?

We can keep it open until the end of February. If you need additional time, please let us know during the March community call and we can follow up directly.

PowerPoint APIs Updates

How do you get access to the beta channels for PowerPoint?

You’ll want to [become an Office Insider](#). And we recommend that you get a tenant on the [Microsoft 365 Developer Program](#).

Contextual tab PnP sample

Can the contextual tab get auto selected?

Custom contextual tabs show or hide on demand based on an API call. They are hidden by default. The guiding principle we have is to provide an experience for custom contextual tabs that is similar to regular contextual tabs. For example, when you select a table in Excel, the "Table Design" tab shows but it does not have focus. Custom contextual tabs follow the same principle.

Can we localize the labels in contextual tabs?

Yes, for more information, see the localization section in [Create custom contextual tabs in Office Add-ins \(preview\)](#).

Can contextual tabs display info in the task pane?

Yes. When your JavaScript function is called (such as in response to the user selecting a button on your contextual tab) it has access to the DOM of the task pane and can make updates.

Does the function called by the contextual tab run in the same process as the task pane?

Yes. The add-in uses the shared runtime so that all code is sharing the same JavaScript runtime. For more information, see [Configure your Office Add-in to use a shared JavaScript runtime](#).

Miscellaneous questions

Is it possible for an Office Add-in to run Node.js processes, such as reading a database?

You can have a web app/server side process running Node.js that can reach out to read the database. For example, see [Create a Node.js Office Add-in that uses single sign-on](#).

In Excel there seems to be a data connection wizard. Would it be possible to reuse that wizard in an Excel add-in?

No. We don't currently have data connection APIs. We are considering APIs that let you refresh existing models in Power Query, however. It would be great if you could log a request for this on [Office Developer Platform UserVoice](#).

Our customers have a persistent ADD-IN ERROR that we think is caused by a corrupt legacy Microsoft Edge installation, and the only solution is to rebuild the user's Windows profile. Users are unable to upgrade to a newer Microsoft Office to enable WebView2, so they are stuck with the error. Please check for more details in [GitHub issues 1620, 1356](#), and also this [StackOverflow thread](#).

We are continuing to investigate the issue. The only current workaround is to update to Webview2. We understand this is not an ideal workaround and will share more once we have an update.

For the addFromBase64 API in Excel, what is the status of opening a collection of worksheets with the current workbook online? Is it going to be on production soon?

Yes. The team is actively working on this feature and we are targeting it to launch with our next Excel 1.13 API requirement set, which will be in about a quarter.

Resources

From February's call

- [Outlook for Mac – Planned update for injection-based plugins](#)
- [Word Extensibility Survey](#)
- [PowerPoint JavaScript preview APIs](#)
- [What's new in PowerPoint JavaScript API 1.2](#)
- [PnP: Contextual tabs sample](#)
- [Office Scripts AMA – February 16, 2021, 9:00-10:00AM Pacific Time](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)

- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, March 10, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please contact us by completing this short [survey form](#).

David Chesnut

Office Add-ins community call–January 13, 2021

November's call, hosted by **David Chesnut**, featured the following presenters and topics:

Michael Aldridge and **Mairead Bushe** walked through best practices and guidelines for submitting your Office Add-in to the store.

Juan Balmori and **Kwame Ansong-Dwamena** shared a preview update and some demos of event-based activation for Outlook add-ins.

To watch the call on YouTube, click the following thumbnail



or tap the following link: [Office Add-ins community call January 13, 2021](#)

Q&A (Question & Answers)

We welcome you to submit questions and topic suggestions prior to each call by using our short [survey form](#).

New Microsoft 365 app submission experience

Regarding the availability date, is there a time limit? For example, "must go live within 3 months"?

No, there's no limit here. First time approvals can take multiple submission cycles, so we recommend that you leave yourself sufficient time if you are scheduling a specific launch date.

Is there a minimum character requirement for submission notes? I tried to update our app today, but I could not figure out how to get the 'publish' button enabled.

If you have entered some notes for the certification team, there shouldn't be a minimum character limit. When you tap publish on the Notes for certification page, it should bring up a quick summary page which would show you if any other page is blocked. If you can't get unblocked, open a ticket with our support team and we'll be happy to help.

<https://aka.ms/marketplacepublishersupport>

Are there any specific steps to publish a trial mode add-in or do we follow the same steps for add-in submission? The sort of sample add-in like the Contoso examples on the forum. This is for developers to verify and check.

Yes, we have beta release capability on the backlog with the Partner Center team.

Outlook event-driven add-ins

Is there a way for the function to access data that could be generated by the task pane?

We have an API for add-ins to communicate or share transient data during an item compose session by using a property bag. Once the item form is closed, the item session is also closed, and the data does **not** persist. To persist data beyond the session, use [roaming settings or custom properties](#).

Would the property bag be on a per mail basis like the existing itemProps?

Yes. It is per item session. It is not persistent. We will purge the data once the item UI or form is closed. The in-memory property bag will be available only during the compose of the email; hence only relevant to the current item (in this case an email).

Is there an ETA for Event Driven support for Outlook for Mac?

Our goal is the second half of CY2021.

Is there a list of all events you are planning to expose?

We will start with the following two events:

- OnNewMessageCompose
- OnNewAppointmentOrganizer

Future events will include the following:

- OnMessageAttachmentsChanged
- OnAppointmentAttachmentsChanged
- OnMessageRecipientsChanged
- OnAppointmentAttendeesChanged
- OnAppointmentTimeChanged
- OnAppointmentRecurrenceChanged
- OnInfobarDismissClicked

There are going to be two special events for the On-Send feature:

- OnMessageSend
- OnAppointmentSend

The demo showed accessing Office JS, and you explained where that came from. Usually, you add a reference to the Office.JS scripts yourself (being able to switch to beta and whatnot) but you said this was injected by the sandbox. Can I still shift between versions such as beta or other versions?

Yes, it is injected into the sandbox. If you want to use beta APIs, we have a registry key that you can set to place Outlook in dev-mode. In dev-mode, you can use beta APIs.

Is there an onAttachmentAdded event?

There's an event for attachment changed.

Is there any plan for an Outlook startup event and frequent event, for example every 30 minutes after startup?

Thanks for this suggestion. At this point we don't have an Outlook started contemplated but we will be interested in learning more about the scenarios you have in mind. This is challenging as usually in that code path we don't want to have too much logic going on to let Outlook start fast.

Are you targeting Ignite 2021 for event-based add-ins?

We are hoping to be earlier than that.

We are enthusiastic about the new Event-based activation combined with the InsightMessage notification with a custom action you announced on <https://docs.microsoft.com/en-us/office/dev/add-ins/reference/objectmodel/preview-requirement-set/outlook-requirement-set-preview> and presented in one of the latest community calls. This is currently available in preview in Outlook on the web and we are hoping to have it available in the new 2.0 requirement set. Could you please give an estimation when this is planned to be available for Outlook on Windows clients?

We need your help to release it as soon as possible. We are finishing the feature and will be in win32 soon. Right now, we are feeling good about where the feature is at, but we need folks in

the developer community to give feedback and help to drive quality. The goal is to release in the first quarter of this year.

Miscellaneous questions

What is the most effective way to debug .exe Outlook add-ins? I saw a video that mentioned sideloading for Outlook was on the way (see <https://www.youtube.com/watch?v=qD6eTKjg4fE>) but I couldn't find the exact date or date range.

This debug feature just went to GA, so you should see it appear any day.

When can we expect WebView2 with Edge Chromium to be generally available in our Office Add-Ins?

We expect it to become widely available starting in early February 2021. The actual timelines may vary depending on your Microsoft 365 plan: consumer, education, enterprise, etc.

For EWS and REST API usage, Office.js provided the getCallbackTokenAsync to get an authorized token for making EWS or REST API calls. With the push towards the Graph API, can you provide a version of this method to authenticate with Graph? i.e., that requires only to install the add-in and then perform this call through office.js, rather than needing to perform additional external setup. getAccessToken is not equivalent to this.

Although it is a bit more complex, to ensure better security, we now use the OAuth flows for Outlook SSO and Microsoft Graph. There unfortunately isn't a way to get a token without registering your app with Azure Active Directory.

Is it possible to automatically pin an add-in (manifest enforcing pin) persistently across compose mode/read mode?

Unfortunately, there isn't a way to achieve this now.

**We have an Excel add-in that functions correctly when running in a standalone browser like Chrome or on Excel Desktop. Whenever we run it in Excel (iframe) on the web we get CSRF validation errors, and this is because of the SameSite attribute in cookies not being set properly. Does the Office Add-in team have any workaround or solution regarding this that does not include disabling the two flags in Chrome and Edge?
[We've logged this in UserVoice as well.](#)**

We need some more information. Can you please post this issue to <https://stackoverflow.com> using the tag [office-js]? UserVoice is for feature suggestions so it will be hard to answer there.

Is there an official date set for the release of the latest version of PowerPoint API which is currently in preview?

The PowerPoint team doesn't have a specific date yet, but they are actively working hard on getting APIs ready for GA this quarter. Keep providing feedback and stay tuned to this community call for future updates.

Is there any information regarding the 13012 error when establishing SSO with our enterprise AAD?

We have troubleshooting information about this error code and others at the following link: [Troubleshoot error messages for single sign-on \(SSO\) - Office Add-ins | Microsoft Docs](#). If you are still stuck please post a question on <https://stackoverflow.com> using the tag [office-js] or [outlook-web-addins].

Resources

From January's call

- [Microsoft AppSource](#)
- [App validation documentation](#) with "Am I ready to publish?" checklist.
- [Microsoft Partner Network](#)
- [Craft effective Microsoft AppSource store images](#)
- [Configure your Outlook add-in for event-based activation \(preview\)](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday**, February 10, 2020 at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please contact us by completing this short [survey form](#).

Office Add-ins community call–December 9, 2020

November's call, hosted by **David Chesnut**, featured the following presenters and topics:

- **Raymond Lu** shared updates on the addFromBase64 AP, and some additional Excel APIs.
- **Nisarg Kinariwala** showed how to use Office Add-in usage reports to check usage data about your Office Add-in.
- **David Chesnut** demonstrated a new PnP sample showing how you can use web workers in custom functions to keep you task pane UI from being blocked.

Watch the call [here](#).

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

AddFromBase64 API updates

- **Are there plans to update the base64 APIs for Word soon?**

This update is focused on the Excel API which was in beta. We don't have any current plans to update this one but if you have issues with it, please log a bug on our office-js Github repo, or create a new feature request on UserVoice (<https://officespdev.uservoice.com>).

- **Will the insertFromBase64 API be kept?**

If referring to the old API, addFromBase64, then, yes. We don't have plans to deprecate the addFromBase64 beta API. But we recommend to use the new insertFromBase64 API after it releases to GA.

Miscellaneous questions

- **Are there plans to update the JavaScript API for PowerPoint to broaden access? Example: allow access to interact with the presentation directly.**

Yes, we are starting back up on the PowerPoint APIs. We covered this in the September community call where you can find more info: [Office Add-ins community call–September 9, 2020 - Microsoft 365 Developer Blog](#).

- **Any plans for insert slide in PPT?**

Yes, this is in preview. See [Insert and delete slides in a PowerPoint presentation](#).

- **Is the open workbook like in Word also available in Excel?**
`context.application.createDocument(mybase64).open();`

Yes it's available in Excel. See [Excel.createWorkbook](#).

- **Do you have any plans to move the shared runtime from IE 11 to EdgeView2?**

Yes, as covered in the November call, Sudheer talked about this capability that is rolling out in our November fork. If all goes well, you should be able to try it in the monthly channel in mid-late January. For more info, see [Office Add-ins community call–November 11, 2020 - Microsoft 365 Developer Blog](#).

- **What's the expected timeline for resolving add-in instability on the Android platform as seen in backlog issue <https://github.com/OfficeDev/office-js/issues/1027>?**

Unfortunately, we still don't have a timeline around this issue. It's still on the backlog, and we'll post an update to the GitHub issue once we have more information.

- **There are two open entries on UserVoice asking for a solution to sending an email in Outlook compose view using the office.js API. Specifically, to handle Outlook on Window's cached mode causing delays (of more than 60+ seconds in some cases) when trying to use EWS or REST to send the email. Are there any plans or advised workarounds to help developers improve this scenario for end-users?**

Can you provide us links to the UserVoice requests, or more details? Or please post the details to StackOverflow and tag it [office-js][outlook-web-addins]. Thanks!

- **With the deprecation of the Outlook REST API v2.0 (to be decommissioned in November 2022) are there plans to provide a similar seamless usage API option in the add-in API? Seamless as in requires no user interaction e.g. via pop-up, to obtain a usable token client-side as there is today with `getCallbackTokenAsync`.**

We recommend moving from Outlook REST API to Microsoft Graph. A good starting point would be SSO `getAccessTokenAsync` and use the Graph API. This recommendation is documented here: [\[DEPRECATED\] Outlook Mail REST API reference \(version 2.0\) | Microsoft Docs](#)

Resources

From December's call

- [Create an account in the Partner Center](#)
- [Publish your app to Microsoft AppSource via Partner Center](#)
- [Analyze the usage of your app](#)
- [Analyze the acquisition of your app](#)

- Feedback on Office Add-in usage report: nisark@microsoft.com
- [PnP: web workers in custom functions PR](#)
- [Lance Austins's web worker, IndexedDB, web socket, service worker sample from May 2019 community call](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, December 9, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Office Add-ins community call–November 11, 2020

November’s call, hosted by **David Chesnut**, featured the following presenters and topics:

- **Sudheer Maremanda** talked about our Webview2 plans going forward.
- **Matt Geimer** introduced the new Microsoft 365 certification for Office Add-ins.
- **Abid Rahman** showed the new shortcut keys and contextual tabs features that are now in public preview.
- **Courtney Owen** showed a demonstration of how to debug Outlook add-ins with Visual Studio Code.

Watch the call [here](#).

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

Shortcut keys and contextual tabs

Can shortcut keystrokes be customized by the end user? This is an important feature for users.

Customization of shortcuts are currently not supported, but we are working on it! We may have an update around customizations sometime later in 2021.

Would you be able to assign a keyboard shortcut to a ribbon command that currently does not have one? For example, assign a keyboard shortcut to Insert->Add-Ins->Admin Managed->Refresh.

Unfortunately not. Keyboard shortcuts can only invoke add-in related actions (from within the add-in’s runtime). We currently do not support custom shortcuts for native application actions.

Can keyboard shortcuts invoke functionality in the task pane or only invoke commands defined in the manifest? My add-in currently implements its own keyboard event handler. Will the new keyboard shortcuts provide another way to handle keyboard events?

You can invoke any JavaScript functions you have, including the Office.js API. The actions invoked by shortcuts all require callback functions (defined in the task pane/add-in runtime). The pattern is similar to ExecuteFunction commands for add-in commands. The actions must be defined in the JSON, but their corresponding JS code is in your task pane/runtime.

For your add-in's keyboard event handler, we're assuming it is only able to handle keystrokes done while the keyboard focus is within the task pane. For add-in shortcuts, you'll be able to expand that functionality so users can use shortcuts when the keyboard focus is also within the worksheet.

What happens in the case where multiple add-ins have the same shortcuts?

Currently, this behavior is undefined for public preview. For GA, we will provide a modal dialog to the user the first time they use a conflicting shortcut, which will allow them to select which add-in action to invoke with that shortcut.

Is there a way to programmatically set focus to the worksheet from the task pane?

We currently do not have a focus API.

Does the keyboard focus control keyboard shortcut availability, or the mouse location?

The keyboard focus must be in the worksheet. It's, not the mouse location, however, if as a user you click inside the task pane with the mouse, the keyboard focus will switch to the task pane.

For keyboard shortcuts... Is this limitation (task pane focus) going to be improved?

We are currently investigating ways to improve this limitation. As a workaround, add-ins can currently make key handlers inside their add-ins that can handle key presses when the keyboard focus is inside the add-in's UI.

Are there any plans to remove ribbon commands xml from the versionoverrides section of the manifest.xml and replace it with a standalone json file - similar to how the contextual tabs are working?

We have been experimenting with that idea and are investigating if that is possible in the future. We are taking small steps towards that goal by introducing contextual tabs APIs. The contextual tabs APIs require a JS object that defines the ribbon.

For contextual tabs, will it be possible to have a "super/grouping" tab? For example, "Chart Tools" groups the native "Design" and "Format" tabs.

Unfortunately, it is not possible to change the existing native tabs. For example, you cannot combine native tabs into a super tab/show or hide native tabs, etc.... These functionalities are only available for custom tabs created by the add-in.

What if I want to group my custom tabs under a "super" tab similar to the way that those two native tabs group.

You can definitely do that. You can create three custom tabs, for example, Tab 1, Tab 2, and Tab 3. Tab 3 can be a "super" tab that includes all your add-in commands. You'd be able to show/hide tabs 1, 2, and 3 depending on the user's context.

M365 add-in certification

What is the value of doing the M365 app certification, that differs from the self-attestation?

Self-attestation is a great step for providing documentation, but it doesn't go as deep into the various security domains and doesn't include a 3rd party assessment on that. The different areas that are considered for both are available at <https://aka.ms/appcertification>

Will the Microsoft Certification program Certification hold more power than the "preferred solution" ribbon or would they be shown side by side?

Please take a look at the [Microsoft 365 App Compliance program blog post](#) for more information.

Visual Studio Code debug

Does the debug of OnSend only work on the beta channel version?

Debug of OnSend will be available soon in production/GA.

Roadmap and PowerPoint API questions

I'd love to follow-up on one of the questions from the October community call: "*In Word we have 'context.document.save()' and in Excel we have 'context.workbook.save()' – so question is, when will this be implemented in PowerPoint? We are missing a method to close the active document – is this planned and if yes, when will it be released? This is related to both Word, Excel and PowerPoint.*" The response, as always, was to post on user voice.... Please any chance you can follow-up on this and help the community with any info / estimate?

Unfortunately, we don't have specific ETAs yet, but here are some updates from the team to the PowerPoint API questions asked in the October community call:

In Word we have 'context.document.save()' and in Excel we have 'context.workbook.save()' – so question is, when will this be implemented in PowerPoint?

context.presentation.save() API is on the roadmap but we do not have an ETA that we can share at this stage. We are prioritizing the API's based on customer feedback, so make sure to create a request in [Office Developer Platform UserVoice](#) or upvote if there is an existing request already.

We are missing a method to close the active document – is this planned and if yes, when will it be released? This is related to both Word, Excel and PowerPoint.

We have added PowerPoint, presentation.close() API to the roadmap but we do not have an ETA that we can share at this stage. We are prioritizing the API's based on customer feedback, so make sure to create a request in [Office Developer Platform UserVoice](#) or upvote if there is an existing request already.

We'd like to see better parity between the specific host PowerPoint API with the control that was available in VSTO add-ins for PowerPoint. We have an add-in we want to build out in the new web add-in platform, but can't due to the extremely limited capabilities of the current Office.js API.

We have recently started focusing on reducing the PowerPoint's OfficeJS surface gap. The first set of PowerPoint OfficeJS API's are available in the preview channel and more will be coming soon.

Is there an accessible updated roadmap for where the office-js api is heading and what features are being developed on what timescale for which office applications? This would be really helpful. I have seen several requests for this in the past. Also, is there a plan to follow up on this request? <https://officespdev.uservoice.com/forums/224641-feature-requests-and-feedback/suggestions/33200554-powerpoint-api>

We definitely hear you that we need a better roadmap and communication around API plans. This is something we're working on and hope to share more information soon.

For now, the best way to track new features and updates coming out for preview or general availability is to keep an eye on the [Microsoft 365 developer blog](#). We of course also cover upcoming schedule and plans in this community call.

We also have host specific preview APIs that are in development which you can participate and provide feedback on.

- [Excel JavaScript preview APIs](#)
- [Outlook preview API reference](#)
- [PowerPoint JavaScript preview APIs](#)

Miscellaneous questions

The only way I have been able to successfully debug and step through code is by using Visual Studio 2019. I am currently running into an issue with debugging if I use create document because it loads in a new Word context. Any advice?

There's a section in our [docs on debugging](#) that may help. Otherwise please post a question on Stack Overflow tagged [office-js] with more details so folks can take a look and help.

We have developed an Office Add-in for Excel, when an updated manifest is uploaded, the next time Excel is opened, the add-in should be updated. While this holds true for the add-in panel, the command ribbon is not showing the latest changes for all users. Some see it

right away, but others have to go into Admin Managed and do a refresh to get the latest, or clear the cache in %LOCALAPPDATA%\Microsoft\Office\16.0\Wef\. Is there a way to force an update command ribbon without the end user having to do a Admin Managed-Refresh or clear the cache manually?

Can you please log an issue on GitHub at: <https://github.com/officedev/office-js/issues> please? This sounds like a bug, as commands should refresh when the store add-in is updated.

Can the yeoman scaffolding use a .NET server instead of starting its own?

Yes, it can use any server. You would need to update the yo office project start script to not call the dev-server script.

I'm running the following code from the [getRestApiId\(\) documentation](#):

```
OneNote.run(function(ctx) {
    // Get the current page.
    var page = ctx.application.getActivePage();
    var restApiId = page.getRestApiId();

    return ctx.sync().
        then(function() {
            console.log("The REST API ID is " + restApiId.value);
            // Note that the REST API ID isn't all you need to interact with
the OneNote REST API.
            // This is only required for SharePoint notebooks. baseUrl will
be null for OneDrive notebooks.
            // For SharePoint notebooks, the notebook baseUrl should be used
to talk to the OneNote REST API
            // according to the OneNote Development Blog.
            // https://blogs.msdn.microsoft.com/onenotedev/2015/06/11/and-
sharepoint-makes-three/
        });
});
```

However, the result is always empty. If I change to getActiveNotebook() everything seems to work fine. The OneNote file is stored on personal OneDrive. Is this a bug? I'm blocked.

Sorry that you are running into this. This does look like a bug, so we are investigating. Can you post an issue on GitHub at <https://github.com/officedev/office-js/issues>? We'll post any updates to the issue.

In the new Outlook for Mac experience, can support be added for add-in individual buttons to be displayed on the ribbon rather than grouped under an add-in icon? For multi-button add-ins the grouping is going to impact our user-uptake if sensible button labels are not seen in the ribbon by default.

Thank you for the feedback. We are evaluating this request currently and will let folks know in the near future if we plan to add this support.

Will there be support for admins automatically pin add-ins to the taskbar in the new Outlook for Mac UI? By default, the add-in is not visible on screen and hidden under the '...' menu.

We do not have a capability for add-ins to be pinned by Admins; however, users can customize their own toolbar and pin add-ins. We are exploring various measures such as a first-run experience and pinning frequently used add-ins to improve add-in discoverability.

Regarding the recent change to Exchange on-premise OWA where if a security update had not been installed on the server within two days add-ins stopped working, can the add-in team do something to improve communication of such changes so that developers are aware beforehand and can prepare steps for customers who encounter issues when a new version of the office.js scripts are deployed?

The [security update](#) was essential to improve the security around add-ins across all Office products. The nature of the vulnerability required customers to update to the latest security update and prevented us from sharing information around it. However, we appreciate your feedback on providing additional documentation around these types of releases and will work towards better communication in the future.

My understanding is that other than on the iPad (where Word and Excel are supported as well), the only mobile app that allows JS add-ins is Outlook. Are there plans to add add-in support to Word and Excel on iOS and Android? Will Outlook support add-ins for Gmail accounts (currently I understand only Exchange supported?)

Currently we don't have plans this year to do more on iPad for Word, Excel, or PowerPoint.

Outlook Add-ins have client and server requirements, and therefore not supported elsewhere but Exchange. There's more information in the article, [Outlook JavaScript API requirement sets](#).

When required to use EWS with on-premises Exchange, my Outlook add-in gets a bearer token with getCallbackTokenAsync() and the endpoint from the property Office.Context.Mailbox.ewsUrl. When the external and internal EWS URL differ, the token correctly target the external EWS URL (confirmed by its audience claim), but the ewsUrl is almost always the internal EWS URL, so the endpoint can't be called from a third party service (like my server) and even if it could, the token would be invalid. Is there an additional requirement to get or set the external EWS URL when running on an on-premises mailbox?

We need more information to help answer. Can you post this question as an issue on <https://github.com/officedev/office-js>? It would help to know the client version, server version, and repro steps. We can follow up from there.

Resources

From November's call

- WebView2: [Announcing WebView2 for Office Add-ins platform – preview](#)
- Keyboard shortcuts: [Add Custom keyboard shortcuts to your Office Add-ins \(preview\)](#)
- Keyboard shortcuts (PnP sample): [Use keyboard shortcuts for Office add-in actions](#)
- Microsoft 365 certification: [Partner Center](#)
- Microsoft 365 certification: [Microsoft 365 App Compliance Program](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, December 9, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Office Add-ins community call–October 14, 2020

October's call, hosted by **David Chesnut**, featured the following presenters and topics:

- **Courtney Owen** showed a demonstration of how to register Office Add-ins in Office on the web using URL query parameters. You can effectively sideload add-ins to Office on the web.
- **Juan Balmori Labra** presented recent updates and announcements for Outlook add-ins. He also talked about optional connected experiences.
- **Keyur Patel** presented recent updates and announcements for Excel add-ins.
- **David Chesnut** demonstrated a new PnP article and sample code showing how to open your add-in in Excel from your SaaS web application.

Watch the call [here](#).

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

We'd like to see better parity between the specific host PowerPoint API with the control that was available in VSTO add-ins for PowerPoint. We have an add-in we want to build out in the new web add-in platform, but can't due to the extremely limited capabilities of the current Office.js API.

What's the status on the limited capabilities of the current Office.js API?

We are building the OfficeJS API surface area for PowerPoint based on the scenarios that we can enable. For example, in the [Office Add-ins community call \(September 2020\)](#) we announced a few new scenarios we will support with new APIs.

VSTO is a huge API surface, which is why we approach building out the Office JS API based on prioritized scenarios. It is tremendously helpful to understand the scenarios you are trying to unlock and what API's are needed. That helps us prioritize. Let us know which scenarios you need enabled on our UserVoice page at <https://officespdev.uservoice.com/>. We are planning to deliver increased API surface area for PowerPoint going forward.

In Word we have 'context.document.save()' and in Excel we have 'context.workbook.save()' - so question is, when will this be implemented in PowerPoint? We are missing a method to close the active document - is this planned and if yes, when will it be released? This is related to both Word, Excel and PowerPoint.

Please let us know which scenarios you need enabled on our UserVoice page at <https://officespdev.uservoice.com/>. It helps us prioritize as we plan API development going forward. There are several save and close related requests there you might want to take a look at or upvote.

BTW: In Excel you can close the workbook using the following code:

```
async function run() {
  return Excel.run(function (context) {
    context.workbook.close(Excel.CloseBehavior.save);
    return context.sync();
  });
}
```

Are there plans to make it possible to develop Office add-ins using C#? Using Blazor, for example? Blazor runs in a browser and supports JavaScript interop. (<https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor> <https://docs.microsoft.com/en-us/aspnet/core/blazor/call-javascript-from-dotnet>)

We don't have support for Blazor today as we're currently focused on JavaScript. Please let us know more about this scenario on our UserVoice page at <https://officespdev.uservoice.com/> (There's actually already a [Blazor request](#) there you can upvote.)

Do we have a list of the benefits of WebView2 vs WebView?

A couple things are:

- It is available to all versions of Windows.
- It uses the Chromium runtime and provides more consistency across browsers.
- And one of our favorites is that it brings right-click > Inspect Element to the context menu.

For more information see the blog, [Announcing WebView2 for Office Add-ins Platform](#)

I'd like to make sure I understand the essence of [this blog](#) around the optional connected services and add-ins. Is it the case that with the right Office version now, one can just use centralized deployment and everything works? Is there anything else they need to know or configure?

The short answer is yes. The feature is currently released for the monthly channel. Semi Annual is in preview on the Semi Annual Channel Insiders Fast, and will be included in the January 2021 Semi Annual Release.

When can we try out registering Office Add-ins in Office on the web using URL query parameters?

Please note, that you must have a dev-server running and your project manifest must be served up by the dev-server. You can try this feature in preview on PowerPoint, OneNote, and Word by joining [Office insiders](#).

How do you unregister an add-in?

You can unregister them through the add-in UI or by clearing your Application cache in the browser. You can also right-click > remove the add-in command.

I am experiencing an issue getting the VS Code extension to step into code at a breakpoint. Any advice on troubleshooting that?

Debugging can be tricky. Often it is a combination of using the right WebView, used OfficeJS version, used Office versions and more. See [Test and debug Office Add-ins](#) for more help with various debugging scenarios.

What is Fast Food version and could you explain a bit more about Inner rings and Outer rings for identifying which version you have installed?

Courtney meant Insiders Fast. Just go to **File > Account** and there should be a button to select the channel you need. Look for insiders fast, and then just update Office. You can find more information on the Office insider program at <https://insider.office.com/>.

My organization is in a GCC High Tenant. This limits the plugins and features available to me as a developer when using the web hosted versions of the Apps. How can I get access to Script Lab/ Edge DevTools without them being available in our tenant?

You can get your own developer subscription of Microsoft 365 by joining our developer program: <https://aka.ms/m365devprogram>. We recommend everyone building add-ins join that. It gives you a free, renewable account for development purposes with 25 users.

Also, add-ins are actually supported in GCC High environments. But, what might not be enabled is the communication that the add-in needs to do with external services. In house developed add-ins should be doable.

How do I debug event add-ins in Outlook ?

We are actually adding a feature in the tooling to do this for Outlook on Windows.

When will you support compose mode in mobile add-ins?

This is something that is in our backlog and we expect to tackle this soon.

What is Preview - does that mean available to Office insiders ?

Yes it's Office Insiders Fast. There is an Insiders for Semi Annual Channel. Actually the demo was using the Preview for Semi Annual.

Where can you submit a request for an addition to the JavaScript Excel API (for example, for a feature that was in the C# API)?

We use UserVoice for this. check out of there is an existing Idea or create a new one at <https://officespdev.uservoice.com/>

Regarding the contextual ribbon tabs, one of the things that we've always wanted was the ability to "group" customized tabs the way that Excel does. For example, if you click into an Excel table/ListObject, there are two grouped contextual tabs (Design and Layout).

Yes! That's exactly what the contextual ribbon tab APIs are about. We'll have guidance on how you can try this out in a future community call.

What's needed in addition to Contextual Ribbon Tabs is the ability to create checkboxes, radio buttons, separators, and active/inactive elements. in a custom ribbon tab.

Yes, additional UX controls are also on the backlog. We haven't started on that specifically, but we hope to bring more flexibility to the commands space.

If there are any specific controls you're interested in, please upvote or log a new request on UserVoice as it will help our user experience team prioritize: <https://officespdev.uservoice.com>

Are there any plans to update the templates for creating Office add-ins using Visual Studio (as opposed to Visual Studio Code / Yeoman)

We aren't able to update those templates as frequently. However, there is a new template for Office UI Fabric React for Visual Studio 2019 available here: <https://github.com/OfficeDev/PnP-OfficeAddins/tree/master/Samples/Office.ASP.NETCoreReactAddins>.

Is there similar functionality like the PnP Open in Excel sample available for PowerPoint also?

Yes, the same principles to open in Excel also apply to Word and PowerPoint. We wanted to drive focus on Excel in particular because we're seeing a lot of "Export to CSV" file capabilities, which would definitely benefit from this "Open in Excel" pattern.

Is there an example of how to launch a distinct task pane from each ribbon button?

From your add-in commands, you can use the ShowTaskpane action and point to different URLs to show different task panes.

Resources

From October's call

- Outlook add-in resources
 - As Juan mentions in the call, we highly recommend the <https://aka.ms/officeaddinsignite2020> to end users/IT admins who are interested in Office add-ins. Would also love your feedback on that video to see if it resonates with your customers to help your add-ins get deployed or not.
 - For more information about the Optional Connected Experiences announcement please see [Outlook add-ins and optional connected experiences - Microsoft 365 Developer Blog](#)
 - Seamlessly integrating your solution into Outlook: <https://aka.ms/OutlookIgnite2020>
 - Build Outlook Add-ins <https://aka.ms/OutlookAddinOverview>
 - New preview capabilities for Outlook add-ins <https://aka.ms/PreviewOutlookAddin>
 - Configure Event-activated add-ins for Outlook <https://aka.ms/EventActivatedAddins>
 - Suggest new Outlook add-in features: <https://aka.ms/OutlookUserVoice>
- Microsoft Ignite 2020
 - A detailed blog on Ignite can be found here: <https://developer.microsoft.com/office/blogs/office-at-ignite-2020/>
 - The main ignite blog can be found here: <https://developer.microsoft.com/microsoft-365/blogs/microsoft-teams-microsoft-graph-dev-platform-ignite-2020/>
- [PnP Open in Excel](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, November 11, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please connect with us by completing this short [survey form](#).

Office Add-ins community call–September 9, 2020

September's call, hosted by **Alex Jerabek**, featured the following presenters and topics:

- Lillian Liu and Amit Mohindra presented how to do two-way messaging with dialogs in addition to some new PowerPoint APIs and the PowerPoint API roadmap.
- Nick Gladkov, Vit Holy, and Volodymyr Onyshchuk shared Infor d\EPM's Office add-in features and their experience building on the Office add-in platform.

Watch the call [here](#).

Developer Survey

We want to hear about your experiences building on the Office Platform. Head over to <https://aka.ms/officedevsatisfaction> to take a quick survey.

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

Are there any plans for a PowerPoint API call to save a presentation?

Since we are focusing on PowerPoint Online right now, the save operations are done automatically. So not in the near future.

Can I use a WebView2 control in a PowerPoint task pane add-in?

Per the last community call, we are targeting Webview2 this fall (based on quality). It is currently on insiders and would be available across Word, Excel, PowerPoint and Outlook. You can find more details [here](#).

I noticed you are using an undo function. In Excel it is not possible to use the undo action when there is a write operation done using the office.js API. Is there some chance that it would get into Excel as well?

Undo support is on the backlog, but no ETA yet when we'll be able to get started on this effort. It would be great to consider partner scenarios when we get to planning. It will be a very large effort, so we're focusing engineering on other places where we can move the platform.

Are the PowerPoint APIs only available as an online beta?

Currently, the PowerPoint APIs are in private preview, but moving to public preview soon. Once the APIs have been in preview long enough and get some public usage, we'll move them into GA (general availability).

Are there any plans to roll out custom functions to iPads.

Yes, richer mobile support is in our plans (delivering the great capabilities such as shared runtime, add-in commands, centralized deployment, and custom functions). Right now, we don't have an ETA, as we've been first focused on unblocking partner scenarios first on Mac, Web, and Windows. When we start investing back into iPad, we want to do so in a holistic way to deliver the same experience we have on the other platforms.

Resources

From September's call

- Learn how to use **messageChild()** - [Use the Office dialog API in Office Add-ins](#)
- Availability of DialogAPI 1.2 requirement set - [Dialog API requirement sets](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, October 14, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please connect with us by completing this short [survey form](#).

Office Add-ins community call–August 12, 2020

August’s call, hosted by **Alex Jerabek**, featured the following presenters and topics:

- Juan Balmori Labra showcased the Outlook JavaScript APIs coming to the next version of office.js.
- Matt Geimer demoed the new single sign-on support in Office Add-ins.
- Chris Mentor shared Jedox’s add-in solutions and talked with their team about the experience of building on the platform.

Watch the call [here](#).

Developer Survey

We want to hear about your experiences building on the Office Platform. Head over to <https://aka.ms/officedevsatisfaction> to take a quick survey.

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

How can I add a shortcut key to a Word Add-in Command?

Custom keyboard shortcuts are currently not supported for Office Add-ins on Word.

However, for ribbon buttons (add-in commands) you can use the auto-generated [KeyTips](#) by pressing Alt and using the appropriate keys that pop up in the KeyTips for that specific ribbon tab and buttons.

How can an action (menu item, button) on the Word side execute a JavaScript function in the Add-in?

[Add-in commands](#) can be used to invoke add-in related actions such as showing a task pane or executing a JavaScript function in the add-in.

Would love to hear more about timelines for Webview2 rolling out with Office and any news about improved debugging support from VS Code.

Edge WebView2 is now available in Office Insiders. Here are the steps to try out the preview version:

- Join the Office Insiders, if you aren’t already part of it [\[https://insider.office.com/en-us/\]](https://insider.office.com/en-us/)

- Get Office 365 version **16.0.13127.20082** [or later]
- Get the Edge WebView2 runtime from aka.ms/webview2installer
- Launch your add-in, right click and you should see an option to “Debug”. Selecting this option should launch the new Edge Webview2 debugger.
- Test your add-in and give us feedback on github [<https://github.com/officedev>].

As for debugging support, we would hope to have great debugging capabilities including VS Code debugging. One of the really nice things about the Webview2 control is you'll also be able to simply right click and inspect like you can do in the web apps.

Does this SSO use MSAL, if so does it use Implicit Flow or PKCE ?

It depends on the client as to the library it uses. As of now, it's using Implicit flow

With the new SSO features, if we want to limit our add-ins to Office 365 users (disallow Live accounts), is that possible?

It's something we are looking in to, but don't have it as part of the GA. Would love to know if there are other options you need around account types. You can limit the service app registration to work only for organizational accounts.

Is there an SSO-related event that fires when user switches accounts?

Currently, there isn't an event that lets you know a user switch happened around SSO, but it's something we are looking in to.

Is this SSO capability only limited to AzureAD? What if customers use other iDPs?

Yes, SSO support is limited to AzureAD services. at current time. Other identity providers would still need to be handled via the same method we handle fallback - primarily the dialog API.

Any update on connecting custom functions to asp.net servers?

For your CORS/Preflight, we'd recommend checking the shared runtime: <https://docs.microsoft.com/en-us/office/dev/add-ins/excel/configure-your-add-in-to-use-a-shared-runtime>. This will support Full CORS for custom functions.

Is there a plan to add more than one function (createPresentation) to Powerpoint JavaScript API?

Our team is investigating new APIs right now, if you have specific asks on APIs, please log issues on our UserVoice site: <https://officespdev.uservoice.com>.

Is there or will there be an onSave or beforeSave event as in VSTO, allowing process to be implemented before a document is save? Same for onClose or beforeClose.

It's something we're thinking about. Currently this is a challenge as our model for Office.js runs code asynchronously. This gives users greater control, but makes it difficult to catch events before they happen. So, we're thinking of different ways to unblock these scenarios.

When are you likely to get feature parity between VBA and Office.js APIs?

This is both a great aspirational and inspirational question. Strictly speaking, we might never reach parity, because the Web Add-in Model is intended to be used across the platforms where Office is supported, and in VBA there are hundreds of members that are not multiplatform compatible (e.g., Application level stuff, multi window hosts, COM interoperability, among many others). So, from that perspective its possible they will never be identical. On the other hand, the team is doing a very bold investment to enable as much as possible the most common development scenarios. In Excel, this is close to 70% of the functional capabilities of VBA (that can be multiplatform). That is literally thousands of APIs that have been added in the last years. It's our aspiration to be growing the API set as needed and for each application using this approach. Right now, the focus is on Excel and Outlook.

Do you have any plans to extend the Word API so that you can make pages landscape/portrait or programmatically change the margins? It seems you cannot go this even using OOXML. This can be done in the COM API and is blocking our migration to Office.js.

This is a great question. Effectively, this is a gap in the Word.js API, but I have seen many customers implementing it is using OOXML, so it should not be a blocker in your case. Margins, and ultimately orientation of pages are defined within sections of a document, and sections are defined on the last paragraph of it. So, the basic strategy is to insert an OOXML that defines a section with the desired margins, and then the content that is expected to be accommodated in the boundaries of the defined section (in the right margins). This requires the insertion of at least 2 OOXMLs one with the margin definitions, and another one with the actual content.

Here is how the margins OOXML needs to be defined and the JS code to make the magic. Note that if you have more sections (i.e., a document with different margin definitions) you will have to repeat as needed

1. The OOXML the defines the margins should look like the one below; I will call this one margins.xml. Please note the trick here is that there are 2 paragraphs, the idea is that after insertion we will delete the first paragraph so that it keeps the last paragraph includes the section definition including margins.

```
<?mso-application progid="Word.Document"?>
```

```
<w:wordDocument xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml" w:macrosPresent="no" w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
```

```
  <w:body>
```

```
    <w:p>
```

```
<w:pPr>
  <w:sectPr>
    <w:pgSz w:w="12240" w:h="15840"/>
    <w:pgMar w:top="1440" w:right="1440" w:bottom="1440" w:left="1440" w:header="720" w:footer="720" w:gutter="0"/>
    <w:cols w:space="720"/>
    <w:docGrid w:line-pitch="360"/>
  </w:sectPr>
</w:pPr>
<w:r>
  <w:t xml:space="preserve"></w:t>
</w:r>
</w:p>
<w:p/>
<w:p/>
<w:p>
  <w:pPr>
    <w:sectPr>
      <w:type w:val="continuous"/>
      <w:pgSz w:w="12240" w:h="20160"/>
      <w:pgMar w:top="1418" w:right="1701" w:bottom="1418" w:left="3402" w:header="709" w:footer="709" w:gutter="0"/>
      <w:cols w:space="708"/>
      <w:docGrid w:line-pitch="360"/>
    </w:sectPr>
  </w:pPr>
</w:p>
<w:sectPr>
  <w:type w:val="continuous"/>

```



```

return context.sync()
.then(function () {
    var last = myPars.items.length;
    myPars.items[0].delete();
    return context.sync()

        .then(function () {
            // the marings OOXML includes a
            carriage return indicating the exact position where we need to
            //insert the document.xml, via
            search we find this position.

            var results =
context.document.body.search(String.fromCharCode(13, 12, 13));
context.load(results,
{expand:'font'});

            return context.sync()
                .then(function () {
                    results.items[0].delete();
                    return context.sync()
                        .then(function () {
                            //time to insert the
                            Office.context.document.
setSelectedDataAsync(ooXmlDocument, { coercionType: 'ooxml' }, function
(result) {
                    if (result.status ==
                        "succeeded") {
                            //if you have
                            Word.run(functionio
                                n (context) {
                                    var
                                    context.load
                                    return
                                    .then(funcuti
                                        on () {
                                            mySectio
ns.items[0].getHeader("primary").insertOoxml(ooXmlHeader, "replace");
                                            return
                                            context.sync()
                                                .then(funcuti
                                                    on () {
                                                        console
                                                        .log("Success!!");
                                                    })
                                                })
                                                    .catch(functionio
                                                        n (e) {
                                                            console.log(
                                                                e.message);
                                                        })
                                                    })
                                                })
                                                    });
                });
            });
        });
    });
};

```

```
        })
    })
})

})

})

    .catch(function (e) {

        console.log(e.message);

    })

}

})

})

})

    .catch(function (e) {

        console.log(e.message);

    })

}
```

Resources

From August's call

- [Office Developer Satisfaction survey](#)
- [SSO Quick Start](#)
- [Jedox's Excel Add-in](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, September 9, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please connect with us by completing this short [survey form](#).

Office Add-ins community call—July 8, 2020

July's call, hosted by **Alex Jerabek**, featured the following presenters and topics:

- **Artur Tarasenko**, Software Engineer, presented a new developer tool: the Web View Switcher for Office Add-ins.
- **Sudhi Ramamurthy**, Program Manager, shared the API improvements that have arrived in Office Scripts.
- **Michelle Scharlock**, Dev Writer, fielded a survey about the programming languages and frameworks our developers use.

Watch the call [here](#).

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

When are you planning on switching the browser engine to the Chromium version of Edge for Office Add-ins?

Office add-ins are already using the Chromium based Edge Webview2 control in the Insiders [Fast] channel. We are planning to GA this integration later this Fall. The exact monthly fork which will make this available to all users is still TBD. Please continue to dial into our monthly community call for the most up-to-date info.

Why does the Shared Runtime only support IE 11 and not Edge? That is based on the [Browsers used by Office Add-ins](#) page. Is this going to change to Edge as well?

This is a platform limitation in the current release of Shared Runtime. We are planning to support Edge in a future update.

When will there be a newer version of the PowerPoint JavaScript API?

We are actively working on new PowerPoint APIs, but do not have a release date for the next PowerPoint API requirement set at this time.

Can I disable mail editing mode without displaying a dialog using `Office.context.ui.displayDialogAsync()` after the `ItemSend` event is triggered by the `On-Send` feature?

No, this is not possible to do today.

Can an Office script be called from an Office Add-in? Or could it work the other way round, calling functions of an add-in from the script?

Neither is possible today, but those are interesting feature ideas. You can call a script through Power Automate. We're exploring the ability to record a broader set of actions for scripts.

Are Office Scripts part of the Workbook object, separate, both?

Your scripts are stored on your OneDrive. You can access your own scripts with any workbook in Excel on the web. You can also [share a script](#) with a workbook or others to use.

Is there any UI or Form element for Office Scripts?

No. If you need a UI, that would be the domain of Office Add-ins

Is it possible to call Rest API like Microsoft Graph API from Office Add In scripts?

[We are working on proper, secure support for external REST calls.](#)

Can Excel be setup to automatically start an add-in for any workbook (even if that workbook has never opened that add-in before) based on user group?

You can deploy an add-in with an autoopen taskpane feature to a particular user group. There are 3 topics I'd recommend checking out: (<https://docs.microsoft.com/en-us/office/dev/add-ins/develop/automatically-open-a-task-pane-with-a-document> OR <https://docs.microsoft.com/en-us/office/dev/add-ins/excel/run-code-on-document-open>) AND (<https://docs.microsoft.com/en-us/office/dev/add-ins/publish/centralized-deployment>).

Can multiple manifests be deployed dependent on user group with one code location?

Yup, that is possible today. That said, if you're looking to build a contextual ribbon based on the user who opens the add-in, we will have contextual Ribbon APIs in development which we'd recommend as a model to follow (so you don't have to manage multiple manifests).

Can multiple add-ins be deployed to one site in sub sites? e.g., addin1.site.com, addin2.site.com, site.com/addin1, site.com/addin2

Yes! Each tenant will have its own centralized deployment instance and within each instance you can target specific users. all of those could be different manifests.

Resources

From July's call

- [Community Call Developer Survey](#)
- [Office Add-in WebView Switcher Tool](#)
- [Office Scripts: Announcing a simplified API, Power Automate support, and sharing](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, August 12, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please connect with us by completing this short [survey form](#).

Office Add-ins community call–June 10, 2020

June’s call, hosted by **Alex Jerabek**, featured the following presenters and topics:

- **Courtney Owen**, Software Engineer, presented two new developer tools: the VS Code debugger extension for Office Add-ins and Script Lab for Outlook.
- **Sudheer Maremanda**, Program Manager, discussed the new Chromium-based Edge WebView2 controller, coming to Office.
- **Daniel Wilson**, Program Manager, demoed the new integrations between Power Automate and Office Scripts.

Watch the call [here](#).

Q&A

We welcome you to submit questions and topic suggestions in advance of each call by using our short [survey form](#).

Can the messageChild API (for Dialogs) be used in older Office versions?

KP: No, the messageChild API won’t be able to be used in older versions. For those you will have to depend on message passing through storage to to communicate from the add-in to the dialog.

Is s/mime encryption in your roadmap for Outlook office web add-ins?

Not in the short term. Please use [UserVoice](#) to request or upvote other events.

Is there better way to trigger a new meeting than the OnSend event?

The OnSend event is not triggering a new meeting, but it’s more an event that gets triggered when a user clicks “send” to an email or an appointment. You can use Microsoft Graph to create a new meeting, but that will slow the sending and you might get timed-out

How can I trigger a send event only in appointments, not in other mail items?

This is not possible today. We are working on a new feature that will allow such filter.

For my Word (2016) Task Pane Add-in, how do I add a keypress event handler to the Word body object so I get access to every character, one at a time, that is typed in the main part of the document?

In Word programmability history, there has never been a keypress event due to the fact that it would significantly degrade the main experience of typing a document in Word. There is also no API in Word.js to get character by character. There are many APIs for range that can give you similar capabilities. You can also try the Search API with wildcards and use a search string like “{9}” that will give you, for instance, the next 9 characters in the range. Please note that you need to be careful when using this API since you might not get what you expect depending on the content (ie: if it includes hyperlinks, tables, footnotes or other invisible characters).

Is it possible to start an add-in panel automatically when any Excel workbook is opened?

Yes! You can review the document in our docs page [here](#).

Can we control the position in the ribbon of the add-in tab?

Unfortunately, no. Currently we do not allow changing the order of tabs. Please request this feature on [UserVoice](#) to help prioritize it.

Can you make the ribbon dynamic?

The capabilities we’re enabling include enabling/disabling ribbon buttons (preview) and allowing an add-in to add contextual tabs (preview). We’re also investigating moving the default position of the tab right now, although this is in the early stages. If this is enabled, this will be a static placement that is defined, there will be no API to programmatically move this.

We don’t allow dynamic show/hide of add-in buttons, dynamic movement of tabs and dynamic changes to icons/labels of add-in buttons

Is SSO with SAML2?

SSO in Office Add-ins is not SAML2. You can find additional information on how to enable SSO for an add-in here in our [documentation](#) which also details some of the flows used.

Are there application-level (Excel, Word, Outlook) events, such as onWorkbookOpen, beforeWorkbookClose?

Events based on open and close behavior are not yet in the product. Outlook has some events related to event-based activation, as described [here](#). There is a complete list of Excel events [here](#). Please head to [UserVoice](#) to request or upvote other events.

Is there an ETA on Office Add-ins debugger extensions for Mac Office apps?

Not currently. The current debugging solutions are Windows-centric.

Will the Office Add-ins VS Code debugging extension support debugging projects built with bundlers like Webpack for React/Vue based apps?

Yes.

Are there any plans on making the Office Add-ins debugger available in other IDE's?

Not currently, but additional platform requests can be made on [UserVoice](#).

This VS Code Add-In is awesome. Was waiting for it long time. You recommend using a runtime lifetime long for Excel Add-Ins. This results in Office on Windows using IE. When will this change and when can I debug a runtime with lifetime long in VS Code?

For the long running/shared runtime, it does currently depend on the IE runtime so you won't be able to use the VS Code debugger quite yet, but we are focused on the Chromium version and providing VS Code debugging support.

Is it possible to get the version of the manifest through Office.js?

We don't have an API for getting manifest version, but it's an interesting idea. Please request this feature on [UserVoice](#).

Is there a way to evaluate a formula in a cell without triggering a worksheet recalculation?

Not yet, we do have adding an office.js API for this on the backlog.

What are the current plans for additional PowerPoint and Word APIs?

We're starting to investigate PowerPoint APIs and capabilities need it and will have some information to share in a few months. We aren't working on any new Word APIs in the short-term. In either case, please head to [UserVoice](#) and create or upvote feature requests to help us prioritize API development.

Is the Script Lab problem with Firefox at all related to my experience with it taking an exceedingly long time to load a taskpane in Excel from a dev-server under Mac OS X?

We have seen many issues while using Firefox and script lab that we are investigating. While we look and fix, we recommend you to use Edge or Chrome. If you see issues, please submit it to [our GitHub repo](#).

Is it possible to start an add-in panel automatically when any Excel workbook is opened?

Yes! [Here's the documentation](#). I'd also recommend combing that with a [long running runtime](#), so your add-in can continue to respond to events when the taskpane is closed.

Can power automate make an external call to an external web service?

Yes! Check out the HTTP Request connector.

Is there an integration point to Power Automate for Office.js Add-in?

There is no default connection between Power Automate and Office Add-ins. However, many add-in providers have Power Automate connectors for their services (such as Trello).

Are power automate connectors monetizable?

Premium connectors are available, but you can review Power Automate's pricing model [here](#). The likely approach may be to charge for the underlying API.

Resources

From June's call

- [Script Lab for Outlook](#)
- [Microsoft Office Add-in Debugger Extension for Visual Studio Code](#)
- [Edge developer channel](#)

Office Add-ins community call

- [Recurring, monthly community call calendar invite](#)
- [Community call topic requests and questions](#)
- Community call recordings on the [YouTube Office Dev channel](#)

Office Add-ins feedback

- Technical questions – [Stack Overflow](#)
- Issues – [GitHub](#)
- Recommendations and suggestions – [Office Developer Platform UserVoice](#)

The next Office Add-ins community call is on **Wednesday, July 8, 2020** at 8:00AM PDT. You can download the calendar invite at <https://aka.ms/officeaddinscommunitycall>.

Engage with us

Are you interested in delivering a presentation during an upcoming Office Add-ins community call? Or would you be willing to provide feedback aimed at helping to improve the [Office Add-ins documentation](#)? If you answered yes to either (or both!) of these questions, please connect with us by completing this short [survey form](#).