

Microsoft Office Add-ins

Community call

- Open in Excel pattern
- Use Linq-to-XML in Office Add-ins
- Community engagement update
- Outlook API update on REST decommission and Mailbox 1.12 Scope



August 10, 2022

Agenda

- Open in Excel pattern
- Use Linq-to-XML in Office Add-ins
- Community engagement update
- Outlook API update



Yawei Zhu

Senior Product Manager



Thomas Barnekow

Senior Principal Architect



Linda Cannon

Senior Content Product Manager



Juan Balmori

Principal Product Manager

Excel add-in pattern and APIs



Yawei Zhu: Senior Product Manager

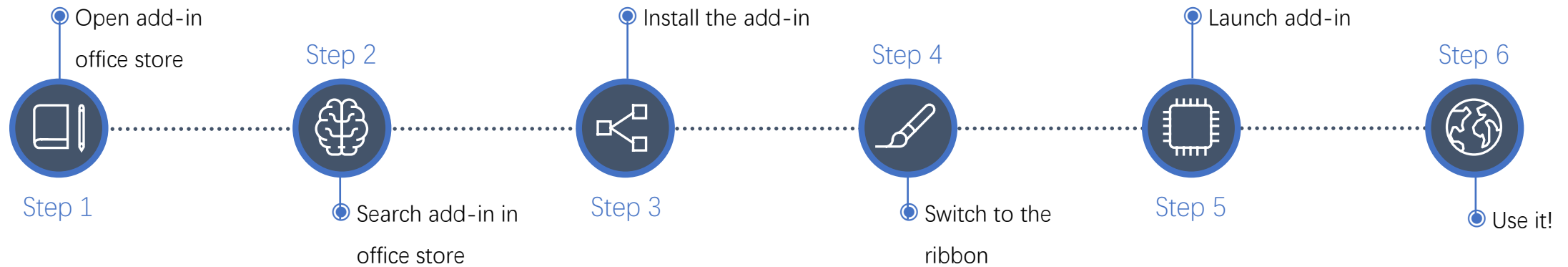


Proposal of a User Pattern Open in Excel

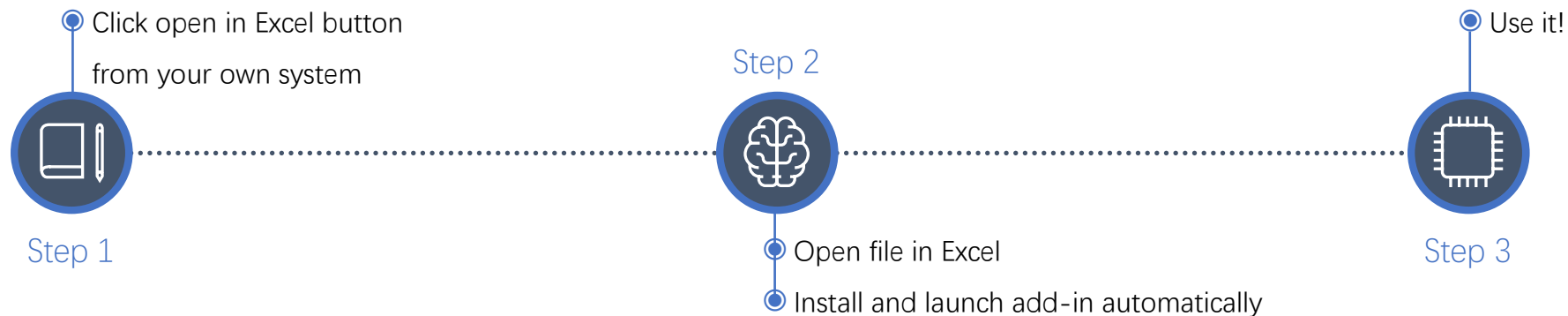
Excel add-in PM: Yawei Zhu

User Journey

Traditional Way



Open in Excel Pattern



User Pattern – Open in Excel Desktop Client

The screenshot displays the Jira Software web interface. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. A search bar is located on the right. The left sidebar shows a list of filters, with 'Done issues' selected. The main content area shows the 'Done issues' filter results, including a table with one issue: 'Kickoff' (HEN-1). The issue details for 'Kickoff' are displayed on the right, showing the status as 'Done' and the assignee as 'Unassigned'. The 'Excel' icon in the top right of the issue details is highlighted with a red box, indicating the user pattern to open the issue in the Excel Desktop Client.

Filters

Search issues

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

View all filters

Done issues Save as

Project: All Type: All Status: All Assignee: All + More Contains text Search Switch to JQL

Status Category: Done

Order by Updated

HEN-1 Kickoff

Projects / Henson项目 / Add 长篇故事 / HEN-1

Kickoff

Attach Add a child issue Link issue

Description

Add a description...

Activity

Show: All Comments History Newest first

Add a comment...

Pro tip: press M to comment

Done Done

Pinned fields

Click on the next to a field label to start pinning.

Details

Assignee

Unassigned

Assign to me

Labels

None

Reporter

朱亚威

Created yesterday

Updated 33 seconds ago

Resolved 33 seconds ago

Configure

28°C Mostly cloudy

2:04 PM 7/6/2022

User Pattern – Open in Excel Online

The screenshot displays the Jira Software web interface. At the top, the navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. A search bar is located on the right. The left sidebar shows a list of filters, with 'Done issues' selected. The main content area is titled 'Done issues' and includes a 'Save as' button. Below this, there are filters for 'Project: All', 'Type: All', 'Status: All', 'Assignee: All', and a '+ More' button. A 'Search' button is also present. The 'Status Category: Done' is selected. The 'Order by Updated' dropdown is set to 'Updated'. A list of issues is shown, with 'HEN-1 Kickoff' selected. The issue details for 'Kickoff' are displayed, including the 'Description' field, 'Activity' section, and 'Pinned fields' section. The 'Open in Excel Online' button is highlighted with a red box. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 2:09 PM on 7/6/2022.

Done issues Save as

Project: All Type: All Status: All Assignee: All + More Search Switch to JQL

Status Category: Done

Order by Updated

☒ HEN-1 Kickoff

Projects / Henson项目 / Add 长篇故事 / HEN-1

Kickoff

Attach Add a child issue Link issue

Description

Add a description...

Activity

Show: All Comments History Newest first

Add a comment...

Pro tip: press M to comment

Done Done

Pinned fields

Click on the next to a field label to start pinning.

Details

Assignee

Unassigned

Assign to me

Labels

None

Reporter

朱亚威

Created yesterday

Updated 33 seconds ago

Resolved 33 seconds ago

Configure

Benefit & Value

This pattern will bring clear value to both user and developer

- To user
 - Better experience
 - Smooth workflow, no need to install add-in from store first
- To add-in developer
 - Better funnel
 - Add-in can be launched automatically, no user friction
 - Better link
 - Stay in link with users when they jump out to Excel
 - Better user awareness
 - Sharing the file will let other related users launch the add-in automatically as well
 - **Better usage**

When to Use It

- User can export data from your system to Excel
 - Any system: ERP, CRM, Finance, BI, EDU, etc.
 - Export data from your system to Excel is a common activity of user
- Your Excel JS add-in integrate your service with Excel
 - Use your add-in to stay in connection with your system data
 - More actions
 - Data analysis & visualization
 - Data write back
 - Etc.

How to Use it

PnP Sample Code for Easy Start

- [Open Excel from your web page and embed your Office Add-in - Office Add-ins | Microsoft Docs](#)

When User's Store is Disabled

- If the store is disabled by the tenant admin, developer can reach the admin to central deploy the add-in with Microsoft admin center to their end user first. Then, when users open Excel from your system/product, they will be able to use the add-in directly.
- [Deploy add-ins in the admin center - Microsoft 365 admin | Microsoft Docs](#)

Running on Background

- Configure shared runtime to let add-in can be still running when user closes the side panel if you need to keep the connection between Excel and your system consistent.
- [Configure your Office Add-in to use a shared JavaScript runtime - Office Add-ins | Microsoft Docs](#)



Thank You!

yaweizhu@microsoft.com

Feedback Forms: <https://forms.office.com/r/14KL0MjPfJ>



OOXML libraries demo



Thomas Barnekow: Senior Principal Architect

Using Linq to XML in Office.js Add-ins

Office Add-in Community Meeting
August 10, 2022

Dr. Thomas Barnekow

thomas.barnekow@dokumate.com

<https://dokumate.com>

Your Speaker: Thomas Barnekow

Not Quite the Typical Software Engineer

- **Analog Devices** (since Aug 2022): Senior Principal Architect
- **Dr. Thomas Barnekow** (since May 2017): Developer of DokuMate (see <https://dokumate.com>)
- 22 years in management consulting and outsourcing sales
 - **ISG**: Senior sourcing advisor and thought leader, EMEA sourcing capability lead, advised and led several multi-billion USD outsourcing deals
 - **IBM**: Outsourcing deal maker for large IT infra deals
 - **Accenture**: Outsourcing “market maker” for large application deals, IT management consulting
 - **McKinsey**: IT management consulting
- **Fraunhofer**: Member of Software Technology Competence Center and Ph.D. thesis

Open-Source Developer

- Significant open-source contributions in the Open XML space: Open XML SDK and Open XML PowerTools
- Developer of TypeScript libraries for dealing with Open XML markup (see <https://github.com/OpenXmlDev>)
 - @openxmldev/linq-to-xml – Think LINQ to XML in .NET
 - @openxmldev/linq-to-ooxml – Namespace-related classes
 - linq-add-in – Demo Office.js add-in

3. → **SERVICE DELIVERY CONTRACTS; PROJECT** CONTRACTS; SERVICES

3.1 → ~~Service-Delivery-Contracts-and-Project-~~ **Contracts.** Details with respect to the ~~S~~_s services ~~(as defined below)~~, resources, and ~~D~~_d deliverables ~~(as defined below)~~ to be provided by Provider hereunder, ~~including with respect to managed services and Projects,~~ will be described in written ~~Service-Delivery-Contracts-or-Project-~~ **Contracts** to this Master ~~Digital~~ **Digital** Services Agreement, each of which will be in substantially the forms attached hereto, respectively, as Attachment H ~~1 (Form of Service-Delivery-Contract) and Attachment H.2~~ (Form of ~~Project~~ **Contract**) (each, as executed and agreed to by Lead Provider (or an Affiliate of Lead Provider (Lead Provider and/or any such Affiliate which is a party to any such ~~Service-Delivery-Contract-or-Project~~ **Contract**, **"Provider"**)) and Customer and as may be amended or supplemented, including pursuant to a Contract Change Request ~~referenced herein,~~ respectively, as a **"Service-Delivery Contract"** or an **"SDC,"** and as a **"Project Contract"** or **"PC,"** and each Service-Delivery Contract and Project Contract is interchangeably referenced herein as a ~~(~~ **"Contract"**). Each of Provider (or any Affiliate of Provider) and Customer will negotiate

in· good· faith· in· accordance· with· this· Section· 3.1· and·
Section· 6.1· (New-Services)· any new-Contract that·
is requested by Customer. Each Contract will, at a minimum, include a reference to this Master
Digital Services Agreement that, except as otherwise expressly set forth in a Contract, will·
be deemed to incorporate the provisions of this Master Digital Services Agreement.·
The Parties acknowledge that no Services will be performed by Provider pursuant to a
Contract, and that Customer will not be responsible for the payment of any fees in connection with
such Contract, until such Contract has been executed by an Authorized Representative of the
party to such Contract.

3. → ~~SERVICE DELIVERY CONTRACTS; PROJECT CONTRACTS; SERVICES~~

3.1 → ~~Service-Delivery-Contracts-and-Project-Contracts.~~ Details with respect to the ~~Services~~ (as defined below), resources, and ~~Deliverables~~ (as defined below) to be provided by Provider hereunder, including with respect to managed services and Projects, will be described in written ~~Service-Delivery-Contracts-or-Project~~ Contracts to this Master Digital Services Agreement, each of which will be in substantially the forms attached hereto, respectively, as Attachment^{H.1} ~~(Form of Service-Delivery-Contract)~~ and Attachment^{H.2} ~~(Form of Project-Contract)~~ (each, as executed and agreed to by Lead Provider (or an Affiliate of Lead Provider (Lead Provider and/or any such Affiliate which is a party to any such Service-Delivery-Contract-or-Project Contract, “**Provider**”)) and Customer and as may be amended or supplemented, including pursuant to a Contract Change Request referenced herein, respectively, as a “~~Service-Delivery-Contract~~” or an “~~SDC,~~” and as a “~~Project-Contract~~” or “~~PC,~~” and each Service-Delivery-Contract and Project-Contract is interchangeably referenced herein as a “~~Contract~~”). Each of Provider (or any Affiliate of Provider) and Customer will negotiate in good faith in accordance with this Section^[3.1] and Section^[6.1] ~~(New Services)~~ any new Contract that is requested by Customer. Each Contract will, at a minimum, include a reference to this Master Digital Services Agreement that, except as otherwise expressly set forth in a Contract, will be deemed to incorporate the provisions of this Master Digital Services Agreement. ~~The Parties acknowledge that no Services will be performed by Provider pursuant to a Contract, and that Customer will not be responsible for the payment of any fees with respect to such Contract, until such Contract has been executed by an Authorized Representative of each party to such Contract.~~ ¶

Transformed version
without w:sdt elements

Original version with
w: **sdt** elements highlighted in Design Mode

Open XML Markup with Nested w:sdt Elements

```
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:sdt>
      <w:sdtContent>
        <w:p>
          <w:sdt>
            <w:sdtContent>
              <w:sdt>
                <w:sdtContent>
                  <w:ins w:id="82" w:author="Author One" w:date="2022-02-25T09:09:00Z">
                    <w:del w:id="83" w:author="Author Two" w:date="2022-02-25T12:29:00Z">
                      <w:r w:rsidR="00324C55">
                        <w:delText>]</w:delText>
                      </w:r>
                    </w:del>
                  </w:ins>
                </w:sdtContent>
              </w:sdt>
            </w:sdtContent>
          </w:sdt>
        </w:p>
      </w:sdtContent>
    </w:sdt>
  </w:body>
</w:document>
```

Block-level w:sdt for sample paragraph consists of **816** lines of Open XML markup

This excerpt represents a “]” that was first inserted and then deleted

Transformed Open XML Markup

```
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:ins w:id="82" w:author="Author One" w:date="2022-02-25T09:09:00Z">
        <w:del w:id="83" w:author="Author Two" w:date="2022-02-25T12:29:00Z">
          <w:r w:rsidR="00324C55">
            <w:delText>]</w:delText>
          </w:r>
        </w:del>
      </w:ins>
    </w:p>
  </w:body>
</w:document>
```

Transformation with XSLT Stylesheet

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="..." xmlns:w="...">

  <xsl:output method="xml" indent="yes"/>
```

```
<!-- Recursively remove all content controls (unwrapping content) -->
<xsl:template match="w:sdt">
  <xsl:apply-templates select="w:sdtContent/node()" />
</xsl:template>
```

```
<!-- Perform identity transformation on everything else -->
<xsl:template match="@* | node()">
  <xsl:copy>
    <xsl:apply-templates select="@* | node()" />
  </xsl:copy>
</xsl:template>
</xsl:stylesheet>
```

Leverage `WordprocessingDocument.ToFlatOpcDocument()` etc. and
`System.Xml.Xsl.XslCompiledTransform` class

Transformation with LINQ to XML

```
private static object RemoveSdtsTransform(XNode node)
{
    return node switch
    {
        XElement element when element.Name == W.sdt =>
            element.Element(W.sdtContent)?
                .Nodes()
                .Select(RemoveSdtsTransform),
        XElement element =>
            new XElement(element.Name,
                element.Attributes(),
                element.Nodes()
                    .Select(RemoveSdtsTransform)),
        _ => node
    };
}
```

Leverage `DocumentFormat.OpenXml.Linq` (`XName` properties of `W` class) and
`System.Xml.Linq` classes such as `XNode` and `XElement`

With the `@openxmldev/linq-to-xml` and `@openxmldev/linq-to-ooxml` packages, you can now use the same approach in TypeScript and Office.js add-ins

- **@openxmldev/linq-to-xml** package implements LINQ to XML for TypeScript
 - See <https://github.com/OpenXmlDev/linq-to-xml>
 - See <https://www.npmjs.com/package/@openxmldev/linq-to-xml>
- **@openxmldev/linq-to-ooxml** package provides namespace-related classes (e.g., `W`, `W14`)
 - See <https://github.com/OpenXmlDev/linq-to-ooxml>
 - See <https://www.npmjs.com/package/@openxmldev/linq-to-ooxml>
- **linq-to-xml** sample add-in shows how to use the above libraries: <https://github.com/OpenXmlDev/linq-add-in>

When to Use

- You have previous experience with Open XML or even C# code you could port to TypeScript
- The Office.js API does not provide adequate support for your use case

DEMO

Thank You

Q&A

Community engagement update



Linda Cannon: Senior Content Product Manager

Challenge: many channels and types

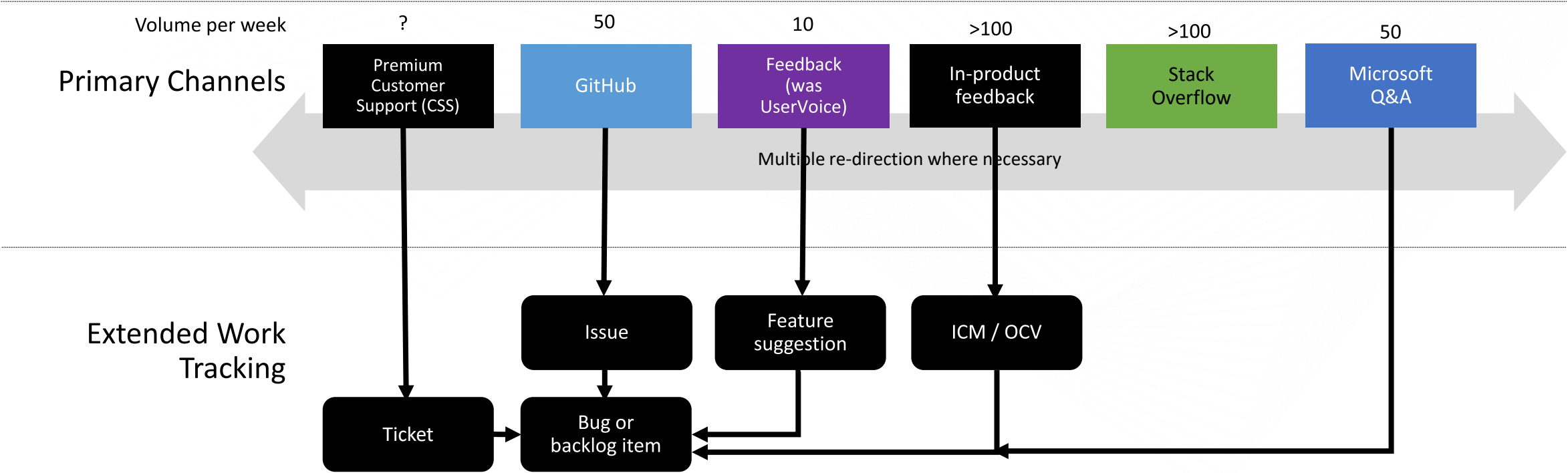
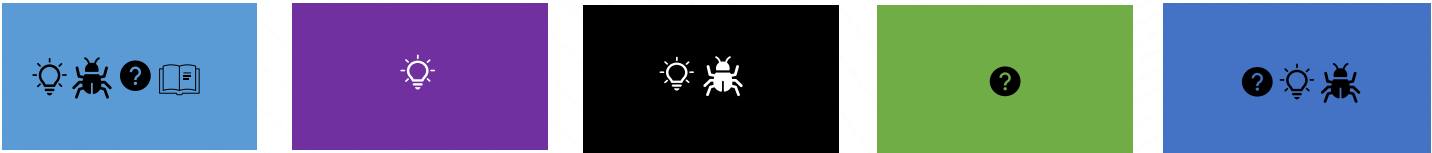
Note this does not include legitimate service issues or outages, nor does it cover paid services for Premium support. This covers free support developers can find across our many channels. The types are:

- Product **ideas** or feature requests
- Product **bugs**
- Troubleshooting or help with code (**how-to**) requests
- Requests for **docs** or samples

Mix of Contact Types

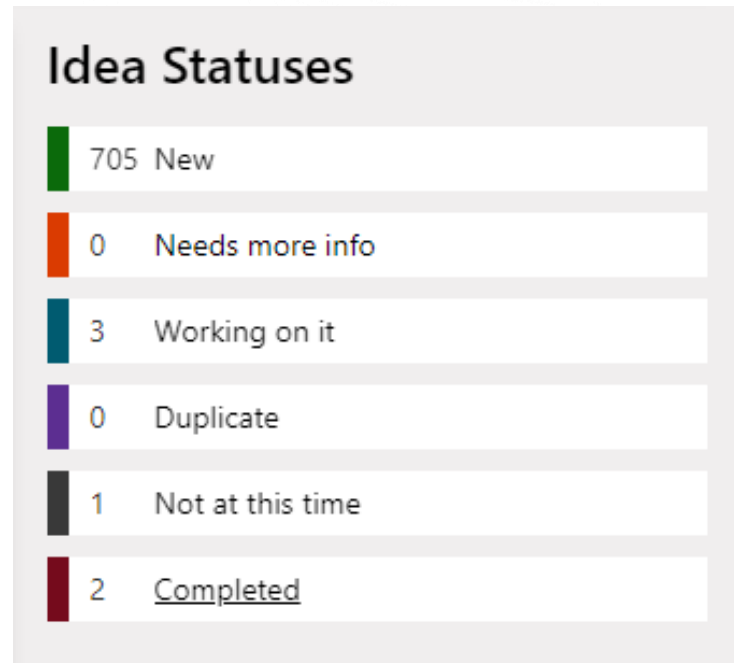
idea | bug | how-to | docs

5% | 45% | 45% | 5%



Problem to solve: What happens to customer suggestions/ideas?

- Since UserVoice retired in 2021, we've asked customers to go to <https://aka.ms/m365dev-suggestions> (on Tech Communities) to submit ideas or vote on them.
- Added to weekly Feedback Process Coordinator process to triage.
- This was a temporary solution until the [Feedback Portal](#) shipped.
- We currently have 650+ new ideas. Most of these have not been triaged, however many have multiple upvotes.
- We also have 16 open issues in GitHub office-js labeled [product feature request](#).



What is the problem to be solved?

Send customers to new Feedback portal or keep in GitHub?

	Feedback portal	GitHub office-js
Pros	Official One Microsoft solution Can upvote Searches for possible dupes	Homegrown solution Can customize Can upvote Discussions Customers prefer this because it is triaged daily Can attach logs, screenshots, etc.
Cons	<ul style="list-style-type: none">• Product centric, not platform• No forum for Office or office-js• “Feedback” is a big bucket includes all contact types (bugs, ideas, how-tos, Docs)• Another place to triage where we’ll be unresponsive due to resource constraints• We only have one OCV bucket (Extensibility) so no way to integrate• No customization, no file attachments	<ul style="list-style-type: none">• Difficult to know who to assign issues• Growing list of issues in backlog• Integrating with ADO for bugs is possible but requires a resource to develop it• Also, it is possible to create PowerBI reports but requires a resource

Your feedback matters

- Where we're at now:
Feature spec has been submitted requesting **we change to GitHub**. Note this decision is not final, the ETA is end of this month.
- Please review the new Feedback Portal vs. GitHub issues and send your thoughts to the channel or email me directly lindalu@microsoft.com with "Office Add-ins feature suggestions" as email subject.
- If you have any suggestions that you submitted to the current forum on Tech Communities, please @lindalu-msft in the comments so I can prioritize triaging your submission!
- If you are interested in participating in a future User Research session about customer support channels, please let me know. Email me lindalu@microsoft with "Office Add-ins Developer User Research" as email subject.

Outlook API update



- Mailbox 1.12 Scope Defined
- REST Decommission Update and FAQ

Juan Balmori: Principal Product Manager

Mailbox 1.12

Scope



Mailbox 1.12 Scope defined

Smart Alerts

- OnMessageSend
- OnAppointmentSend

New Compose events

- OnMessageCompose
- OnAppointmentOrganizer



Outlook Smart Alerts in a nutshell

- Event-based activation feature that allows logic to run after a user clicks “Send” from an Outlook message or appointment.
- Activated by the OnMessageSend and OnAppointmentSend events.
- Sample scenarios include:
 - Verifying that the correct signature is used.
 - Verifying that a sensitivity label is applied.
 - Verifying that the correct color categories are applied.
- Feature is currently in preview on Windows and OWA!
- Will be accepted in the store!

Key parts of the manifest

```
<LaunchEvents>
  <LaunchEvent Type="OnNewMessageCompose" FunctionName="onMessageComposeHandler"/>
  <LaunchEvent Type="OnNewAppointmentOrganizer" FunctionName="onAppointmentComposeHandler"/>
  <LaunchEvent Type="OnMessageSend" FunctionName="onMessageSendHandler" SendMode="SoftBlock"/>
  <LaunchEvent Type="OnAppointmentSend" FunctionName="onAppointmentSendHandler" SendMode="Block"/>
</LaunchEvents>
```

- Uses the **<LaunchEvents>** element in the manifest.
- **Three SendMode options** available.
 - **PromptUser** – use it for recommended changes (provides a “Send Anyway” option)
 - **SoftBlock** – use it for mandatory changes: Does not provide an Option to “Send Anyway” except if add-in is unavailable.
 - **Block** – use it for mandatory changes and will not allow send if add-in unavailable (will not be allowed in store)



Outlook V2 Decommission Update

Some Context: Back in February we shared:

- On Nov 30th, 2022, the Outlook REST V2 Endpoint will be decommissioned.
 - Impacts Exchange Online, and Exchange OnPrem on Hybrid.
- Scenarios impacted
 - Using the REST APIs
 - Delegate Scenarios using REST

```
function createAppointment() {
  Office.context.mailbox.getCallbackTokenAsync({ isRest: true }, function(result) {
    var token = result.value;
    var getMessageUrl = Office.context.mailbox.restUrl + "/v2.0/me/events";

    var xhr = new XMLHttpRequest();
    xhr.open("POST", getMessageUrl);
    xhr.setRequestHeader("Authorization", "Bearer " + token);
    xhr.setRequestHeader("Content-Type", "application/json");

    xhr.onload = function(e) {
      var ewsID = JSON.parse(this.response).Id;
      console.log(ewsID);
      Office.context.mailbox.displayAppointmentForm(ewsID);
    };
    xhr.send(
      '{"Subject":"This is a Test32!","Start":{"DateTime":"2021-04-03T18:30","TimeZone":"UTC"},"End":{"DateTime":"2020-08-12T19:30","TimeZone":"UTC"}}'
    );
  });
}
```


1. Good news: all active Add-ins using REST will be granted exceptions to keep using the service until Office 2019 goes out of support (October 14,2025)
2. We will automatically ID traffic coming from Add-ins.
 - Currently testing will all partners who sent RESTCheck. Looks good!
3. Participate on Validation? Please send <https://aka.ms/RESTCheck>
4. Move to Graph ASAP
 1. PLEASE share any issues you find. (either graph scope or non functional issues like perf.)

When the Outlook REST API is decommissioned will there be any programmatic way to confirm at add-in runtime that the REST API is not usable by the add-in?

There are some scenarios where the add-in could still use the REST API and we want to be able to know those.

e.g. Code that an add-in which has had its ID registered on the Microsoft form, an add-in using the REST API of Exchange 2016 (which does exist and does work, have confirmed this against customer deployments).

For example, what will `Office.context.mailbox.restUrl` return after Outlook REST API is decommissioned?

Solution for Mobile?

1. Our Add-in used the REST API on mobile devices. We have no facility to run server-side code for the add-in to support the Graph SSO flow to achieve the same experience with that API.

Given this, we were wondering if we could use `getCallbackTokenAsync` on mobile to retrieve a token usable with EWS? Can you confirm this is possible? Then we could issue the API requests manually to EWS client-side instead of through `makeEwsRequestAsync` (not available on mobile).

2. Can we get examples of using Outlook add-ins on mobile with an alternative to the REST API? The add-in documentation still refers to using the REST API on mobile - <https://docs.microsoft.com/en-us/office/dev/add-ins/outlook/outlook-mobile-addins>

More mobile questions

Can we get some documentation/guidance on how to update mobile add-ins to work post REST API decommissioning?

getAccessToken for use with Graph isn't available in the requirement set supported by Outlook on iOS or Android.

getCallbackToken will return an EWS token but it only works for GetItem and GetAttachment, so add-ins that were accessing operations other than those e.g. FindFolders, won't work with that approach.

Is there an available migration option here for an add-in that uses operations beyond GetItem and GetAttachment (that doesn't force user popups where previously there were none)?

Using REST in OnPrem?

The Outlook REST API has been available in Exchange 2016 since CU3 (and is also in Exchange 2019). However all talk of decommissioning of the REST API seems to be around O365. Are on-premise Exchange installs also going to be impacted by the decommissioning? Or will the REST API remain available there?

Note: Previously it was stated on a call that Exchange 2016/2019 does not support REST API, I've provided a Microsoft article below stating it is supported from 2016. Also we have customers using Exchange 2016 and the REST API with our add-ins and I've used it myself against our lab Exchange environment

<https://techcommunity.microsoft.com/t5/exchange-team-blog/on-premises-architectural-requirements-for-the-rest-api/ba-p/605609>

Word add-ins survey

Please take a few minutes to fill in the survey, which we'll use as an input for prioritizing future work. Thank you!

aka.ms/wordAPI

Q&A

Question:

Is there a way to stop the mouse cursor icon changing to a loading icon when calling the 'Powerpoint.run()' function in an office add-in?

Answer:

This behavior is by design because Office JS APIs run on PowerPoint's UI thread. The user is unable to interact with PowerPoint until API operations are complete. If an operation takes a bit longer than usual, the cursor change informs the user they are temporarily blocked.

Q&A

Question:

When performing a search using Microsoft Graph, preferably using the '/search(q='\${searchText}' endpoint, is it possible to search SharePoint files by their 'ImageTags'?

Answer:

We recommend you follow up with the Microsoft Graph team at <https://docs.microsoft.com/answers/products/graph>

Q&A

Question:

Is there a way to start a web addin (open the web task pane) from a VSTO add-in? This capability would make the transition from VSTO to web much easier to implement some functionality in a web add-in and have a smooth transition from VSTO to web as if it was a single add-in. Migrating the entire add-in from VSTO to web in a big bang is not possible since the JS api is lacking some COM APIs.

Answer:

No this capability does not exist. There are resources available for this scenario though. Please see <https://docs.microsoft.com/en-us/office/dev/add-ins/develop/make-office-add-in-compatible-with-existing-com-add-in> and <https://docs.microsoft.com/en-us/office/dev/add-ins/tutorials/migrate-vsto-to-office-add-in-shared-code-library-tutorial>

Q&A

Question:

How can we create new prepopulated emails from a web add-in hosted by another Office application? E.g. have a button in a Word add-in to create an email with subject and body already set and the Word document attached (as pdf, maybe with some pdf manipulation in between) as well. Something like this was possible with VSTO add-ins but I can't find a solution with web add-ins.

Answer:

If the goal to attach the currently open document as a PDF, then in that case you can use the **Office.document getAsync** API. See [https://docs.microsoft.com/en-us/javascript/api/office/office.document?view=word-js-preview#office-office-document-getfileasync-member\(2\)](https://docs.microsoft.com/en-us/javascript/api/office/office.document?view=word-js-preview#office-office-document-getfileasync-member(2))

Unfortunately there isn't direct way to send the email from the Word add-in. You can open a dialog box using the [Office dialog API](#). Then use [Microsoft Graph mail API](#) to construct and send the email with the PDF attached.

Q&A

Question:

With VSTO it is possible to create PDF/A versions of Office files. In our target sector (public sector, governments) using PDF/A is important and required in most cases. How can a PDF/A representation of an office file can be created using the JS APIs?

Answer:

Unfortunately Office JS only supports the PDF format right now. This would be a great feature request. Please let us know more at <https://aka.ms/m365dev-suggestions>

Q&A

Question:

[Regarding GitHub issue: Sharing Excel Workbooks dependent on Add-ins can cause irrecoverable corruption](#). This also relates to the Open in Excel functionality. There is no practical workaround. We were advised to submit a request for resolution to the Microsoft 365 Developer Platform. We submitted the "idea" on June 1 but have received no upvotes or responses so far. We were informed "the PM team will consider it at the Quarterly planning." What are the dates of the last and next Quarterly planning meetings? Why do critical breaking issues need to be posted on an "ideas forum"? When can we expect to see a response from Microsoft?

Answer:

We apologize for the time it is taking to investigate this issue. There is some nuance to how an add-in is activated when embedded in a document. We are validating the different scenarios here and will follow up on the GitHub issue this week.

Q&A

Outlook Question:

We have customers asking for more control over the sizing and positioning of our add-in in Outlook. Particularly the new versions of Outlook where the size of the add-in icon has been decreased and it can be pushed into a sub-menu, customers want to be able to make the add-in button more prominent for their users (automatically across an organization). Is there any work being done in this area?

Outlook Question:

Is it possible to create an OWA add-in that will replace/augment the public location suggestions when creating a Calendar Appointment?

Currently it shows results using Bing, and I'd like to create an add-in that would bring in results from a different source.

Answer:

We'll provide an answer in the blog post following the call.

New to Office Add-ins?



Documentation

<https://aka.ms/office-add-ins-docs>



Quick starts

[Outlook](#), [Excel](#), [Word](#), [PowerPoint](#)



Script Lab

<https://aka.ms/getscripmlab>



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>

(Notice. This call is on a summer break between July 1st and 31st of August 2022)

Adaptive Cards (monthly) <https://aka.ms/adaptivecardscommunitycall>

Microsoft identity platform (monthly) <https://aka.ms/IDDevCommunityCalendar>

Office Add-ins (monthly) <https://aka.ms/officeaddinscommunitycall>

Power Platform (monthly) <https://aka.ms/PowerPlatformMonthlyCall>

Microsoft 365 & Power Platform
Development (bi-weekly) <https://aka.ms/m365-dev-sig>

Viva Connections &
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: **September 14th at 8:00 AM Pacific Time**

<https://aka.ms/officeaddinscommunitycall>

Agenda: <https://aka.ms/officeaddinsagenda>

Submit your questions: <https://aka.ms/officeaddinsform>

