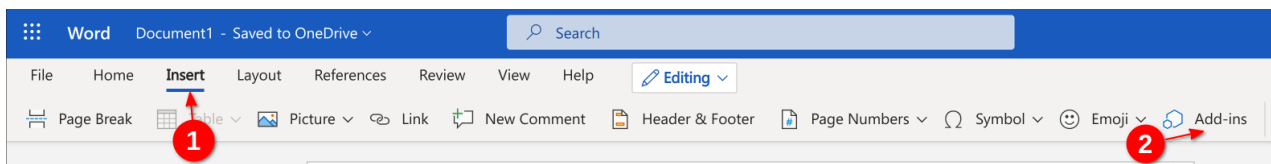


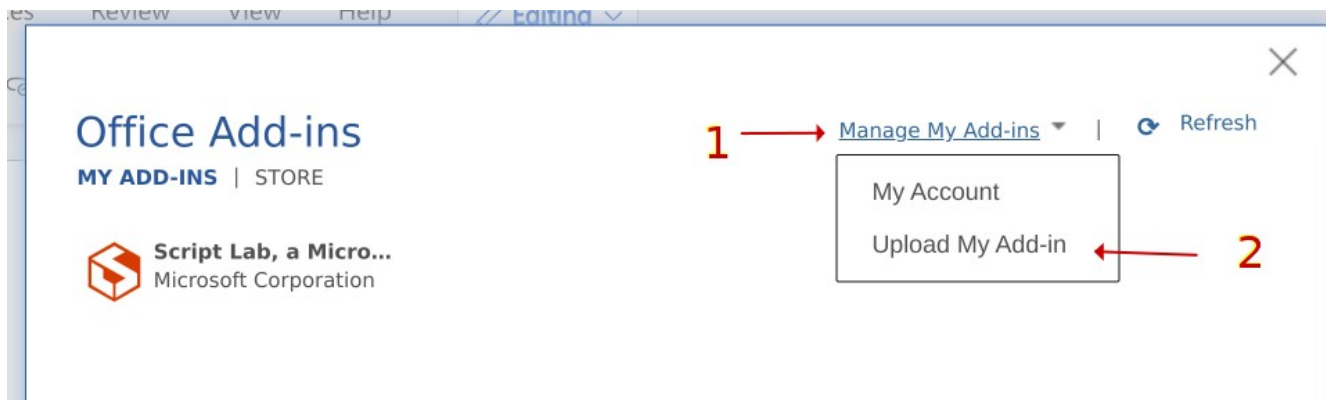


Install DEV version of Word 365 add-in

1. Download manifest from URL <https://wds-studio.com/test/word-addon/manifest.xml>
2. Go to “Insert” tab & click “Add-ins”

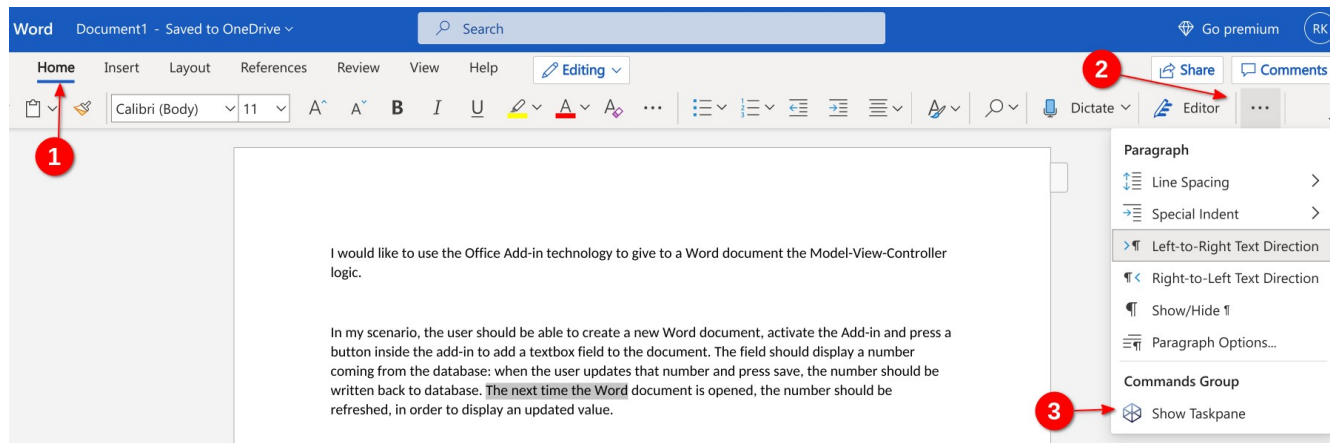


3. Open dialog to choose manifest file



4. Choose downloaded manifest.xml

5. Open Taskpane



6. Select text and click one of buttons

The screenshot shows a Word document on the left and the 'Integra Word Add-in' pane on the right. In the document, a paragraph is selected, indicated by a red circle with the number '1'. The add-in pane on the right contains a dropdown menu set to 'Standard Lease Template' and several buttons: 'Lessee', 'Lessor', 'Lease Start Date', 'Lease Duration', 'Property Name', 'Property Address', 'Lease Notes', and a '+' button. A red circle with the number '2' is placed over the 'Lessor' button, with a red arrow pointing to it from the selected text in the document.

I would like to use the Office Add-in technology to give to a Word document the Model-View-Controller logic.

In my scenario, the user should be able to create a new Word document, activate the Add-in and press a button inside the add-in to add a textbox field to the document. The field should display a number coming from the database: when the user updates that number and press save, the number should be written back to database. The next time the Word document is opened, the number should be refreshed, in order to display an updated value.

What I would like to know is whether the Add-in technology can be used (with binding support maybe?) or it is still in experimental stage. I'm having trouble trying to understand the best way to build a compact OOXML that simply represents my textbox: the method `getSelectedDataAsync` outputs a very complex document, it can't be the smarter way to describe a single-light-basic-poor-naked-textbox field! And how to set the binding to a custom XML? Where should that XML be placed? Inside the document? I looked at the samples provided on GitHub by Microsoft but they aren't so clear, since in many cases they're based on existing Word documents, which already have the necessary data and fields.

7. Done, document's custom props was updated & saved

8. You can download file from OneDrive and check custom props in desktop Word

Remove/reinstall add in

This process is a little bit tricky for dev versions of add-in (thanks to Microsoft).

1. Open Developers tools (screens from chrome)

The screenshot shows the Chrome browser interface. The address bar contains the URL 'CDI123&parId=E50914AECE17F8CD1101&app=Word'. The Chrome menu is open, showing options like 'New tab', 'New window', 'New incognito window', 'History', 'Downloads', 'Bookmarks', 'Zoom', 'Print...', 'Cast...', 'Find...', 'More tools', 'Edit', 'Cut', 'Copy', 'Paste', 'Settings', 'Help', and 'Exit'. A red circle with the number '1' is placed over the 'More tools' option. A red circle with the number '2' is placed over the 'More tools' option. A red circle with the number '3' is placed over the 'Developer tools' option, which is highlighted in blue. A red arrow points from the 'Developer tools' option to the 'Developer tools' option in the bottom left corner of the browser window.

CDI123&parId=E50914AECE17F8CD1101&app=Word

New tab Ctrl+T

New window Ctrl+N

New incognito window Ctrl+Shift+N

History

Downloads Ctrl+J

Bookmarks

Zoom 100%

Print... Ctrl+P

Cast...

Find... Ctrl+F

More tools

Edit Cut Copy Paste

Settings

Help

Exit

Save page as... Ctrl+S

Create shortcut...

Clear browsing data... Ctrl+Shift+Del

Extensions

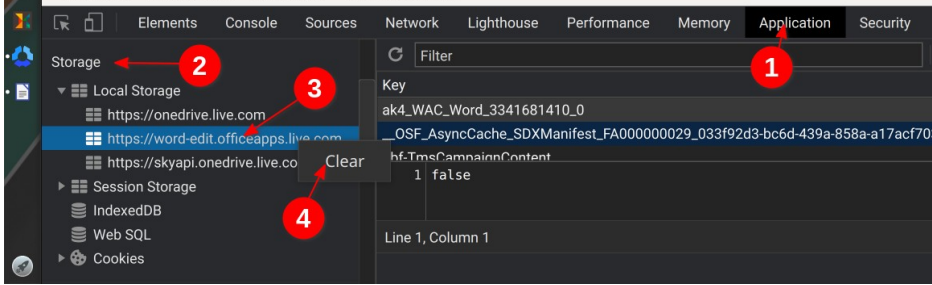
Task manager Shift+Esc

Developer tools Ctrl+Shift+I

2. Goto "Application" tab

3. Scroll left block to "Storage", open "Local Storage"

4. Right click on item <https://word-edit.officeapps.live.com> and choose “Clear” from context menu.



5. RELOAD page

6. Install add-in again