# Word add-ins documentation

**General**: <https://learn.microsoft.com/en-us/office/dev/add-ins/word/>

**Best Practices**: <https://learn.microsoft.com/en-us/office/dev/add-ins/concepts/add-in-development-best-practices>

**Word JavaScript API:**

<https://learn.microsoft.com/en-us/office/dev/add-ins/reference/overview/word-add-ins-reference-overview>

<https://learn.microsoft.com/en-us/javascript/api/word?view=word-js-preview>

# Setup Work Environment

**Tools**:

<https://learn.microsoft.com/en-us/office/dev/add-ins/overview/set-up-your-dev-environment?tabs=yeomangenerator>

* Node.js
* Npm
* Code Editor (VS Code)
* Yo Office
* Office JavaScript Linter
* Script Lab

**VS Code**:

<https://code.visualstudio.com/docs/?dv=win>

**Yeoman Generator**:

<https://learn.microsoft.com/en-us/office/dev/add-ins/quickstarts/word-quickstart?tabs=yeomangenerator>

Ablauf Setup:

1. Node.js
2. Npm
3. Code Editor (Vs Code)
4. (opt) JS Linter

# Project Folder

**./manifest.xml: Definiert die Einstellungen und Fähigkeiten des Add-Ins.**

**./src/taskpane:**

* **/taskpane.html Enthält das HTML-Markup für den Aufgabenbereich**
* **/taskpane.css Enthält das CSS, das auf den Inhalt des Aufgabenfensters angewendet wird**
* **/taskpane.js Enthält den Office JavaScript-API-Code, der die Interaktion zwischen dem**

**Aufgabenbereich und der Office-Clientanwendung erleichtert.**

# Testing the Addin

CMD:

* cd "My Office Add-in"
* npm start

Erklärung:

* npm: Paketmanager für Node.js. Startet node server.js
* <https://docs.npmjs.com/cli/v6/commands/npm-start>
* Änderungen im Code können direkt in Word betrachtet werden

# Word API (Potenziell relevante)