OxOffice Online

整合手冊

- 簡介
- 如何整合
- 逐步建立
- 簡單的範例（來自 Collabora）
- 進階整合
- 個人資訊流程
- Cookies 和使用者本地儲存
- PostMessage API

簡介
這份文件將協助您整合現有的雲端儲存方案，以便您的使用者可以透過 Web 瀏覽器編輯儲存在雲端硬碟中的文件。

因此，您必須將下列部份設定好，並且組合在一起：

OxOffice Online 主機
能開出 iframe 的頁面
使用者驗證
連接雲端硬碟

整合 OxOffice Online 最簡單的方法，就是使用 WOPI 協定。

OxOffice Online 實作了一個名為 WOPI（全名是： Web 應用程式開放平台介面）的協定。 OxOffice Online 是 WOPI 客戶端 ，可以和 WOPI 主機 （也就是您現有的雲端應用和儲存方案）整合在一起。 WOPI
是一個開放協議，更多詳細資料，請造訪 WOPI documentation.

如何整合

OxOffice Online 主機

儘管可以和 Web 伺服器安裝在同一臺主機，不過，我們建議將 OxOffice Online 安裝在專用的 VM 或伺服器。這部伺服器（ WOPI 客戶端），必須能讓外部網際網路連線，而且還要能夠連線到您的 WOPI 伺服器（您的雲端應用系統）。

開出編輯 iframe 頁面的網站

我們假設您希望將編輯功能整合到現有網站中。 因此，在網站上，您需要提供一個 iframe，其中將顯示 OxOffice Online 的編輯介面和文件本身。

要設定 iframe， WOPI 主機（您的雲端應用系統）需要從 WOPI 客戶端（ OxOffice Online 伺服器）上定義的位置讀取 discovery XML。位址如下:

```
https://<WOPI 客戶端 URL>:<port>/hosting/discovery
```

這會得到 discovery.xml，其中包含各種文件格式的 **urlsrc**。 **urlsrc** 為編輯文件用的 iframe 需要使用的位址。

接著您需要為欲編輯的檔案，提供一個名為 **WOPISrc** 的位址， **WOPISrc** 網址看起來如下:

```
https://<WOPI 主機 URL>/<...>/wopi/files/<file id>
```

這裡的 `/wopi/` 實際上可以是任何以 `wopi` 開頭的字串，像是 `/wopifiles/` 或 `/wopi_implementation/` 都可以，但為了簡單起見，我們之後只會使用 `/wopi/`。

`<file id>` 必須以 base64 編碼，也就是說只能有英文 `A-Z` 或 `a-z`、數字 `0-9` 以及 `-` 和 `_` 符號。

使用者驗證

為了能夠安全地存取文件，您的應用程式必須將身份驗證用的 token 傳遞給 OxOffice Online 的 access_token。對 OxOffice Online 而言， token 可以是任何隨機的數字或字串，它會在存取文件過程中（ WOPI 客戶端對 WOPI 主機）當作 URL 的一部份傳遞。

唯一的要求是，對於識別使用者而言， token 必須是唯一的，不能和其他使用者重複，意即，若 WOPI 客戶端對 **WOPISrc** 所指定的 URL 進行存取時時，您必須檢查所傳遞的 token 是否與當初建立 iframe 所傳遞的 token 相符，否則您的應用程序應該拒絕錯誤的 token 存取檔案

目前這是唯一支援的身份驗證方式。

連接雲端硬碟

作為 WOPI 主機，您的應用程式必須為 OxOffice Online （ WOPI 客戶端）實作幾個進入點，以便 OxOffice Online 下載使用者想要編輯的檔案，以及傳回更新檔案。

WOPI 客戶端（ OxOffice Online）會呼叫上面所建立的 **WOPISrc** 來下載檔案:

```
GET https://<WOPI 主機 URL>/<...>/wopi/files/<file id>/contents?access_token=<token>
```

以及傳回更新檔案:

```
POST https://<WOPI 主機 URL>/<...>/wopi/files/<file id>/contents?access_token=<token>
```

目前， OxOffice Online 使用 4 個 WOPI 的檔案操作，分別是: **CheckFileInfo**、**GetFile**、**PutFile** 以及 **PutRelativeFile**。

您的應用程式，必須產生參照檔案及使用者的唯一 token，該 token 可以為固定或過期，只要檔案編輯期間，可供辨識即可。該 token 會透過 URL 的 access_token=<token> 傳遞，您的應用程式也可以在接收的 HTTP header 中，取得 OxOffice Online 傳遞的 token。

```
Authorization: Bearer <token>
```

並且需支援下列 4 個功能:

1. **CheckFileInfo**: 以下 URL 被呼叫時，至少要以 JSON 格式，傳回 BaseFileName 和 Size。

   ```
   GET https://<WOPI 主機 URL>/<...>/wopi/files/<file id>?access_token=<token>
   ```

   OxOffice Online 可用的 CheckFileInfo 屬性列表:

屬性	型別 （ Data type）	必要 Required	說明
BaseFileName	string	yes	不含路徑的檔案名稱，用來顯示在使用者的編輯畫面。
OwnerId	string	yes	識別該檔案擁有者的唯一 ID

Size	number (int64)	yes	以 bytes 為單位的檔案大小，64 位元長整數。
UserId	string	yes	存取該檔案的使用者 ID。
UserFriendlyName	string	no	使用者名稱，用來顯示在使用者的編輯畫面。
UserCanWrite	boolean	no	如果要能夠編輯，必須設為 true，預設為 false。
UserCanNotWriteRelative	boolean	no	禁止使用者重新命名或另存新檔。預設 true。
PostMessageOrigin	string	no	這是告訴 OxOffice Online。PostMessage API 的 WOPI 主機位址。若未指定，將無法接收 OxOffice Online 傳來的狀態。
HidePrintOption	boolean	no	隱藏 OxOffice Online 的列印按鈕。但可透過 PostMessage API 來實作 WOPI 主機自己的列印 UI。
DisablePrint	boolean	no	禁止列印文件。此外，也會在 UI 中隱藏列印選項。
HideSaveOption	boolean	no	隱藏 OxOffice Online 的存檔按鈕。但仍可透過 PostMessage API 觸發，這並不會影響自動存檔功能。
HideExportOption	boolean	no	隱藏 OxOffice Online「下載為 ...」按鈕及選項，但仍可透過 PostMessage API 觸發。
DisableInactiveMessages	boolean	no	當停止編輯文件一段時間後，預設會在文件編輯區顯示提示文字。若禁用此功能，您監聽的 JavaScript，就必須在取得 Session_Closed 或 User_Idle 的 PostMessage API 時，自行向使用者提供適當的訊息。
DisableExport	boolean	no	禁止匯出或下載文件。此外，也會在 UI 中隱藏下載選項。

DisableCopy	boolean	no	禁止複製內容到剪貼簿，但文件內部仍然可以複製以及貼上外部資料。
DownloadAsPostMessage	boolean	no	WOPI 主機自行接管下載任務，例如使用者觸發列印、權螢幕投影簡報以及其他檔案下載功能時，將觸發名為 Download_As 的 PostMessage API，傳送的值中，包含以下 JSON: `{ Type: 'print'|'slideshow'|'export', URL: '..際下載的URL...' }`
EnableOwnerTermination	boolean	no	授予檔案擁有者可以終止其他共編者編輯的權利，意即可在共編時，將其他共編者踢出共編行列。
LastModifiedTime	string	no	檔案最後修改時間。使用 ISO8601 格式表示之 UTC 日期時間（ YYYY-MM-DDThh:mm:ss.nnnnZ）。
UserExtraInfo	object	no	使用者延伸資訊。請參考: **UserExtraInfo Details** 。

UserExtraInfo Details:

屬性	型別 （ Data type）	必要 Required	說明
avatar	string	no	使用者頭像位址。
mail	string	no	使用者的電子郵件位址。

ip	string	no	使用者的 IP 位址。由於 OxOffice Online 可能透過 reverse proxy 轉址，所以自行抓取的 IP 可能不正確，此種狀況下，需透過您的應用程式，將使用者 IP 置於此屬性。
watermark	object	no	自訂編輯浮水印。請參考 **Watermark Details**。

Watermark Details:

屬性	型別 （Data type）	必要 Required	說明
editing	boolean	no	編輯時啟用，預設為 false。
printing	boolean	no	列印時啟用，預設為 false。
opacity	number(double)	no	不透明度。範圍 0.01-1.00 預設為 0.20。此值愈小愈透明。
familyname	string	no	字型名稱。
angle	number(int)	no	旋轉角度。範圍 0-360，預設為 45°
color	string	no	CSS #rrggbb 表示之顏色。預設 #000000
bold	boolean	no	粗體。預設為 false。
italic	boolean	no	斜體。預設為 false。
outline	boolean	no	空心字。預設為 false。
shadow	boolean	no	陰影字。預設為 false。
text	string	no	浮水印文字，若需多行，各行需以 \n 分隔。

2. **GetFile**: 取得欲編輯的檔案 。該 URL 被呼叫時，您的應用程式必須回傳檔案，給 WOPI 客戶端（OxOffice Online）。

```
GET https://<WOPI 主機 URL>/<...>/wopi/files/<id>/contents?access_token=<token>
```

3. **PutFile**: 更新正在編輯的檔案 。該 URL 被呼叫時，您的應用程式必須接收 WOPI 客戶端（ OxOffice Online）傳來的檔案。

```
POST https://<WOPI host URL>/<...>/wopi/files/<id>/contents?access_token=<token>
```

WOPI 主機必須檢查 OxOffice Online 傳來的標頭 – `X-WOPI-Override` 是否為英文大寫 **PUT**。

再檢查 – `X-LOOL-WOPI-Timestamp`，這個標頭含有 ISO8601 格式的最近修改時間，您的應用程式需檢查該時間是否與雲端硬碟的最近修改時間一致，若不符合，則 WOPI 主機不應將之儲存到雲端硬碟，須回應給 OxOffice Oline 一個 HTTP 409 的狀態碼，以及一個 OxOffice Online 特定狀態碼:

```
{
□LOOLStatusCode': 1010
}
```

若時間相符，且應用程式成功儲存更新的檔案後，需回應 JSON 內容，格式同 **CheckFileInfo** 的 LastModifiedTime，以便讓 OxOffice Online 與雲端硬碟之間，保持檔案時間同步:

```
{
  "LastModifiedTime": "YYYY-MM-DDThh:mm:ss.nnnnZ"
}
```

> 此外， **PutFile** 時， OxOffice Online 另外會帶入以下幾個標頭:
> `X-LOOL-WOPI-IsModifiedByUser` – 使用者是否在存檔之前修改了文件（ true ），或者他們只是按了存檔（ false）卻沒有進行任何修改。
> `X-LOOL-WOPI-IsAutosave` – 自動存檔（ true）或使用者按了「存檔」（ false）按鈕。
> `X-LOOL-WOPI-IsExitSave` – 當所有編輯者（共編狀態）都結束編輯，或離線時，將觸發自動存檔，並且該標頭會被設成 true。

4. **PutRelativeFile**: 另存新檔或更改檔名。 該 URL 被呼叫時，您的應用程式必須接收 WOPI 客戶端（ OxOffice Online）傳來的檔案。

```
POST https://<WOPI host URL>/<...>/wopi/files/<id>?access_token=<token>
```

WOPI 主機必須檢查 OxOffice Online 傳來的標頭 – X-WOPI-Override 是否為英文大寫 PUT_RELATIVE。

如果您的應用程式不想讓使用者另存新檔或重新命名的話，請將 CheckFileInfo中的 UserCanNotWriteRelative 設成 true。
否則執行此功能時， OxOffice Online 會帶以下標頭：
X-WOPI-SuggestedTarget – UTF-7 編碼的新檔名。

逐步建立

想要一次把所有事物建立成功是不太可能的，我們將透過一步一步小型的、容易測試的步驟，來完成建置。

1. 在專用伺服器或虛擬機器中安裝 OxOffice Online，並確定瀏覽器可以正確連接到 OxOffice Online。

```
https://<WOPI 客戶端 URL(OxOffice Online)>:<port>/hosting/discovery
```

2. 在您的應用程式增加 WOPI REST API 的進入點。目前只須在 GET 請求時，回應「Hello World」即可，您可以透過瀏覽器進行測試。必要的話，需要用伺服器 rewrite 功能，將 REST API 導向您設計的進入點。此時，測試該進入點，應該會得到 Hello World 的回應:

```
https://<WOPI 主機 URL>/<...>/wopi/files/<file id>/contents
```

3. 實作 CheckFileInfo 進入點。確保以下位址:

```
https://<WOPI 主機 URL>/<...>/wopi/files/<file id>
```

回應 JSON 內容，類似:

```
{"BaseFileName": "test.txt", "Size": 11}
```

4. 您應該可以從步驟 1，取得 OxOffice Online 預先建置的 discovery.xml，其中的 **urlsrc** 即為 OxOffice Online 編輯擋案用的位址，您可以把 `WOPISrc=https://<WOPI 主機 URL>/<...>/wopi/files/<file id>` 加在其後，結果如下:

```
https://<WOPI 客戶端 URL>:<port>/loleaflet/<hash>/loleaflet.html?WOPISrc=https://<WOPI
主機 URL>/<...>/wopi/files/<file id>
```

接著建立一個名為 test.html 的測試頁，其中包含以下內容:

```
<html><body>
<form action="面所建立的位址" enctype="multipart/form-data" method="post">
    <input name="access_token" value="test" type="hidden"/>
        <input type="submit" value="Load OxOffice Online"/>
    </form>
</body></html>
```

當您從瀏覽器載入 test.html 並點擊 **`Load OxOffice Online`** 按鈕，將會開啟一份文字文件，內容是步驟 2 WOPI REST API 所提供的「Hello World」。如果您的 WOPI 主機和 OxOffice Online 位於不同的電腦上，請修改位於 OxOffice Online 主機的 /etc/oxool/oxoolwsd.xml，在 `<storage>` → `<wopi>` 下，新增一筆 `<host>` WOPI 主機的位址及埠號。

5. 到了這裡您已經掌握了基本要領了，然後我們開始擴充 JavaScript 的部份：建立一個包含 OxOffice Online 的 iframe，並且提供一個真正有存取權的 token。對 OxOffice Online 而言，這個存取用的
token 可以是隨機的文字、或是只有數字、字元以及底線所組成。
6. 現在，請更新您的 REST API 進入點，讓它能提供真實的檔案資料，以代替「 Hello World」
以及步驟 3 所寫死的 JSON 回應。這裡，您需要送出真實檔案內容，並參考 CheckFileInfo
表列屬性，送出相應的 JSON 回應。
7. 實作 PutFile進入點，讓編輯結果也能回存雲端硬碟。因此，需實作 POST 請求的進入點，如下：

https://<WOPI 主機 URL>/<...>/wopi/files/<file id>/contents

8. 如果您暫時不想實作 PutRelativeFile進入點，請把 CheckFileInfo的 UserCanNotWriteRelative 設為 true，待前述功能都完成後再完善它。

簡單的範例（來自 Collabora）
各種不同語言的簡單範例，不會很複雜，很好入手，您可以隨意地混合使用。

Node.js 範例
透過 Node.js 中的 iFrame 整合 OxOffice Online 的簡單範例。 我們假設您熟悉 npm 和 node.js 框架。 Node.js Example on GitHub

PHP 範例

透過 iFrame 在 PHP 中整合 OxOffice Online 的簡單範例。 我們假設您已經在電腦上安裝並啟動了 Apache Web 伺服器，並且 Apache 的 PHP 模組也已安裝並載入。
PHP Example on GitHub

Python 範例
透過 iFrame 在 Python 中整合 OxOffice Online 的簡單範例。 我們假設您熟悉 Python 和 Django 框架。 Python Example on GitHub

ReactJS 範例
一個透過 iFrame 在 ReactJS 和 Express 中整合 OxOffice Online 的簡單範例。 我們假設您熟悉 npm 和 node.js 框架。
ReactJS Example on GitHub

. NET 範例

透過 iFrame 將 OxOffice Online 與 .NET 後端整合的簡單範例。

.NET Example on GitHub

進階整合
OxOffice Online 使用類似 WOPI 的協定和整合的主機進行互動。有關該協議的詳細資訊，請參考 WOPI docs.

個人資訊流程

OxOffice Online 與使用者資料互動的唯一地方是從 CheckFileInfo取得的資料（包括文件名稱）。這包含兩個地方：日誌和使用者界面。
透過匿名功能可以去識別化所有個人資訊，而瀏覽器使用者介面只是暫時的，不會做任何儲存。

Cookies 和使用者本地儲存
OxOffice Online 不使用任何 cookies。 但 OxOffice Online 的後臺管理會使用 session cookie （關閉瀏覽器即消失） 儲存 JSON Web Token (JWT)，用來認證管理員身份。

使用者的偏好設定，會儲存在瀏覽器的 local storage（本地儲存） 中。請見下表：

切換暗色模式／亮色模式。
切換傳統介面（下拉選單 + 工具列）／分頁式操作介面。
關閉／顯示側邊攔。
關閉／顯示文件導覽。
關閉／顯示狀態列。
停用／啟用語法檢查。
最近使用的顏色及調色盤。

PostMessage API

當 WOPI 主機將 OxOffice Online 包在一個框架（ iFrame）時，可以使用 PostMessage API 用來於和 OxOffice Online 互動。這對想要完善整合的 WOPI 主機來講，非常有用。

這個 API 主要基於 WOPI 規範，很少擴充或修改。 所有送出的訊息，都採用以下形式:

```
{
	"MessageId": "<MessageId>",
	"SendTime": "<Timestamp when message is sent>",
	"Values": {
		"<key>": "<value>"
	}
}
```

`SendTime` 是瀏覽器的 Date.now() 傳回的時間戳記。從 WOPI 主機發送的 PostMessage 也應該使用相同的格式。

需要注意的是，如 WOPI 規範中所提到的，如果尚未收到 Host_PostmessageReady， OxOffice Online 框架將忽略來自 WOPI 主機框架所有訊息。

另外，由於將 OxOffice Online 嵌入 iFrame 中，因此， OxOffice Online 要求 WOPI 主機回應 **CheckFileInfo** 中的 `PostMessageOrigin` 屬性，必須指定，否則 OxOffice Online 將不會送出任何訊息。

初始化

編輯器 □□ WOPI 主機

MessageID	Values	說明

`App_LoadingStatus`	`Status: <String>`
	`DocumentLoadedTime: <Timestamp>`
	`Features: {"VersionStates":true}`
		Status 值如下:
		• **`Frame_Ready`**: OxOffice Online 的框架已經載入，並且顯示了。
		• **`Features`**: 編輯器的能力。支援的值有 - VersionStates，表示編輯器支援文件版本復原的功能。
		詳請請見: 文件版本復原
		• **`Document_Loaded`**: 使用者文件已經完全載入，WOPI 主機可以開始使用 PostMessage API。`DocumentLoadedTime` 代表載入文件所花費的時間。
		• **`Failed`**: 文件載入失敗，WOPI 主機可以顯示 OxOffice Online 框架，向使用者顯示錯誤訊息。

WOPI 主機 □□ 編輯器

MessageID	Values	說明
`Host_PostmessageReady`		通知 OxOffice Online，WOPI 主機已經準備好收送 PostMessage 了。

查詢

WOPI 主機可以透過 PostMessage API 向編輯器查詢資料。所有來自編輯器回應的 MessageID，都會在您所傳遞的 MessageID 後面加上 "_Resp"。

WOPI 主機 □□ 編輯器

MessageID	Values	說明
`Get_Views`		向編輯器查詢，目前正在編輯文件所有使用者資訊。回應會以 MessageID 為 `Get_Views_Resp` 的格式傳回。
`Get_Export_Formats`		向編輯器查詢，目前開啟的文件，支援哪些匯出格式。

得到的回應。

編輯器 □□ WOPI 主機

MessageID	Values	說明
`Get_Views_Resp`	`[ { ViewId: <Number> UserId: <String> UserName: <String> Color: <Number> ReadOnly: <Boolean> IsCurrentView: <Boolean> },... ]`	使用 `Get_Views` 查詢，所得到的所有使用者詳細資訊。
`Get_Export_Formats_Resp`	`[ { Label: <String> Format: <String> },... ]`	使用 `Get_Export_Formats` 查詢，所得到的回應。 • `Label` 是解釋該格式的本地語系內容。 • `Format` 是該格式的副檔名。

編輯者管理

WOPI 主機 □□ 編輯器

MessageID	Values	說明
`Action_RemoveView`	`ViewId: <Number>`	移除指定 Id 的編輯者，也就是將某位編輯者踢出共編行列。

編輯器 □□ WOPI 主機

MessageID	Values	說明
`Views_List`	與 `Get_Views_Resp` 相同。	所有正在參與該文件編輯的使用者詳細資訊。每次有編輯者加入或離開都會觸發此資訊。

各類指令

WOPI 主機 編輯器

MessageId	Values	說明

`Action_Save`	`DontTerminateEdit: <boolean>`
	`DontSaveIfUnmodified: <boolean>`
	`Notify: <boolean>`
	`ExtendedData: <String>`
		儲存文件。
		• `DontTerminateEdit` 和試算表相關，這會中斷編輯模式（文字游標消失）。設為 true 的話，就能儲存檔案，而不會終止編輯模式，也不會中斷使用者編輯試算表。
		• `DontSaveIfUnmodified` 如果文件內容沒有變更（只是移動了游標），這會阻止 OxOffice Online 把檔案回存雲端硬碟，如此可防止不必要的版本修訂。
		• `Notify` 設為 true 時，會在文件儲存完後，通知 WOPI 主機。詳情請參考 `Action_Save_Resp`
		• `ExtendedData` 如果標頭 `X-LOOL-WOPI-ExtendedData` 有資料，則會原封不動地傳給 WOPI 主機。資料內容不可是空白、換行或其他不可列印字元。OxOffice Online 可以傳遞多個值，然後 WOPI 主機可以進一步對處理這些值。

`Action_SaveAs`	`Filename: <String>` `Notify: <boolean>`	建立文件副本，並給予新檔名（另存新檔）。 • `Filename` 新文件檔名。 • `Notify` 設為 true 時，會在文件儲存完後，通知 WOPI 主機。詳情請參考 `Action_Save_Resp`
`Action_FollowUser`	`Follow: <Boolean>` `ViewId: <Number>`	開啟或關閉跟隨特定編輯者功能。這會讓編輯畫面和指定對象同步。 • `Follow` true 開啟 / false 關閉。 • `ViewId` 要跟隨的編輯者 ID。
`Action_Close`		關閉文件。
`Action_Print`		列印文件。
`Action_Export`	`Format: <String>`	以 `Format` 指定的格式下載文件。格式只能是 `Get_Export_Formats` 所指定列表其中之一。
`Action_InsertGraphics`	`url: <String>`	從指定的位址下載圖片並插入文件中。

MessageId	Values	說明
`Action_ShowBusy`	`Label: <String>`	顯示具有 Lable 文字的忙碌畫面，類似存檔中的畫面。
`Action_HideBusy`		如果有忙碌畫面的話，關掉忙碌畫面。
`Action_ChangeUIMode`	`Mode:'classic'` 或 `'notebookbar'`	變更使用者介面： • classic: 傳統下拉選單+工具列。 • notebookbar: 分頁式介面。
`Action_Paste`	`Mimetype: <string>` `Data: <string>`	繞過內部貼上機制，直接貼上資料到文件內。 例如： `Values: {Mimetype: "text/plain;charset=utf-8", Data: "haha"};`

編輯器 □□ WOPI 主機（回應）

MessageId	Values	說明

`Action_Load_Resp`	`success: <boolean>`
	`result: <string>`
	`errorMsg: <string>`
		文件載入完成。
		• `success` true 成功、 false 失敗。
		• `result` 文件未載入的原因。
		• `errorMsg` 載入失敗時的詳細錯誤訊息。也可能包含從 WOPI 主機傳回的錯誤訊息。

MessageId	Values	說明

`Action_Save_Resp`	`success: <boolean>`
	`result: <string>`
	`errorMsg: <string>`
	`fileName: <string>`
		文件儲存完成。只有 `Action_Save` 或 `Action_SaveAs` PostMessage API 將 `Notify` 參數設為 true 時，才會得到這個回應。
		• `success` true 成功、 false 失敗。
		• `result` 文件未儲存的原因。如果文件因未修改而未儲存，則此參數包含字串「 unmodified」。這種情況下， WOPI 主機可以確保不會儲存到位編耕的文件。
		• `errorMsg` 儲存失敗時的詳細錯誤訊息。也可能包含從 WOPI 主機傳回的錯誤訊息。
		• `fileName` 如果 `success` 為 true 則包含已儲存的檔案名稱。

`FollowUser_Changed`	`FollowedViewId: <Number>`
	`IsFollowUser: <Boolean>`
	`IsFollowEditor: <Boolean>`
		跟隨狀態變更。
		• `FollowedViewId` 哪個編輯者被跟隨。
		• `IsFollowUser` 是否啟用跟隨編輯者。
		• `IsFollowEditor` 是否啟用跟隨編輯器。
		• 如果 `IsFollowUser` 和 `IsFollowEditor` 都是 false ，表示取消跟隨功能。

`Action_ChangeUIMode_Resp`	`Mode: <string>`	變更使用介面。 `Mode` 使用哪種介面。

文件版本復原

WOPI 主機 編輯器

MessageId	Values	說明

`Host_VersionRestore`	`Status: <string>`	復原版本。唯一可能的值是 Pre_Restore 此訊息由主機在實際復原文件 之前 以及使用者確定復原 之後 傳送。 這樣，如果目前文件，還有未儲存的修改 ， OxOffice Online 可以在恢復文件之前將它們儲存到雲端硬 碟中。

編輯器 WOPI 主機

MessageId	Values	說明
`App_VersionRestore`	`Status: <string>`	這是對 `Host_VersionRestore` 訊息的回應。 目前 `Status` 的值為 `Pre_Restore_Ack` 表示 WOPI 主機可以繼續將文件回復到較早的版本。

只有當 `App_LoadingStatus` 包含功能中的 **VersionStates** 時，才會發出這些訊息。
否則，主機可以立即將版本還原到早期版本。

其他

WOPI 主機 編輯器

MessageId	Values	說明

`Insert_Button`	`id: <string>`
	`imgurl: <string>`
	`hint: <string>`
	`accessKey: <string>`
	`mobile: <boolean>`
	`tablet: <boolean>`
	`insertBefore: <string>`
	`unoCommand: <string>`
		在頂端工具列插入自訂按鈕（僅顯示在傳統介面）。該按鈕被點擊時，若未指定 `unoCommand` 屬性，會觸發 `Clicked_Button` 的事件給 WOPI 主機，讓 WOPI 主機做相應處理。
		• `id` 代表該按鈕的不可重複 ID。為了避免和原先工具列的其他按鈕 ID 衝突，建議在此類 ID 前，加上主機特有的名稱。如果和其他按鈕衝突，則不會新增按鈕。
		• `imgurl` 按鈕圖示位址。建議的大小為 24x24 px。圖片位址應和 WOPI 主機為同一來源，易免違反 `Content-Security-Policy`。
		• `hint` 按鈕的提示。
		• `accessKey` 快捷鍵。使用者可以搭配 Alt 或 Alt+Shift 加上此鍵。注意不要和其他快捷鍵衝突。
		• `mobile` 手機模式時，是否顯示該按鈕。
		• `tablet` 平板模式時，是否顯示該按鈕。
		• `insertBefore` 新增在哪個按鈕 ID 之前。請參考 查詢工具列按鈕 ID
		• `unoCommand` UNO 命令（請參閱此處）。設置這個屬性，就不會傳送 `Clicked_Button` 訊息給 WOPI 主機，而會交由 LibreOffice 直接執行此命令。

`Hide_Button`	`id: <string>`	隱藏工具列按鈕。`id` 為按鈕 ID（查詢工具列按鈕 ID）
`Show_Button`	`id: <string>`	顯示工具列按鈕。`id` 為按鈕 ID（查詢工具列按鈕 ID）
`Remove_Button`	`id: <string>`	移除工具列按鈕。`id` 為按鈕 ID（查詢工具列按鈕 ID）

`Remove_Statusbar_Element`	`id: <string>`	移除狀態列元件。`id` 是狀態列元件 ID（請參考：查詢狀態列元件 ID）
`Hide_Menubar`		隱藏下拉選單列。
`Show_Menubar`		顯示下拉選單列。
`Grab_Focus`		喚醒休眠中的編輯器。通常編輯者在一段時間後沒有任何動作時，編輯器會進入休眠狀態（編輯介面變暗，並有訊息顯示），必要時，會中斷與伺服器連線。`Grab_Focus` 會重新喚醒編輯器，必要時重新連線伺服器。
`Hide_Ruler`		隱藏水平尺規（僅適用文字文件）。 Writer
`Show_Ruler`		隱藏水平尺規（僅適用文字文件）。 Writer
`Hide_Menu_Item`	`id: <string>`	隱藏下拉選單中的選項。`id` 是指選項 ID，它們分別定義在： • 文字文件：uiconfig/text/menubar.json • 試算表文件：uiconfig/spreadsheet/menubar.json • 簡報文件：uiconfig/presentation/menubar.json • 繪圖文件：uiconfig/drawing/menubar.json
`Show_Menu_Item`	`id: <string>`	顯示下拉選單中的選項。`id` 是指選項 ID。請參閱前項 `Hide_Menu_Item`

`Disable_Default_UIAction`	`action: <string>` `disable: <Boolean>`	停用或啟用某個 UI 命令的預設行為。 • `action` 動作名稱。 • `disable` 表示停用 **(true)** 或啟用 **(false)** 預設行為。 設為 true 時，指定的動作只會發出 `postMessage` 而不會執行預設行為。這讓 WOPI 主機可以根據 `postMessage` 事件，進行必要處理。 請注意，某些動作沒有任何預設行為（如 `UI_SaveAs` 和 `UI_Share` ），因此，這對它們沒有影響；但無論如何，他們只會發出 `postMessage` 通知，而不會採取預設行為 舉例來說，`UI_Save` 預設會呼叫儲存命令（透過選單、工具列按鈕或鍵盤快捷鍵）。如果透過 `Disable_Default_UIAction` 停用 `UI_Save`，則不會呼叫儲存命令。 停用 `UI_Save` 的 WOPI 主機，應該在文件需要儲存時，自行發出 `Action_Save` 。 同樣地，當停用 `UI_Close` 時，呼叫 `UI_Close` ，並不會真的文件關閉，而是會發出 postMessage 通知，並且由 WOPI 主機在需要時發出 `Action_Close` 。 WOPI 主機必須很小心，不要在啟用預設行為的狀態下，發出重複的 `Action_xxyyzz` 等指令。相反地，只能在停用預設行為時，才能自行發出操作。 注意：目前也只有 `UI_Save` 和 `UI_Close` 以及 `UI_Hyperlink` □□□□
`Send_UNO_Command`	`Command: <string>` `Args: <object>`	向編輯器傳送 UNO 命令。請參考: framed.doc.html

查詢工具列按鈕 ID

工具列按鈕 ID 在下列程式中的 getToolItems/create 函數定義:

- Control.TopToolbar.js: 用於桌面或平板電腦上的頂部工具列。
- Control.MobileTopBar.js: 用於智慧型手機上的頂部工具列。
- Control.MobileBottomBar.js: 用於智慧型手機底部工具列。

- Control.StatusBar.js: 用於桌面上的狀態列。

這些通常不會改變，但並不保證這些 ID 會穩定存在。

查詢狀態列元件 ID

狀態列元件 ID 定義在 Control.StatusBar.js 的 `onDocLayerInit` 函數中。通常不會改變，但並不保證這些 ID 會穩定存在。

事件通知

編輯器 □□ WOPI 主機

MessageId	Values	說明
`Clicked_Button`	`id: <string>`	點擊透過前面的 `Insert_Button` API 新增的自訂按鈕時會發出此事件。
`Download_As`	`Type: 'print'|'slideshow'|'export'` `URL: <string>`	當使用者選擇「列印」或「全螢幕投影」或「下載為 ...」，且 WOPI 主機在 **CheckFileInfo** 的 `DownloadAsPostMessage` 屬性中，指定要接管時，就會觸發此事件。 這應用於無法依賴瀏覽器下載的情況，例如手機的 Web view 模式。
`UI_Close`	`EverModified: true | false`	當使用者選擇關閉文件，或點擊關閉按鈕，將觸發此事件。 `EverModified` 指出此文件是否修改過。
`UI_Save`	`source: 'filemenu' | 'notebookbar' | 'toolbar'`	使用者選擇儲存。 `source` 指出透過何種方式儲存。
`UI_CreateFile`	`DocumentType: 'text' | 'spreadsheet' | 'presentation' | 'drawing'`	要求 WOPI 主機開啟新的瀏覽器分頁，並建立新文件。文件類型以 DocumentType 參數傳遞，可以是「 text （文字文件）」、「 spreadsheet （試算表）」、「 presentation （簡報）」或「 drawing （繪圖）」。

MessageId	Values	說明
`UI_SaveAs`		要求 WOPI 主機顯示另存新檔 UI，以便使用者可以選擇路徑和檔案名稱來建立目前檔案的副本。 若使用者確定存檔，則 WOPI 主機需以 `Action_SaveAs` 命令編輯器發送檔案內容。
`UI_Cancel_Password`		通知 WOPI 主機使用者在開啟有密碼保護的檔案時，點擊了「取消」，而不是輸入密碼。
`UI_Hyperlink`		通知 WOPI 主機只用者點擊了超連結，並確認確實想要離開文件以開啟超連結。這對於將 OxOffice Online 嵌入到行動裝置 iframe 中的 App 特別有用，實際嘗試開啟新視窗，應觸發在 Android 上的 Activity（或在 iOS 上類似的操作）。 WOPI 主機可能也想利用 `Disable_Default_UIAction` 來接管 `UI_Hyperlink`
`Doc_ModifiedStatus`	`Modified: true | false`	更新文件的修改狀態。如果文件自上次儲存後被修改，則 `Modified` 將為 **true**，否則，如果文件已儲存，則 `Modified` 為 **false**。 請注意，這個通知可能會在不更改先前值的情況下觸發，因此必須檢查 `Modified` 的值，不要假設這個通知暗示文件的修改狀態。

呼叫 **Python** 巨集

WOPI 主機 ◫ 編輯器

MessageId	Values	說明

`CallPythonScript`	`ScriptFile: <string>`
	`Function: <string>`
	`Values: <object>`
		執行指定功能的 Python 巨集:
		• ScriptFile: 巨集檔名
		• Function: 函數名稱
		• Values: 參數內容（key: value, ...）

□□□ □□ WOPI □□

MessageId	Values	□□
`CallPythonScript-Result`	`commandName: <string>` `Values: <object>`	□□ Python □□□□□ `commandName` □□□□□□ URL□