ASUS CLOUD CORPORATION

WebRelay

Version 11.15

Austin Chen

2012

Rm. 817, 8F, No. 53, Minquan Rd., Yonghe Dist., New Taipei City, Taiwan

Content

WebRelay 功能簡介

WebRelay 的功能主要是提供客戶端在 ASUS WebStorage 系統中進行實體檔案的存取動作，並支援網路瀏覽器(Web Browser)的執行。例如：檔案上傳、檔案下載、影片格式轉換和取得圖片檔的縮圖…等。

WebRelay 分享功能簡介

ASUS WebStorage 系統所提供的分享(Share)功能有：一般分享和群組分享。

1. 一般分享

可將檔案／目錄分享給所有人，無論是否擁有 ASUS WebStorage 帳號皆可瀏覽下載。

2. 群組分享(Group Share)

設定協同合作資料夾

用戶可呼叫/fsentry/setadvancedsharecode/ API 將自己的目錄指定分享給特定的 ASUS WebStorage 用戶，以便啟用**協同合作資料夾**功能。

這群特定的 ASUS WebStorage 用戶屬於**白名單**，白名單內的用戶可以新增子目錄、上傳檔案到該分享目錄，或是對該目錄下的子目錄／檔案進行新增、搬移、更名或刪除等動作；若非白名單用戶，則無任何權限。

※ 備份資料夾無法被設定為協同合作資料夾。

※ 群組分享功能目前僅支援設定目錄(Folder)為協同合作資料夾。

存取協同合作資料夾

客戶端可依照下述步驟取得協同合作資料夾的 Folder ID：

1. 收到雲端分享碼之後，可呼叫 ServicePortal 的/sharecode/findservicegateway/ 用雲端分享碼查詢 ServiceGateway 的位址。
2. 取得 ServiceGateway 位址後，可呼叫 ServiceGateway 的/member/acquiretoken/

以透過使用者帳號、密碼的驗證，來取得授權金鑰(Token)，並查詢到各 Server 的位址(例如：InfoRelay 位址)，以便進行其他 API 的操作。

3. 得到 Token 和 InfoRelay 位址之後，可呼叫/fsentry/getinfobysharecode/查詢雲端分享碼所代表的協同合作資料夾 Folder ID。

使用協同合作資料夾

如欲使用協同合作資料夾功能時，其操作之客戶端應用程式(Client Application)必須在 HTTP Header 中提供 **SID 及 ProgKey**(必須採用 Authorization 的方式，Authorization Header 請參見 ServiceGateway 技術文件的 /member/acquiretoken/)，當 **SID 及 ProgKey** 通過 ASUSCloud Server 的認證後，方能啟用協同合作資料夾功能。其功能如下：

1. 協同合作資料夾與檔案／目錄操作相關之 InfoRelay API 列表：
   - /folder/create/
   - /folder/move/
   - /file/move/
   - /folder/rename/
   - /file/rename/
   - /folder/remove/
   - /file/remove/
   - /find/propfind/

以上 API 皆有一個選擇性輸入參數**<isgroupaware>(** /find/propfind/ 則是參數**<isshared>)**，將輸入參數<isgroupaware>值(或是/find/propfind/的<isshared>)設為 1 才會啟用協同合作資料夾功能；若不輸入該參數或者輸入的值為 0，則將不會啟用協同合作資料夾功能(詳情請參見 InfoRelay 技術文件)。

※ 範例：

用戶 Anson 將目錄 AlienVideo 設為協同合作資料夾並指定分享給白名單成員為 Bush、Cella。當 Bush、Cella 呼叫/file/rename/ API，並將參數<isgroupaware>設為 1 後可將 AlienVideo 目錄下的指定檔案更名；若 Bush、Cella 未指定

<isgroupaware>值為 1，則此更名動作將失敗。

2. 協同合作資料夾上傳檔案相關之 WebRelay API 列表：

- /webrelay/directupload/
- /webrelay/initbinaryupload/
- /webrelay/resumebinaryupload/
- /webrelay/finishbinaryupload/

亦需指定 SID／ProgKey／啟用協同合作資料夾的輸入參數，方可使用協同合作資料夾功能。

WebRelay API 簡介

若客戶端程式欲呼叫使用 WebRelay 的 API，則必須先向 ServiceGateway 取得 Token 及 WebRelay 的位址，步驟如下：

一、首先，客戶端必須先向 ServicePortal 取得本身所屬之 Service Area 的 ServiceGateway 位址。

註：透過呼叫 ServicePortal 的/member/requestservicegateway/可以達成這個目的。

二、接著，客戶端需要帶著使用者的帳號(User ID)和密碼(Password)，向 ServiceGateway 呼叫/member/acquiretoken/以取得 Token。

1. 將客戶端傳來的使用者帳號和密碼進行認證。
2. 若帳號密碼無誤，則 ServiceGateway 將會核發 Token，並且將 InfoRelay / WebRelay 的位址回傳給客戶端。取得 WebRelay 的位址之後，客戶端便可以開始呼叫使用 WebRelay 的 API。

Multipart Upload(/webrelay/directupload/)

此 API 將透過 HTTPS Multipart Upload 的方式將檔案上傳到 ASUS WebStorage 空間，並回傳該檔案所屬的 File ID。

若客戶端欲取得上傳進度，則必須先對/webrelay/initsession.jsp 發一個 HTTP Request，待客戶端收到 Response 後，WebRelay 將在客戶端的 Cookie 中加入 HTTP Session，接著客戶端就可以呼叫/webrelay/directupload/進行檔案上傳，最後才可以經由 /webrelay/getuploadprogress/ 取得檔案的上傳進度。而呼叫 /webrelay/initsession.jsp 、/webrelay/directupload/ 、/webrelay/getuploadprogress/ 這三個 API 的客戶端必須為同一個 IP 才行。

※ API 參考：查詢上傳檔案進度(/webrelay/getuploadprogress/)

Streaming Upload

WebRelay 提供了一組以串流(Stream)為基礎所進行傳輸的 API。並支援斷點續傳以進一步優化檔案上傳作業。

該組 API 分別為：/webrelay/initbinaryupload/、/webrelay/resumebinaryupload/、/webrelay/finishbinaryupload/。若客戶端欲使用 Binary Stream 為基礎的上傳檔案 API 時，按照順序分為三大步驟：

一、初始化串流檔案上傳／續傳作業

1. 呼叫/webrelay/initbinaryupload/。
2. 取得 Transaction ID。
   ※ 若客戶端要進行續傳作業，則呼叫/webrelay/initbinaryupload/時必須指定 tx 參數值為 Transaction ID(可由上一次伺服端回傳的 Payload 中取得)。

二、串流檔案上傳／續傳

1. 接續第一步驟，呼叫/webrelay/resumebinaryupload/。
2. 將檔案的二進位資料透過 HTTP POST 寫至 WebRelay。

※ 若客戶端要執行續傳，可以由第一步驟 WebRelay 回傳的<offset>得知檔案前次上傳進度的 byte 值，藉此接續上傳檔案內容。

3. 客戶端將檔案內容完全上傳完畢之後，須執行最後步驟(步驟三)，方能確實完成串流檔案上傳。

三、完成串流檔案上傳／續傳

1. 客戶端經由/webrelay/resumebinaryupload/將檔案內容完全上傳完畢之後，請客戶端務必呼叫/webrelay/finishbinaryupload/，並指定 Transactoin ID，以通知 WebRelay 檔案已上傳完畢。

※ 若客戶端未呼叫/webrelay/finishbinaryupload/，則 WebRelay 會認為客戶端只是暫停上傳檔案，即未完成上傳。

覆寫檔案(Reuse File ID)：

Streaming Upload 除了可以上傳新檔案和斷點續傳之外，還可將原本存於雲端的檔案覆寫。

客戶端上傳檔案時，若將欲覆寫的檔案 File ID 傳至伺服端，可得到該檔案在 ASUS WebStorage 上的 checksum [註2] (請參閱本文件末的註解)，checksum 即為檔案的特徵值字串。

當客戶端呼叫/webrelay/finishbinaryupload/告知伺服端上傳完成時，客戶端須再將此 checksum 送回伺服端，伺服端藉由比對當時雲端上檔案的 checksum 和客戶端送來的 checksum 是否一致，可以確認欲覆寫的檔案在上傳過程中是否有過異動(若發生過異動則 checksum 將不一致，伺服端會回傳 Status Code 250)，如此可確保各同步裝置在同步過程中的正確率。

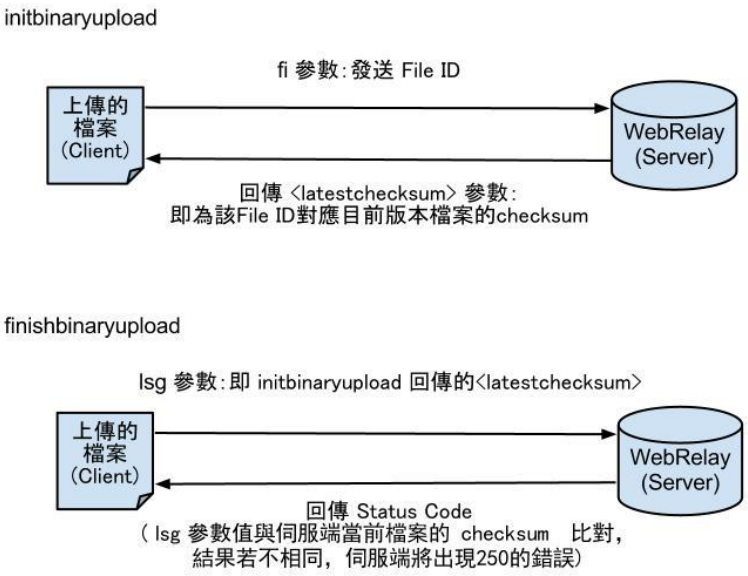
流程說明：

首先，客戶端須在 Streaming Upload 步驟一裡輸入 fi 參數指明 File ID，而 WebRelay 將會回傳伺服器中該 File ID 的 checksum。

接著，客戶端在步驟三時，必須將步驟一中從伺服端接收到的 checksum 傳

回至 WebRelay，伺服端將會把步驟三上傳的 checksum 與存於伺服端的 checksum 進行比對，以確認在上傳過程中是否與其他同步裝置有更新的差異。

比對結果若不相同，伺服端將出現 250 的錯誤訊息，表示欲上傳的檔案在各同步裝置中可能有例外的更新狀況發生，客戶端便可檢查各同步裝置的更新狀況，避免各裝置同步更新發生差異。下圖為覆寫檔案(Reuse File ID)流程圖：



覆寫檔案(Reuse File ID)流程圖

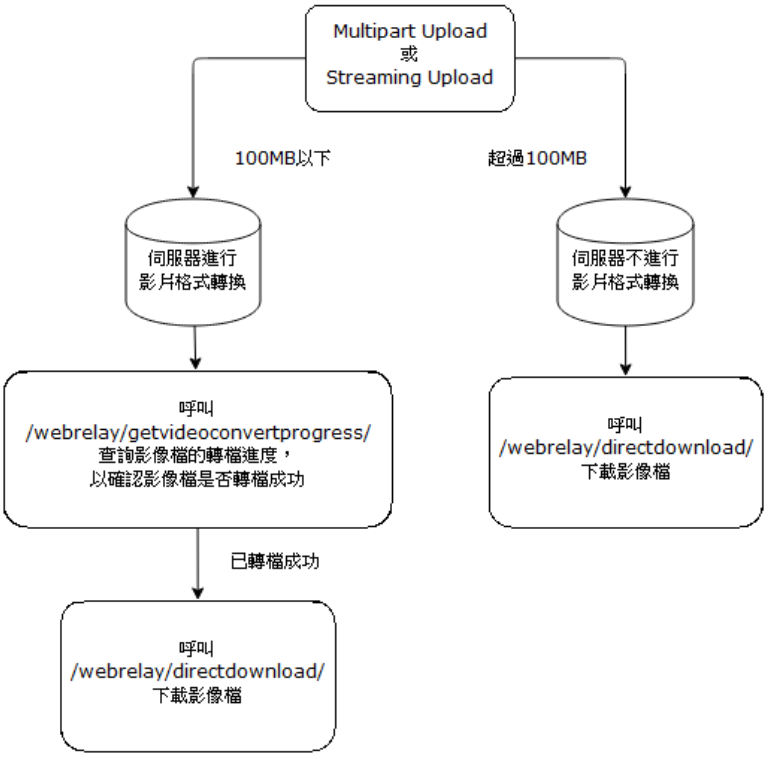
※ 例如：

客戶端上傳檔案 A'，欲覆寫在雲端上的檔案 A，客戶端須將檔案 A 的 File ID 傳至伺服端，伺服端便會回傳檔案 A 的 checksum。當客戶端使用 Streaming Upload 呼叫到/webrelay/finishbinaryupload/時，客戶端須將先前拿到檔案 A 的 checksum 再送至伺服端，伺服端將會拿此 checksum 跟此時在伺服端上檔案 A 的 checksum 做比對，以確保檔案 A 沒有在 Streaming Upload 過程中因同步功能發生異動。

影片格式轉換

客戶端可使用 Multipart Upload 或 Streaming Upload 上傳影片檔，若影片檔大小為 100MB 以下，則伺服端將會自動進行影片格式轉換，伺服器除保留原始檔外，還會產出一個或數個解析度的 MP4 檔在伺服端(並不會額外佔據客戶端的 WebStorage 空間)，提供給不同解析度裝置使用；若影片檔大小超過 100MB，則不進行轉檔。

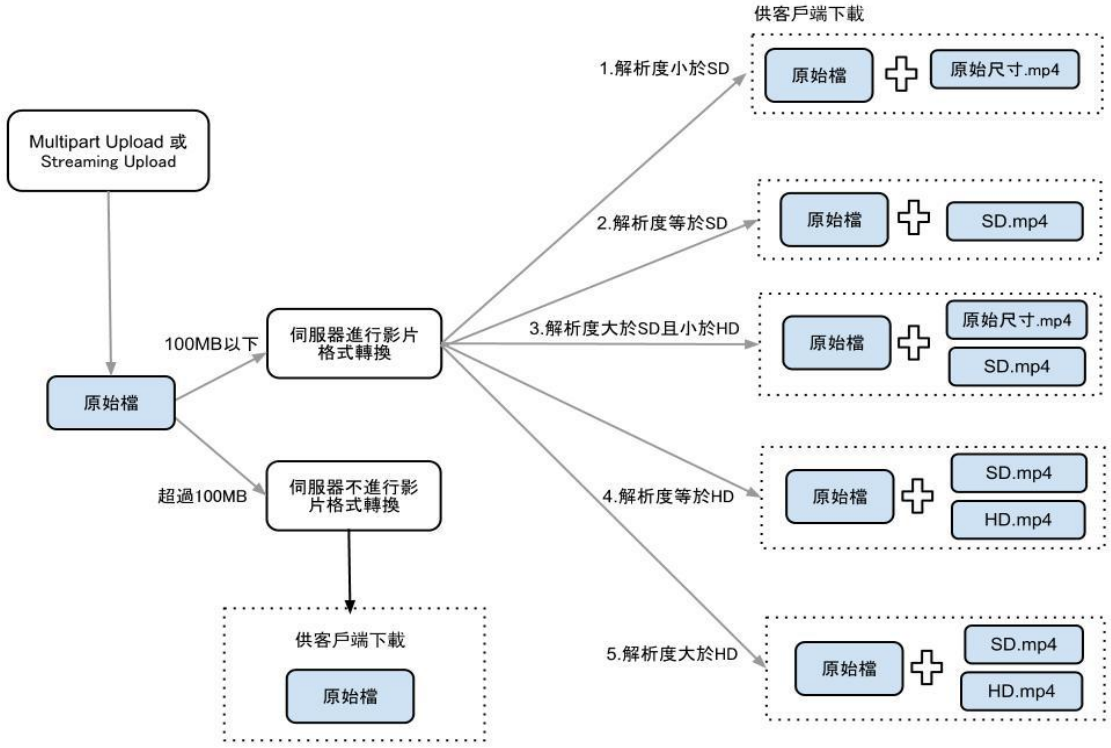
若客戶端欲下載轉換過後的影片檔，則須在下載之前，先呼叫「查詢影片轉檔進度(/webrelay/getvideoconvertprogress/)」以查詢確認影片檔案是否已轉檔成功。待確認已轉檔完畢後，方能順利下載，其流程請參考下圖。



下載影像檔流程圖

轉換標準：

影片格式轉換所使用的解析度分別為 **SD(320 x 480)**和 **HD(1280 x 720)**。客戶端上傳的原始影像檔解析度有下述五種情況，在各情況下，伺服端將產出一個或數個解析度的 MP4 檔。其相關流程請參考下方「影片格式轉換流程圖」：



影片格式轉換流程圖

影片格式轉換有下述五種情況：

1. 解析度小於 SD：

   客戶端可下載原始解析度的 MP4 檔。

   例如：若上傳的影片為 180 x 120 的 WMV 檔，則客戶端可下載：

   - 180 x 120 的 MP4 影像檔。

2. 解析度等於 SD：

   客戶端可下載 SD 解析度的 MP4 檔。

   例如：若上傳的影片為 SD 的 AVI 檔，則客戶端可下載：

   - SD 的 MP4 影像檔。

3. 解析度大於 SD 且小於 HD：

   客戶端可下載原始解析度的 MP4 檔、SD 解析度的 MP4 檔。

   例如：若上傳的影片為 980 x 640 的 RMVB 檔，則客戶端可下載：

   - 980 x 640 的 MP4 影像檔、SD 的 MP4 影像檔。

4. 解析度等於 HD：

   客戶端可下載 SD 解析度的 MP4 檔、HD 解析度的 MP4 檔。

   例如：若上傳的影片為 HD 的 RMVB 影像檔，則客戶端可下載：

   - SD 的 MP4 影像檔、HD 的 MP4 影像檔。

5. 解析度大於 HD：

   客戶端可下載 SD 解析度的 MP4 檔、HD 解析度的 MP4 檔。

   例如：若上傳的影片為 2050 x 2000 的 RMVB 影像檔，則客戶端可下載：

   - SD 的 MP4 影像檔、HD 的 MP4 影像檔。

下載檔案(/webrelay/directdownload/)

當 WebRelay 接收到客戶端下載檔案的請求時，WebRelay 會先取得/member/acquiretoken/中的 Token，將此 Token 送往 ServiceGateway 驗證，驗證通過後，WebRelay 會找出客戶端指定的 File ID 在伺服器上的儲存路徑與檔名，並讀取該檔案回傳給客戶端。

※此 API 支援 HTTP Range Partial Download，也支援續傳。

由指定檔案萃取文字檔 (/webrelay/getfulltextcompanion/)

客戶端使用/webrelay/getfulltextcompanion/可以對指定之檔案萃取出文字檔(網頁中萃取出文字、MS Word 檔案中萃取出純文字…等)，客戶端應用程式可以自行決定這個檔案的應用方式。

擷取影片預覽圖(/webrelay/getvideosnapshot/)

客戶端使用/webrelay/getvideosnapshot/可以對指定之影音檔案擷取預覽圖片，並擷取影片中第 10 秒的畫面做為網頁上供用戶預覽影音檔內容之用。

本文件 API 閱讀須知

此段落將說明如何閱讀本文件的 XML Payload (即各個 API 的 Input、Output)。

- 以/webrelay/directdownload/傳給伺服器的 XML Payload 為例(擷取段落)：

URL rewrite 路徑參數：/{ token }/

Query String 之中各個參數以'&'分隔：

dis = { sid }

fi = { 要下載之 File ID }

pv = [*0* | *1*]

<!-- 是否需要預覽(preview)。0：直接下載(預設值) | 1：預覽。 -->

(以下省略，API 詳情內容請參閱：/webrelay/directdownload/)

閱讀說明：

參數欄位	說明
dis={ sid }	參數欄位可輸入的值以**{ }**做為表示。 如左例：若 sid 為 12345，則 Payload 為 dis = 12345
pv=[*0* | *1*] <!-- 是否需要預覽(preview)。0：直接下載(預設值) | 1：預覽。 -->	參數欄位**[ ](中括號)**的***斜體字***表示為實際上參數欄位可出現的值，並以**<!---->**補充說明文字。 如左例：[*0* | *1*]意指可輸入 0 或是 1。

API 說明(SSL 加密)

Multipart Upload(/webrelay/directupload/)

目的:接受客戶端以 HTTP Multipart Upload 方式將檔案上傳至 ASUS WebStorage 空間。此 API 並不支援續傳,若發生斷線,客戶端必須重頭開始再傳一次。

※ 附註：若客戶端需要斷點續傳，請參考本文件中的「Streaming Upload」。

如果輸入參數 ar (Auto Rename) 設為 1，更名規則即在原始檔名後、副檔名前安插一個空白後再加上括號(數字 n)，若有相同檔名則遞增數字，直到沒有重覆檔名。

例如：原始檔名為

「瑪莉與她的小綿羊.mov」遇到同檔名則改為…

「瑪莉與她的小綿羊 (1).mov」→加上“一個空白”和“括號(數字 n)”

※ 若客戶端欲取得上傳進度，必須先呼叫過/webrelay/initsession.jsp。否則，使用/webrelay/getuploadprogress/時將無法取得正確的上傳檔案進度。

協同合作資料夾：

如欲使用協同合作資料夾功能時，其操作之客戶端應用程式(Client Application)必須以 Query string 的方式帶入 **SID**、**isgroupaware** 參數及 **ProgKey**(必須採用 Authorization 的方式，Authorization Header 請參見 ServiceGateway 技術文件的/member/acquiretoken/)，當 **SID** 及 **ProgKey** 通過 ASUSCloud Server 的認證後，方能啟用協同合作資料夾功能(如欲詳知 ASUS WebStorage 系統所提供的分享功能，請參見 InfoRelay 技術文件)。

※ 註：目前 ASUS WebStorage 系統未提供跨服務區域使用群組編輯的功能。

回傳的狀態碼(Status Code)：

0 Success。

2 Authentication Fail。

5 Developer Authentication Fail。(例如：sid 不存在或 ProgKey 驗證失敗。)

218 Folder ID 不存在(用戶所指定的父目錄不存在)。

221 單一檔案超過大小限制。

224 用戶之儲存空間已用盡。

226 用戶的帳號已被凍結(FROZEN)或關閉(CLOSE)。

999 General Error。

回傳的 HTTP 狀態碼(HTTP Status Code)：

403 Token 驗證失敗。

/webrelay/directupload/
Input
URL rewrite 路徑參數： /{ token }/?**dis**={ sid }& **igw** = [*0* | *1*] & **atrz** = { HTTP Header "Authorization" } Query String 路徑參數詳細說明： **dis** = { sid } **igw** = [*0* | *1*] <!-- 選擇性欄位，預設值為 0。此欄為群組分享參數，同 InfoRelay 技術文件中的<isgroupaware>。若為 1 表示要對協同合作資料夾進行上傳檔案的動作，反之則不啟用協同合作資料夾功能 --> **atrz** = { 其意義同 Service Gateway 的/member/acquiretoken/ API 所需指定的 HTTP Header "Authorization"。因採用 Query string 方式，所以此參數須經 Base64 編碼完畢後再做 URL encode }<!-- 選擇性欄位 --> ※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。

Multipart FORM 輸入參數：

pa = { Parent Folder ID }

d = { 經過 Base64(UTF-8)編碼後 URL encode 的目錄名稱 }

<!-- 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現-->

u = { Response URL }

pr = { Progress ID }

<!-- 由客戶端指定，做為呼叫/webrelay/getuploadprogress/以查詢上傳進度時所用的 ID。請參閱/webrelay/getuploadprogress/的參數 **pr** -->

at = { Attribute 參數[註1] }

<!-- 供客戶端指定任意想儲存在檔案的屬性(attribute)內的屬性字串，指定此值時須先經 URL encode，再傳至 WebRelay 儲存。 -->

fs = { File Size }

<!-- 此參數讓客戶端指定欲上傳之檔案大小(單位為 Byte)，藉此讓 WebRealy 在接收檔案之前可判斷是否超過系統允許的上限。超過系統允許上限大小的檔案 WebRelay 會拒絕接收，回傳狀態碼 221。若未指定此參數，則會接收檔案直到超過系統允許的上限時才回傳狀態碼 221。 -->

fi = { 舊檔案的 File ID }

<!-- 選擇性欄位。如欲覆蓋原存在於伺服器上的檔案，須指定此參數。若指定的 File ID 不存在，則將視為全新上傳。 -->

ar = [*0* | *1*]

<!-- 選擇性欄位(預設值為 0)。該欄位是自動更名(Auto-Rename)的布林參數。當指定上傳的檔案與伺服器上檔案同名時，指定參數為 0 表示不自動更名；若為 1 表示要自動更名。-->

rn = { 將上傳之檔案的檔名重新命名(rename)為此參數指定的檔案名稱 }

<!-- 選擇性欄位。此字串須依序以 Base64 及 URL encode 進行編碼。

例如：欲將下載檔案易名為 Naomi_PhotoShop_Tutorial.mov。

Raw flie name：Naomi_PhotoShop_Tutorial.mov

對 Raw flie name 的編碼步驟：

1. Base64：TmFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg==
2. URL encode：

TmFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg%3D%3D

故設 rn=TmFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg%3D%3D

※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。 -->

Output

1.若輸入參數有指定 u (客戶端為瀏覽器)

以 u 所指定的 URL 作為重新導向目的地網址(URL redirection)，後綴下列參數：

?**s**={ Status Code }&**f**={ Parent Folder ID }&**p**={ Panel ID }&**d**={ Folder display }&**n**={ File ID }

2.若輸入參數未指定 u

XML 回傳結果為：

```
<?xml version="1.0" encoding="utf-8"?>
<directupload>
    <status>{ Status Code }</status>
    <fileid>{ 上傳完成的 File ID }</fileid>
    <rawfilename>{ 檔案名稱 }</rawfilename>
</directupload>
```

查詢上傳檔案進度(/webrelay/getuploadprogress/)

目的：供客戶端查詢指定檔案上傳到 ASUS WebStorage 的上傳進度。

回傳的狀態碼(Status Code)：

0 Success，回傳進度。

999 General Error (例如：Server 查無此 Progress ID)。

/webrelay/getuploadprogress/

Input

Query String 中的輸入參數：

pr ={ 客戶端呼叫/webrelay/directupload/時指定的 Progress ID }&**pgid**={ pgid,前

端網頁所需顯示參數 }&**dis**={ sid }
Output
progress({ pgid,來自輸入參數 }, { Status Code }, { 已上傳之 byte 數 }, { ContentLength 檔案大小 });

Streaming Upload

初始化串流檔案上傳/續傳(/webrelay/initbinaryupload/)

Streaming Upload 是以串流(Stream)為基礎所進行傳輸的 API，其支援斷點續傳以進一步優化檔案上傳作業。此 API 為 Streaming Upload 的第一步驟，在此步驟中有兩個目的：

1. 開始上傳：

   新建立一個串流上傳檔案的 session，並取得一個 Transaction ID，做為後續上傳動作的啟始點。

2. 斷點續傳：

   如果傳輸斷線了，而客戶端想要取得前次上傳中斷點時，則須呼叫此 API，並帶入前次上傳的 Transaction ID，以取得前次中斷點的 offset。

※ 欲使用串流方式上傳檔案，必須以正確步驟呼叫 API，才能上傳成功(詳情請參閱本文 Streaming Upload)。

協同合作資料夾：

如欲使用協同合作資料夾功能時，其操作之客戶端應用程式(Client Application)必須在 HTTP Header 中提供 **SID 及 ProgKey**(必須採用 Authorization 的方式，Authorization Header 請參見 ServiceGateway 技術文件的 /member/acquiretoken/)，當 **SID 及 ProgKey** 通過 ASUSCloud Server 的認證後，方能啟用協同合作資料夾功能(如欲詳知 ASUS WebStorage 系統所提供的分享功能，請參見 InfoRelay 技術文件)。

※ 註：目前 ASUS WebStorage 系統未提供跨服務區域使用群組編輯的功能。

回傳的狀態碼(Status Code)：

0 Success。

2 Authentication Fail。

5 Developer Authentication Fail。(例如：sid 不存在或 ProgKey 驗證失敗。)

211 檔案名稱為空白(輸入參數 na)。

213 檔案名稱長度超過限制(輸入參數 na)。

214 檔案名稱重複(輸入參數 na)。

218 要處理的目錄不存在或已刪除(輸入參數 pa)。

219 檔案不存在或已刪除(客戶端進行 Reuse File ID 時)。

220 一般檔案錯誤。

221 單一檔案超過大小限制。

224 用戶之儲存空間已用盡。

226 用戶的帳號已被凍結或關閉。

251 Transaction ID 不存在。

252 Transaction ID 與 File ID 不符(客戶端進行 Reuse File ID 時) 。

999 General Error。

/webrelay/initbinaryupload/
Input
Query String 之中各個參數以'&'分隔： **dis** = { sid } **tk** = { token } **na** = { 經過 Base64(UTF-8)編碼後 URL encode 的檔案名稱 } <!-- 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現 --> **pa** = { Parent Folder ID } **sg** = [SYSTEM.RESERVED] <!-- 系統保留參數，填固定字串 --> **at** = { Attribute 參數[註1](請參閱本文件末的註解) } <!-- 供客戶端指定任意想儲存在檔案的屬性(attribute)內的屬性字

串，指定此值時須先經 URL encode，再傳至 WebRelay 儲存。 -->

fs = { File Size }

<!-- 此參數讓客戶端指定欲上傳之檔案大小(單位為 Byte)，藉此讓 WebRealy 在接收檔案之前可判斷是否超過系統允許的上限。超過系統允許上限大小的檔案 WebRelay 會拒絕接收，回傳狀態碼 221。若未指定此參數，則會接收檔案直到超過系統允許的上限時才回傳狀態碼 221。-->

tx = { 續傳時才使用的 Transaction ID }<!-- 選擇性欄位 -->

fi = { 要覆寫的 File ID }<!-- 選擇性欄位 -->

sc = { 此檔案上傳目的地為 Sync Folder 或 Sync Folder 的子目錄，以此參數指明其最上層為 Sync Folder }<!-- 選擇性欄位 -->

igw = [*0* | *1*]

<!-- 選擇性欄位，預設值為 0。此欄為群組分享參數，同 InfoRelay 技術文件中的<isgroupaware>。若為 1 表示要對協同合作資料夾進行上傳檔案，反之則不啟用協同合作資料夾功能 -->

Output

```
<?xml version="1.0" encoding="utf-8"?>
<initbinaryupload>
    <status>{ Status Code }</status>
    <transid>{ Transaction ID }</transid>
    <offset>{ 新檔案為 0，續傳則為前一次上傳檔案的進度(EOF+1)，客戶端由此欄指示的位元組開始上傳 }</offset>
    <latestchecksum>{ 當輸入的 Query String 中指定了 fi 參數時，即表明要 Reuse File ID。若 DB 中已存有該 File ID 對應之目前版本檔案的 checksum 則 WebRelay 會將 checksum 回傳給客戶端 }</latestchecksum><!-- 若非 Reuse File ID 則不會有此欄位 -->
</initbinaryupload>
```

串流檔案上傳/續傳(/webrelay/resumebinaryupload/)

目的：在呼叫/webrelay/initbinaryupload/ API 取得 Transaction ID 後，即可以此 API 透過串流的方式 (Stream) 進行指定 Transaction ID 的檔案上傳作業。若為續傳的狀態，在取得/webrelay/initbinarupload/ API 回傳的 offset 後，即可進行後續的檔案上傳。

協同合作資料夾：

如欲使用協同合作資料夾功能時，其操作之客戶端應用程式(Client Application)必須在 HTTP Header 中提供 **SID 及 ProgKey**(必須採用 Authorization 的方式，Authorization Header 請參見 ServiceGateway 技術文件的 /member/acquiretoken/)，當 **SID 及 ProgKey** 通過 ASUSCloud Server 的認證後，方能啟用協同合作資料夾功能(如欲詳知 ASUS WebStorage 系統所提供的分享功能，請參見 InfoRelay 技術文件)。

※ 註：目前 ASUS WebStorage 系統未提供跨服務區域使用群組編輯的功能。

回傳的狀態碼(Status Code)：

0 Success。

5 Developer Authentication Fail。(例如：sid 不存在或 ProgKey 驗證失敗。)

220 一般檔案錯誤。

251 Transaction ID 不存在。

999 General Error。

回傳的 HTTP 狀態碼(HTTP Status Code)：

403 Token 驗證失敗。

/webrelay/resumebinaryupload/
Input
Query String 之中各個參數以'&'分隔： **dis** = { sid } **tk** = { token } **tx** = { Transaction ID } **igw** = [*0* | *1*] <!-- 選擇性欄位，預設值為 0。此欄為群組分享參數，同 InfoRelay 技術文件中的<isgroupaware>。若為 1 表示要對協同合作資料夾進行上傳檔案，反之則不啟用協同合作資料夾功能 -->

上傳內容則是客戶端取得 OutputStream 後以 Binary Stream 的方式寫入。
Output
<?xml version="1.0" encoding="UTF-8"?> <resumebinaryupload> 　　<status>{ Status Code }</status> </resumebinaryupload>

完成串流檔案上傳/續傳(/webrelay/finishbinaryupload/)

目的：在呼叫/webrelay/resumebinaryupload/ API 進行檔案上傳完畢後，必須呼叫此 API 告知伺服端已完成檔案上傳，伺服端確知客戶端不是暫停上傳後，便可進行檔案上傳完畢之處理程序，並傳回 File ID。

協同合作資料夾：

如欲使用協同合作資料夾功能時，其操作之客戶端應用程式(Client Application)必須在 HTTP Header 中提供 **SID 及 ProgKey**(必須採用 Authorization 的方式，Authorization Header 請參見 ServiceGateway 技術文件的 /member/acquiretoken/)，當 **SID 及 ProgKey** 通過 ASUSCloud Server 的認證後，方能啟用協同合作資料夾功能(如欲詳知 ASUS WebStorage 系統所提供的分享功能，請參見 InfoRelay 技術文件)。

※　註：目前 ASUS WebStorage 系統未提供跨服務區域使用群組編輯的功能。

回傳的狀態碼(Status Code)：

0　Success。

2　Authentication Fail。

5　Developer Authentication Fail。(例如：sid 不存在或 ProgKey 驗證失敗。)

218　要處理的目錄不存在或已刪除。

219　檔案不存在或已刪除(客戶端進行 Reuse File ID 時)。

220　一般檔案錯誤。

250 輸入參數的檔案 signature(checksum)[註 2]為空白或與伺服端不相符。

客戶端上傳的 latestchecksum(lsg 參數)與 DB 中的值不一致。

999 General Error。

/webrelay/finishbinaryupload/
Input
Query String 之中各個參數以'&'分隔： **dis** = { sid } **tk** = { token } **tx** = { Transaction ID } **lsg** = { 僅用於 Reuse File ID 時，也就是呼叫/webrelay/initbinaryupload/回傳之 XML 中的<latestchecksum> }<!-- 選擇性欄位 --> **igw** = [0 | 1] <!-- 選擇性欄位，預設值為 0。此欄為群組分享參數，同 InfoRelay 技術文件中的<isgroupaware>。若為 1 表示要對協同合作資料夾進行上傳檔案，反之則不啟用協同合作資料夾功能 -->
Output
<?xml version="1.0" encoding="utf-8"?> <finishbinaryupload> <status>{ Status Code }</status> <fileid>{ 上傳完成之 File ID }</fileid> </finishbinaryupload>

查詢影片轉檔進度 (/webrelay/getvideoconvertprogress/)

目的：當客戶端上傳小於 100MB 的影片檔至伺服器後，伺服器除保留原始檔外，將會自動進行影像轉換，產出一個或數個解析度的 MP4 檔(並不會佔據客戶端的 WebStorage 空間)，以提供給不同解析度裝置使用。此 API 可供客戶端查詢上述影像的轉換進度，以及該原始影像檔可產出的幾種解析度檔案列表。

※ 相關資訊請參考本文中的「影片格式轉換」。

該 API Output payload 中的<progressstate>、<abstractstate>為「轉檔抽象進度」及「轉檔進度常數」，代表的是伺服端回覆客戶端轉檔進度的結果。以下為轉檔進度常數值說明列表：

轉檔抽象進度	轉檔進度常數	轉檔進度常數說明
1	10 INITIAL	伺服端完成接收檔案
	20 QUEUEING	檔案進入排程
	30 PROCESSING	檔案處理中
-1	40 NOT_SUPPORTED	不支援的檔案格式
	50 FAILURE	檔案處理失敗
	60 TIME_OUT	處理檔案逾時
	70 FILE_TOO_LARGE	檔案超出允許處理的大小
	80 NOT_NEED_TO_CONVERT	檔案畫面高度小於轉檔參數高度不轉檔
0	0 SUCCESS	檔案處理成功

回傳的狀態碼(Status Code)：

0 Success，回傳轉檔進度或結果。

225 數值不在容許的定義域內(例如：fi 參數傳了空字串或非數字)。

999 General Error。

/webrelay/getvideoconvertprogress/
Input
Query String 之中各個參數以'&'分隔： **fi** = { 原始影片檔案的 File ID } **tk** = { token }
Output
1. 若上傳影片檔符合系統支援轉檔的格式會回覆下列訊息

```
<?xml version="1.0" encoding="utf-8"?>
<getvideoconvertprogress>
    <status>{ Status Code }</status>
    <video><!-- 此 element 可重覆多次 -->
        <type>{ 系統回傳的影片檔解析度常數 }</type><!-- 如果客戶端欲下載轉換過後的 MP4 影片檔，則須在下載時指定該參數值。相關資料請參考/webrelay/directdownload/的 cpt 參數 -->
        <progressstate>{ 轉檔進度常數 }</progressstate>
        <abstractstate>{ 轉檔抽象進度 }</abstractstate>
        <resolution>{ 伺服端轉換過後的解析度 }</resolution><!-- 如果原始檔解析度未經轉換，則此參數將不顯示 -->
    </video>
</getvideoconvertprogress>
```

2. 若上傳影片檔不符合系統支援轉檔的格式會回覆下列訊息

```
<?xml version="1.0" encoding="utf-8"?>
<getvideoconvertprogress>
    <status>{ Status Code }</status>
    <progressstate>{ 轉檔進度常數 }</progressstate>
    <abstractstate>{ 轉檔抽象進度 }</abstractstate>
</getvideoconvertprogress>
```

※ 系統支援轉檔的格式請參考 ffmpeg

下載檔案(/webrelay/directdownload/)

目的：此 API 供客戶端指定 File ID，以下載 ASUS WebStorage 上指定該 ID 的檔案。此 API 支援 HTTP Range Partial Download，也支援續傳。

※ 此 API 支援標準 HTTP Range Partial Download，詳情請參考 RFC 2616 HTTP Range Header。

回傳的 HTTP 狀態碼(HTTP Status Code)：

403 Authentication Fail。

416 若 Payload 中的參數 of 指定了一個小於 0 或大於檔案長度的 offset。

/webrelay/directdownload/
Input
URL rewrite 路徑參數： /{ token }/ Query String 之中各個參數以'&'分隔： **dis** = { sid } **fi** = { 要下載之 File ID } **pv** = [*0* | *1*] <!-- 是否需要預覽(preview)。0：直接下載(預設值) | 1：預覽。 --> **ve** = { 欲下載的目標檔案版本(version) }<!-- 選擇性欄位。預設值為 0。 --> **u** = { Response URL，指定發生錯誤時要重新導向的頁面 URL } **of** = { 指定由距離檔案開頭多少個 bytes 的 offset 開始下載 } <!-- 選擇性欄位，預設值為 0。 --> **fue** = [0 | 1] <!-- fue(Force URL Encode)：指定此參數為 1 時檔名會被強制進行 URLEncode --> **cpt** = { 指定下載的影片檔解析度常數 } <!-- 選擇性欄位。如果客戶端欲下載轉換解析度過後的 MP4 影片檔，則須指定該參數值。相關資料請參考 /webrelay/getvideoconvertprogress/的 type 參數 --> **rn** = { 將上傳之檔案的檔名重新命名(rename)為此參數指定的檔案名稱 } <!-- 選擇性欄位。此字串須依序以 Base64 及 URL encode 進行編碼。 例如：欲將下載檔案易名為 Naomi_PhotoShop_Tutorial.mov。 Raw flie name：Naomi_PhotoShop_Tutorial.mov

對 Raw flie name 的編碼步驟： 1. Base64：mFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg== 2. URLencode： TmFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg%3D%3D 故設： rn=TmFvbWlfUGhvdG9TaG9wX1R1dG9yaWFsLm1vdg%3D%3D ※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。-->
Output
回傳檔案。

取得圖片的縮圖(/webrelay/getresizedphoto/)

目的：為避免因使用者上傳的圖片解析度過大，下載費時而造成相片瀏覽時的速度過於緩慢，因此提供此 API，可對該圖片進行指定解析度的縮圖，以節省資料傳輸的時間。

※ 將各項輸入參數採用 URL rewrite 方式以產生靜態 URL 型式以便 Web Browser 將縮圖保留在 local cache 中。

※ 為了協助客戶端判斷是否需要更新 cache 中的縮圖，須將檔案的最後更新時間記錄在回傳的 HTTP Header **Last-Modified**，以及檔案的最後更新時間設定在 HTTP Header **ETag**(建議以 java.util.Date.getTime()的 long 整數值表示)。

例如(程式碼請參考註解 3)：

Last-Modified： Tue, 17 Nov 2009 07:13:19 GMT

ETag： "1258441999687"

回傳的 **HTTP 狀態碼(HTTP Status Code)**：

304 HttpServletResponse.SC_NOT_MODIFIED。

403 Authentication Fail。

500 General Error。

/webrelay/getresizedphoto/

Input

URL Structure：

https://{ webrelay server host }/webrelay/getresizedphoto/{ token }/{ parameters }.jpg?**dis**={ sid }&**ecd**=[*1*]

parameters 說明：

pfd = { 原始圖片的 File ID }

size = { width x height } <!-- 縮圖的尺寸(寬 x 高)。其中，「x」是英文字母小寫的 x -->

pv = [*0* | *1*] <!-- 0：下載附件(預設值) | 1：預覽 -->

Query String 中各個參數以'&'分隔：

dis = { sid }

ecd = [*1*] <!-- 系統固定參數，請填數字 1 -->

組成 parameters 的步驟：

1. 參數依序以逗點分隔：
   pfd={ Photo File ID },size={ width x height },pv=[*0* | *1*]
2. 將第 1 步驟得到的參數做 Base64 編碼

範例：

當 Token = test123, sid = 123, File ID = 12345

Step	內容	值
1.	將 Token 放入路徑	/test123/
2.	加入參數	pfd=123456,size=640x480,pv=1
3.	參數做 Base64 編碼	cGZkPTEyMzQ1NixzaXplPTY0MHg0ODAscHY9MQ==

4.	後綴 ".jpg"	cGZkPTEyMzQ1NixzaXplPTY0MHg0O DAscHY9MQ==.jpg
5.	加入 Query String 參數	?dis=123&ecd=1

URL 結果：

https://{ webrelay server host }/webrelay/getresizedphoto/test123/cGZkPTEyMzQ1NixzaXplPTY0MHg0ODAscHY9MQ==.jpg?dis=123&ecd=1

※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。

Output

若客戶端傳來的 HTTP Header：

1. 包含 If-Modified-Since 且該時間大於等於欲索取檔案的最後更新時間。
2. 包含 If-None-Match，且客戶端傳來的 If-None-Match 與 Etag 值(本 API 中設為檔案的最後更新時間，long 整數值格式)相同，則回傳 HTTP Status 304 (HttpServletResponse.SC_NOT_MODIFIED)。

回傳縮圖結果：

一、縮圖失敗時，則回傳給客戶端的 HTTP Content-Length 設為 0(讓客戶端可以顯示自身準備的圖示)。

二、縮圖成功即回傳指定的圖檔。

※ HTTP 相關資訊請參考 Hypertext Transfer Protocol -- HTTP/1.1。

由指定檔案萃取文字檔 (/webrelay/getfulltextcompanion/)

目的：為避免客戶端無支援指定文件格式的應用軟體(例如：MS Office、PDF…等)，提供此 API 用以萃取指定之檔案的內容為純文字、XML 或 HTML 格式回傳，以便不時之需。

※ 為了協助客戶端(通常是 Web Browser)判斷是否須更新其 cache 中的檔案，

須將檔案的最後更新時間記錄在回傳的 HTTP Header **Last-Modified**，以及將檔案的最後更新時間設定在 HTTP Header **ETag**(建議以 java.util.Date.getTime()的 long 整數值表示)。

例如(程式碼請參考註解 3)：

Last-Modified： Tue, 17 Nov 2009 07:13:19 GMT

ETag： "1258441999687"

Companion file type(此 API 之參數"k"的值)：

0 - plain text

1 - XML

2 - HTML

回傳的 HTTP 狀態碼(HTTP Status Code)：

304 HttpServletResponse.SC_NOT_MODIFIED。

403 Authentication Fail。

404 指定的 File ID 不存在。

417 未指定 File ID。

500 General Error。

/webrelay/getfulltextcompanion/
Input
URL Structure： https://{ webrelay server host }/webrelay/getfulltextcompanion/{ token }/{ parameters }.txt?**dis**={ sid }&**ecd**=[*1*] parameters 說明： **fi** = { 要下載之中間過程文字檔的原始檔案 File ID } **pv** = [*0* | *1*]<!-- 0：附件下載(預設值) | 1：預覽 -->

k = { companion file type。未指定此參數時預設值為 0，表示純文字檔 }
<!-- 選擇性欄位 -->

Query String 中各個參數以'&'分隔：

dis = { sid }

ecd = [*1*]<!-- 系統固定參數，請填數字 1 -->

組成 parameters 的步驟：

1. 參數依序以逗點分隔：
   fi={ 要下載之中間過程文字檔的原始檔案 File ID },**pv**=[*0* | *1*],**k**={ companion file type }
2. 將第 1 步驟得到的參數做 Base64 編碼

範例：

當 Token = test123, sid = 123, File ID = 12345

Step	內容	值
1.	將 Token 放入路徑	/test123/
2.	加入參數	fi=12345,pv=0,k=0
3.	參數做 Base64 編碼	Zmk9MTIzNDUscHY9MCxrPTA=
4.	後綴 ".txt"	Zmk9MTIzNDUscHY9MCxrPTA=.txt
5.	加入 Query String 參數	?dis=123&ecd=1

URL 結果：

https://{ webrelay server host }/webrelay/getfulltextcompanion/test123/Zmk9MTIzNDUscHY9MCxrPTA=.txt?dis=123&ecd=1

※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。

Output

回傳檔案。

※ 若來源檔是不支援的檔案格式則回傳預設文字「This companion file is broken because converted fail!」

擷取影片預覽圖(/webrelay/getvideosnapshot/)

目的：為輔助客戶端進行影片的預覽，此 API 會擷取指定影片檔第 10 秒的截圖回傳。

※ 為了協助客戶端(通常是 Web Browser)判斷是否須更新其 cache 中的檔案，須將檔案的最後更新時間記錄在回傳的 HTTP Header **Last-Modified**，以及將檔案的最後更新時間設定在 HTTP Header **ETag**(建議以 java.util.Date.getTime()的 long 整數值表示)。

例如(程式碼請參考註解 3)：

Last-Modified： Tue, 17 Nov 2009 07:13:19 GMT

ETag： "1258441999687"

回傳的 HTTP 狀態碼(HTTP Status Code)：

304 HttpServletResponse.SC_NOT_MODIFIED。

403 Authentication Fail。

404 指定的 File ID 不存在。

417 未指定 File ID。

500 General Error。

/webrelay/getvideosnapshot/
Input
URL Structure： https://{ webrelay server host }/webrelay/getvideosnapshot/{ token }/{ parameters }.jpg?**dis**={ sid }&**ecd**=[*1*] parameters 說明： **fi** = { 要下載之預覽影像圖檔的原始影片 File ID } **pv** = [*0* | *1*] <!-- 是否預覽(preview)。0：直接下載(預設值) | 1：預覽。 -->

Query String 中各個參數以'&'分隔：

dis = { sid }

ecd = [*1*] <!-- 系統固定參數，請填數字 1 -->

組成 parameters 的步驟：

1. 參數依序以逗點分隔：

   fi={ 要下載之預覽影像圖檔的原始影片 File ID },**pv**=[*0* | *1*].jpg

2. 將第 1 步驟得到的參數做 Base64 編碼

範例：

當 Token = test123, sid = 123, File ID = 12345

Step	內容	值
1.	將 Token 放入路徑	/test123/
2.	加入參數	fi=12345,pv=0
3.	參數做 Base64 編碼	Zmk9MTIzNDUscHY9MA==
4.	後綴 ".jpg"	Zmk9MTIzNDUscHY9MA==.jpg
5.	加入 Query String 參數	?dis=123&ecd=1

URL 結果：

https://{ webrelay server host }/webrelay/getvideosnapshot/test123/Zmk9MTIzNDUscHY9MA==.jpg?dis=123&ecd=1

※ 若您使用的開發語言為 Ruby，請用 Base64.strict_encode64，避免“\n”在編碼之後出現。

Output

回傳檔案。

※ 若取得 snapshot 失敗，則回傳預設圖片 。

系統資料夾 ID 列表

類別	名稱	Folder ID
system.{ package }.home.root	我的收藏	0
system.backup.root	我的備份	-3
system.sync.root	同步中心	-5

※ 「我的收藏」資料夾：

{ package }是從 AcquireToken API 所回傳的 XML Payload 中取得，為 package 參數中的 display 參數值(請參閱文件「ServiceGateway Technical Spec」)。

服務區域 ID 列表

服務區域 ID	服務區域
1	台灣
2	美國
3	大陸

回傳的 HTTP 狀態碼(HTTP Status Code)

HTTP Status Code	Description
304	HttpServletResponse.SC_NOT_MODIFIED
403	Authentication Fail
404	指定的 File ID 不存在
416	若 Payload 中的參數 of 指定了一個小於 0 或大於檔案長度的 offset
417	未指定 File ID
500	General Error

回傳的狀態碼(Status Code)

Status Code	Description
0	Success
2	Authentication Fail
3	Payload is not validate
5	Developer Authorization Fail
211	檔案名稱為空白
213	檔案名稱長度超過限制
214	檔案名稱重複
218	要處理的目錄不存在或已刪除
219	檔案不存在或已刪除
220	一般檔案錯誤
221	單一檔案超過大小限制
224	用戶之儲存空間已用盡
225	數值不在容許的定義域內
226	用戶的帳號已被凍結(FROZEN)或關閉(CLOSE)
228	若客戶端下載中途發生斷線
250	輸入參數的檔案 signature(checksum)為空白或與伺服端不相符
251	Transaction ID 不存在
252	Transaction ID 與 File ID 不符
999	General Error

註解

1. 屬性(Attribute)

用以儲存檔案／目錄描述資訊，亦可加入各 App 對檔案／目錄獨有的資訊，以 XML 型式儲存，至少須包含以下三個內容：

(1). creationtime ：建立時間

(2). lastaccesstime：最後一次存取時間

(3). lastwritetime ：最後一次變更時間

以上三個時間皆是從 1970 年至今的秒數。

Ex：

<creationtime>1313054123</creationtime><lastaccesstime>1313054123</lastaccesstime><lastwritetime>1313054123</lastwritetime>

※ 由於 Attribute 內容，除了上述三個時間外，可能因不同應用而加上不同的內容。於修改內容時，請僅針對三個時間，或是您自訂的內容做修改，其他的請務必保留。

2. Checksum

用以辨識伺服端與客戶端的檔案是否為同一個檔案的特徵值字串，以 SHA 512 演算法實作。

3. 日期格式程式碼參考

SimpleDateFormat sdf = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss zzz", Locale.US);

sdf.setTimeZone(TimeZone.getTimeZone("GMT"));

System.out.println(sdf.format(檔案最後更新時間));