

## ※ 取得開發金鑰(Sid 及 ProgKey)

在您開始呼叫使用創意雲 API 之前，請您先於創意雲網站申請開發金鑰(Sid 及 ProgKey)。

## ※ 認證使用者帳號密碼

如欲進行使用者帳號密碼認證，客戶端必須先透過呼叫 ServicePortal 的 /member/requestservicegateway 取得本身所屬之 Service Area 的 ServiceGateway 位址。接著，客戶端需要帶著使用者的帳號(User ID)和密碼(Password)，透過 ServicePortal 取得的 ServiceGateway 位址進行/member/acquiretoken/呼叫。

1. 將客戶端傳來的使用者帳號密碼進行認證。
2. 若帳號密碼無誤，則 ServiceGateway 核發的 Token 將伴隨著 InfoRelay、WebRelay 的位址回傳給客戶端。

### 一、取得使用者帳號的服務區(/member/requestservicegateway/)

首先，請於 RequestServicegateway API 的 payload()中，輸入參數完成 RequestServicegateway API 的 Input Payload。

```
#import "GDataXMLNode.h"
+ (GDataXMLElement*) payload
{
    GDataXMLElement* root = [GDataXMLElement
elementWithName:@"requestservicegateway"];
    [root addChild:[GDataXMLElement elementWithName:@"userid"
stringValue:self.userid]];
    [root addChild:[GDataXMLElement elementWithName:@"password"
stringValue:self.password]];    [root addChild:[GDataXMLElement
elementWithName:@"language" stringValue:self.language]];    [root
addChild:[GDataXMLElement elementWithName:@"service"
stringValue:self.service]];
    [root addChild:[GDataXMLElement elementWithName:@"time"
stringValue:[NSString stringWithFormat:@"%qi", self.time]]];
```

```

return root;
}

```

接著，將指定連結的伺服器及組建好的 Input Payload 以 HTTP POST 的方式送出 request。

```

NSURL * url = [NSURL URLWithString: @"https:// sp.yostore.net
/member/requestservicegateway/"];
ASIDataRequest* request = [ASIDataRequest requestWithURL:url];
[request setValidatesSecureCertificate:NO];
[request setRequestMethod:@"POST"];

// POST Query
[request addRequestHeader:@"Cookie" value:cookieStr];

[request appendPostData:[data XMLData]]; // data為Input Payload
[request startSynchronous];

```

特別需要注意的是，開發者須於 header 中輸入 SID 的 Cookie，而且此 Cookie 的「sid」必須為小寫英文字母。

```

NSString* cookieStr = [NSString
stringWithFormat:@"sid=%@;c=0;v=2.0.;x-v=;EEE_MANU_Maunfactory=Apple;EEE
_PROD_ProductModal=iPhone;OS_VER=4;", sid];

```

最後，若 Output Payload 的 status 值為 0，表示 API 呼叫成功，客戶端方能取得 ServiceGateway 位址的值。

```

+ (RequestServiceGatewayResponse*) modelFromXMLData:(NSData*) data
{
    GDataXMLDocument* doc = [[[GDataXMLDocument alloc] initWithData:data
options:0 error:&error] autorelease];
    GDataXMLElement* root = [doc rootElement];
    RequestServiceGatewayResponse* model =
    [[[RequestServiceGatewayResponse alloc] init] autorelease];

```

```

NSArray* child;
child = [root elementsForName:@"status"];
if ( child && child.count )
{
    model.status = [[[child objectAtIndex:0] stringValue] intValue];
}
child = [root elementsForName:@"servicegateway"];
if ( child && child.count )
{
    model.servicegateway = [[child objectAtIndex:0] stringValue];
}
child = [root elementsForName:@"time"];
if ( child && child.count )
{
    model.time = [[[child objectAtIndex:0] stringValue] longLongValue];
}
return model;
}

```

## 二、使用者帳號密碼身份認證(/member/acquiretoken/)

取得 ServiceGateway 位址後，下一步驟為呼叫 AcquireToken API。將 ServiceGateway 位址加上 API 字串(/member/acquiretoken/)以 HTTPS 的方式帶入 cURL 的 CURLOPT\_URL 參數中，同樣地使用 HTTP POST 的方式送出 cURL request 以取得 HTTP response。

其中，須特別留意的是 AcquireToken 的認證機制。其認證機制須將 Header 以 Authorization 的方式組成再發送。而 Authorization Header 相關細節請參「帳號認證」選單目錄下的「使用者帳號密碼身份認證 (/member/acquiretoken/)」：<https://creative.asuscloud.com/?p=notes>。

待客戶端接收 HTTP response 之後，進行解析 AcquireToken 的 Output Payload。若 Output Payload 的 status 值為 0，表示登入認證成功並且取得 Token、InfoRelay 及 WebRelay。建議客戶端節省登入所耗資源，可將 Token 儲存於 Cookie 中(Token

的期限為三天)。

為減少對 ServiceGateway 的存取，客戶端可以在首度取得 ServiceGateway、Token、InfoRelay 及 WebRelay 位址後便儲存在本地電腦，直到發生客戶端無法連上先前儲存的 ServiceGateway 時再重新向 ServiceGateway 詢問所在地的 Service Area 內的 Service Gateway 位址。

```
// responseRequestServiceGateway 回傳
if( response.status == 0 )
{
    [[NSUserDefaults standardUserDefaults] setObject: response.inforelay
forKey:@"_inforelay_"];
    [[NSUserDefaults standardUserDefaults] setObject: response.webrelay
forKey:@"_webrelay_"];
    [[NSUserDefaults standardUserDefaults] setObject:
response.servicegateway forKey:@"_servicegateway_"];
}
```

## ※ 瀏覽目錄(/folder/browse/)

凡是不直接更動檔案本身的各種操作皆由 InfoRelay 負責處理。例如：瀏覽目錄內容、檢視目錄的分享清單及搬移目錄……等。

當客戶端需要瀏覽指定目錄之下的目錄(Folder)或檔案(File)時，可呼叫 FolderBrowse(/folder/browse/) API。

在 FolderBrowse API 的 Input Payload 中有個欄位參數為 Token，客戶端可將儲存的 Token 值帶入，直至 Token 失效，才需要再進行一次 AcquireToken。

```
NSURL * ir_FolderBrowser = [NSURL URLWithString:[NSString
stringWithFormat:@"https://%@/folder/browse/", response.inforelay]];
```

若 Output Payload 的 status 值為 0，表示呼叫 FolderBrowse API 成功。依據客戶端輸入參數的條件，可瀏覽指定目錄下的資料。

## ※ 下載(/webrelay/directdownload/)

當客戶端有下載檔案的需求時，可呼叫 DirectDownload API。此 API 支援 HTTP Range Partial Download(詳情請參考 RFC 2616 HTTP Range Header)，也支援續傳。DirectDownload 所請求的伺服器為 WebRelay，而且以 Query String 的方式帶入各個參數(例如:sid、fi 和 pv……等)以組成 Input。

```
+ (NSURL*) wr_DirectDownload:(NSString*) token fileid:(NSString*) fileid
{
    return [NSURL URLWithString:[NSString stringWithFormat:
@"https://%@/webrelay/directdownload/%@/?fi=%@", token, fileid]];
}
```

當 WebRelay 接收到客戶端下載檔案的請求時，WebRelay 會先取得 AcquireToken API 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。

## ※ 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，請按照順序分為三大步驟進行：

## 一、 初始化串流檔案上傳／續傳作業(/webrelay/initbinaryupload/)

InitBinaryUpload API 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。

須先將 Form 的 enctype 指定為 multipart/form-data，Submit 之後才能取得。

```
ASIFormDataRequest * request = [ASIFormDataRequest requestWithURL:url];  
  
[request setData:uploadData withFileName:name  
andContentType:@"multipart/form-data" forKey:@"data"];
```

接著，在 payload()方法中，以 Query String 的方式帶入各個參數(例如：tk、na、pa……等)。假如，此次上傳為斷點續傳，則 tx 參數為前次上傳的 Transaction ID(可由上一次伺服器端回傳的 Payload 中取得)，若 Transaction ID 值不是 Null，則 Query String 需要再帶入 tx 參數，方能進行續傳。

Function payload()：

```
InitBinaryUploadRequest* request = [[[InitBinaryUploadRequest alloc] init]  
autorelease];
```

```
if(transid != nil )
```

```
request.tx=transid;
```

Function parse ()：

```
InitBinaryUploadResponse *response =  
[InitBinaryUploadResponse parseFromData:[request responseData]];  
  
response.transid;
```

## 二、 串流檔案上傳／續傳(/webrelay/resumebinaryupload/)

在呼叫 InitBinaryUpload API 成功後，方能進行 ResumeBinaryUpload，否則並不需要接續進行下一步驟。Streaming Upload 的三個 API 需要依照順序一一呼叫成功才能順利執行串流上傳。

ResumeBinaryUpload API 得到的 response 中可看到 status 值是否為 0，表示此步驟呼叫 ResumeBinaryUpload API 是否成功。

```
ResumeBinaryUploadRequest* request = [[[ResumeBinaryUploadRequest alloc]  
init] autorelease];  
request.token = [AWSModel singleton].token.token;  
request.sid = sid;  
request.tx = transid;  
request.igw = igw;  
  
NSURL *url = [APIURL wr_ResumeBinaryUpload:request];  
ASIDataRequest * asirequest = [ASIDataRequest  
requestWithURL:url];  
[asirequest setData:data withFileName:display  
andContentType:@"multipart/form-data" forKey:@"data"];  
[asirequest setUploadProgressDelegate:self];  
[asirequest startSynchronous];
```

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(/webrelay/finishbinaryupload/)

客戶端經由 ResumeBinaryUpload API 將檔案內容完全上傳完畢之後，請客戶端務必呼叫 FinishBinaryUpload API，並指定 Transactoin ID，以通知 WebRelay 檔案已上傳完畢。

```
FinishBinaryUploadRequest* request = [[[FinishBinaryUploadRequest alloc] init] autorelease];
request.token = [AWSModel singleton].token.token;
request.sid = sid;
request.tx = transid;
request.igw = igw;

NSURL *url = [APIURL wr_FinishBinaryUpload:request];

ASIHTTPRequest *asirequest = [ASIHTTPRequest requestWithURL:url];
[asirequest startSynchronous];
```

若客戶端未呼叫 FinishBinaryUpload API，則 WebRelay 會認為客戶端只是暫停上傳檔案，即未完成上傳。上傳完畢後，便可得到 File ID。

```
FinishBinaryUploadResponse *response = [FinishBinaryUploadResponse
parseFromData:[asirequest responseData]];

response.fileid
```