

S3-Link APIs for Amazon S3

Introduction

This guide is for REST APIs for Amazon S3. We have APIs for the below purposes.

1. Create Record Home Folder for Salesforce record
2. Upload File in Amazon S3
3. Download File from Amazon S3
4. Sync Record Home Folder
5. Create S3-Files in Salesforce with folder structure
6. Delete S3-Files from Salesforce and Amazon S3

We have Apex REST methods for the above operations in our managed package. Let's go through those methods.

Salesforce Authorization

You will have to authorize all below Salesforce callouts by passing the access token as a request parameter. Check the link below to get the access token using username and password. The value of "access_token" needs to be passed as the "Authorization" request parameter in all below methods.

https://help.salesforce.com/articleView?id=remoteaccess_oauth_username_password_flow.htm&type=5

Create Record Home folder for Salesforce record

The method will get the Salesforce record Id from the URL and create a folder structure AWS Bucket Name/Object Plural Label/Record Name for the Salesforce record.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File, S3-Folder, and object whose record id is passed in the URL. Make sure that the user has access to all the fields of both S3-File and S3-Folder objects.

- **Request Syntax**

GET HTTP/1.1

Host:

https://*instance*.salesforce.com/services/apexrest/NEILON/S3Link/v1/recordfolder/*recordId*

Authorization: Bearer *sessionId*

Notes

- Replace recordId with your Salesforce Record Id to attach file.
- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

The request doesn't require any request body

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 response. In the response, you will get the details of the record home S3-Folder created successfully for the Salesforce record.

- **Sample CURL**

Here is a sample CURL for this action.

curl

```
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v1/recordfolder/00190000027DPLT" -H "Authorization: Bearer 00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhihfW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP1t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1"
```

- **Sample CURL Response**

Here is a sample CURL response for this action.

```
{\"status\":\"OK\",\"message\":null,\"data\":{\"attributes\":{\"type\":\"NEILON__Folder__c\",\"url\":\"/services/data/v48.0/subjects/NEILON__Folder__c/a0P0I00001NKSkEQAX\"},\"NEILON__Amazon_File_Key__c\":\"Accounts/Greenwich Media\",\"NEILON__Bucket_Name__c\":\"testdemos3link\",\"NEILON__Category__c\":\"White papers\",\"Name\":\"Greenwich\"}}
```

```
Media\", \"NEILON__Bucket_Region__c\": \"us-west-2\", \"NEILON__Parent__c\": \"a0P0I00001KJccBUAT\", \"Id\": \"a0P0I00001NKSkEQAX\"}], \"code\": \"200\"}
```

Upload File in Amazon S3

This method will upload a file in Amazon S3 through Salesforce and attach those files with a Salesforce record. There is a limit of 6 MB per file. The method will get the Salesforce record Id from the URL and create a folder structure AWS Bucket Name/Object Plural Label/Record Name for the Salesforce record. After that, it will upload files in Amazon S3 in the above mentioned folder structure. It will also create the S3-File record in Salesforce and that S3-File record will be attached to the Salesforce record.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File, S3-Folder, and object whose record id is passed in the URL. Make sure that the user has access to all the fields of S3-File and S3-Folder objects.

- **Request Syntax**

POST HTTP/1.1

Host: [https://*instance*.salesforce.com/services/apexrest/NEILON/S3Link/v1/uploadfile/*recordId*](https://instance.salesforce.com/services/apexrest/NEILON/S3Link/v1/uploadfile/recordId)

Authorization: Bearer *sessionId*

filename: *fileName*

Content-Type: *contentType*

Body

Notes

- Replace recordId with your Salesforce Record Id to attach file.
- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request requires the following URI parameters.

1. filename

Name of the file. This is required. The file name cannot contain “?” or “/” special characters.

2. Content-Type

Content type of the file.

3. NEILON__Description__c

Description of the file.

4. NEILON__Category__c

Category of the file.

- **Request Body**

The request accepts binary data from a file that needs to be uploaded in Amazon S3.

Body

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 response. In the response, you will get a JSON of NEILON__File__c record created in Salesforce.

- **Sample CURL**

Here is a sample CURL for this action.

```
curl
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v1/uploadfile
/00190000027DPLT" -H "Authorization: Bearer
00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhifW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t
4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1" -H "filename: 3.jpg" -H
"Content-Type: image/jpeg" -H "NEILON__Category__c: Whitepapers" --data-binary @C://3.jpg
```

- **Sample CURL Response**

Here is a sample CURL response for this action.

```
{"status":"OK","message":null,"data":{"attributes":{"type":"NEILON__File__c"},"url":"/s
ervices/data/v48.0/subjects/NEILON__File__c/a090I00001NKSkEQAX"},{"NEILON__Amazon_Fil
e_Key__c":"Accounts/Greenwich
```

```
Media/3.jpg\", \"NEILON__Bucket_Name__c\": \"testdemos3link\", \"NEILON__Size__c\": 2045.0,
 \"NEILON__Access_Type__c\": \"Public Download
Only\", \"NEILON__Category__c\": \"Whitepapers\", \"NEILON__Presigned_URL_Frequency__c\":
 \"Every
Week\", \"Name\": \"3.jpg\", \"NEILON__Bucket_Region__c\": \"us-west-2\", \"NEILON__Extensio
n__c\": \".jpg\", \"NEILON__Folder__c\": \"a0P0I00001KJccBUAT\", \"Id\": \"a090I00001NKSskEQAX
\"}}\", \"code\": \"200\"}
```

Download File from Amazon S3

This method will download a file in Amazon S3 through Salesforce. There is a limit of 6 MB per file. The method will get the S3-File record Id from the URL and download that file from Amazon S3.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File, S3-Folder, and object whose record id is passed in the URL. Make sure that the user has access to all the fields of S3-File and S3-Folder objects.

- **Request Syntax**

GET HTTP/1.1

Host: <https://instance.salesforce.com/services/apexrest/NEILON/S3Link/v2/filecontent/recordId>

Authorization: Bearer *sessionId*

Body

Notes

- Replace recordId with your S3-File Record Id to download the file.
- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

The request doesn't require any request body

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 response. In the response, you will get base64 file content for the Amazon S3 file.

- **Sample CURL**

Here is a sample CURL for this action. Run this CURL from a location other than C:\ drive because sometimes because of your local machine permissions, files cannot be saved in C:\ drive. Below sample URL will download the file from AWS and create a File.pdf file in your local machine.

```
curl
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v2/filecontent/a012v00003IYm6s" -H "Authorization: Bearer
00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhifW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1" -o File.pdf
```

- **Sample CURL Response**

This request will not return any text response unless there is any error in the request.

Sync Record Home Folder

Method will get the Salesforce record Id from the URL and sync all files related to that record at AWS Bucket Name/Object Plural Label/Record Name.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File, S3-Folder, and object whose record id is passed in the URL. Make sure that the user has access to all the fields of both S3-File and S3-Folder objects.

- **Request Syntax**

GET HTTP/1.1

Host: <https://instance.salesforce.com/services/apexrest/NEILON/S3Link/v1/syncfolder/recordId>

Authorization: Bearer *sessionId*

Notes

- Replace recordId with your Salesforce Record Id Or S3-Folder Id to sync.
- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

The request doesn't require any request body

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 or HTTP 201 response. HTTP 201 if the folder will be synced asynchronously.

- **Sample CURL**

Here is a sample CURL for this action.

curl

```
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v1/syncfolder/00190000027DPLT" -H "Authorization: Bearer 00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhihfW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1"
```

- **Sample CURL Response**

Here is a sample CURL response for this action.

```
{  
  
  "status" : "OK",  
  
  "sObjects" : null,  
  
  "message" : null,  
  
  "data" : null,  
  
  "code" : "200"  
  
}
```

Create S3-Files in Salesforce with folder structure

If files will be uploaded in Amazon S3 through Salesforce, we create S3-Folders and S3-Files custom object records to store metadata of Amazon S3 files and Salesforce users can access Amazon S3 files using those S3-File records in Salesforce. But if files are uploaded in Amazon S3 outside of Salesforce, you will have to create those S3-Folders and S3-Files records manually or programmatically in order to allow your Salesforce users to access those files. We have REST APIs which you can call from anywhere outside of Salesforce to create S3-Files and S3-Folders records.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File and S3-Folder objects. Make sure that the user has access to all the fields of both S3-File and S3-Folder objects.

- **Request Syntax**

POST HTTP/1.1

Host: [https://*instance*.salesforce.com/services/apexrest/NEILON/S3Link/v1/creates3files/](https://instance.salesforce.com/services/apexrest/NEILON/S3Link/v1/creates3files/)

Authorization: Bearer *sessionId*

Body

Notes

- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

The request accepts the binary data of JSON with information about Amazon S3 files. Below is sample data.

Body

```
[
  {
    "Name": "4.jpg",
```



```

        "NEILON__Bucket_Name__c":"testdemos3link",
        "NEILON__Amazon_File_Key__c":"Accounts/Greenwich Media/4.jpg",
        "NEILON__Size__c":563425,
        "NEILON__Account__c":"0012v00002BaKVjAAN"
    }
]

```

- **Limitations**

- File name and folder names you are passing in the NEILON__Amazon_File_Key__c cannot contain special characters ? and /
- File name and folder names you are passing in the NEILON__Amazon_File_Key__c must not exceed 80 characters.

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 response. In the response, you will get a JSON of the list of NEILON__File__c records created in Salesforce.

- **Sample CURL**

Here is a sample CURL for this action.

```

curl
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v1/creates3files/" -H "Authorization: Bearer
00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhihfW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1" -H "Content-type:
application/json" -d @C://FilesData.txt

```

- **Sample CURL Response**

Here is a sample CURL response for this action.

```

{"status":"OK","message":null,"data":[{"attributes":{"type":"NEILON__File__c","url":"/services/data/v48.0/subjects/NEILON__File__c/a090i00001NKSkEQAX"},"NEILON__Amazon_File_Key__c":"Accounts/Greenwich Media/4.jpg"},"NEILON__Bucket_Name__c":"testdemos3link"},"NEILON__Size__c":2045.0,"NEILON__Access_Type__c":"Public Download Only"},"NEILON__Category__c":"Whitepapers"},"NEILON__Presigned_URL_Frequency__c":"Every Week"},"Name":"4.jpg"},"NEILON__Bucket_Region__c":"us-west-2"},"NEILON__Extension__c":".jpg"},"NEILON__Folder__c":"a0P0i00001KJccBUAT"},"Id":"a090i00001NKSkEQAX"},"code":"200"}

```

Delete S3-Files in Salesforce and Amazon S3

If files will be deleted in Salesforce, we delete those files from Amazon S3 as well. But if files are deleted directly in Amazon S3, you will have to delete those S3-Files records manually or programmatically from Salesforce. We have REST apis which you can call from anywhere outside of Salesforce to delete S3-Files records.

- **Permissions**

You will need an API user in your Salesforce org. That API user will need View All and Modify All permission of S3-File. Make sure the user has access to all the fields of both objects

- **Request Syntax**

POST HTTP/1.1

Host: [https://*instance*.salesforce.com/services/apexrest/NEILON/S3Link/v1/deletes3files/](https://instance.salesforce.com/services/apexrest/NEILON/S3Link/v1/deletes3files/)

Authorization: Bearer *sessionId*

Body

Notes

- Replace sessionId with your Salesforce Session Id.
- Replace instance with your Salesforce server URL.

- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

The request accepts the binary data of a list of AWS file paths that need to be deleted. Below is sample data. Here "testdemos3link" is the Amazon S3 bucket name.

Body

```
[  
  
  "testdemos3link/Accounts/Greenwich Media/4.jpg"  
  
]
```

- **Response Elements**

If the action is successful, the service sends back an HTTP 200 response. In the response, you will get the AWS file paths whose S3-Files records are successfully deleted.

- **Sample CURL**

Here is a sample CURL for this action.

```
curl
"https://s3linktestorg-dev-ed.my.salesforce.com/services/apexrest/NEILON/S3Link/v1/deletes3files/" -H "Authorization: Bearer
00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhifW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP1t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "X-PrettyPrint: 1" -H "Content-type:
application/json" -d @C://DeleteFilesData.txt
```

- **Sample CURL Response**

Here is a sample CURL response for this action.

```
{"status":"OK","message":null,"data":["testdemos3link/Accounts/Greenwich
Media/4.jpg"],"code":"200"}
```