

S3-Link APIs for Amazon S3

Introduction

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

1. Create Record Home Folder for Salesforce record
2. Upload File in Amazon S3 through Apex
3. Create S3-Files in Salesforce with folder structure
4. 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.

Create Record Home folder for Salesforce record

Method will get the Salesforce record Id from the URL and create folder structure AWS Bucket Name/Object Plural Label/Record Name for 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*](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 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/recordfolder/00190000027DPLT" -H "Authorization: Bearer 00D28000001F3JL!ARUAQD9_gSnqoLyMhpEhihfW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP"
```

- **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/objects/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 through Apex

This method will upload a file in Amazon S3 through Salesforce and attach those files with Salesforce record. There is a limit of 11 MB per file. Method will get the Salesforce record Id from the URL and create folder structure AWS Bucket Name/Object Plural Label/Record Name for Salesforce record. After that it will upload files in Amazon S3 in the above mentioned folder structure. It will also create a S3-File record in Salesforce and that S3-File record will be attached to 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.

- **Request Body**

The request accepts binary data of file which 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_gSnqoLyMhpEhifW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -H "filename: 3.jpg" -H "Content-Type: image/jpeg" -d
@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":"/services/data/v48.0/subjects/NEILON__File__c/a090I00001NKSkEQAX"},"NEILON__Amazon_File_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__Extension__c":".jpg"},"NEILON__Folder__c":"a0P0I00001KJccBUAT"},"Id":"a090I00001NKSkEQAX"},"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 meta data of Amazon S3 files and Salesforce users can access Amazon S3 files using those S3-File records in Salesforce. But if files 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 object. 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.

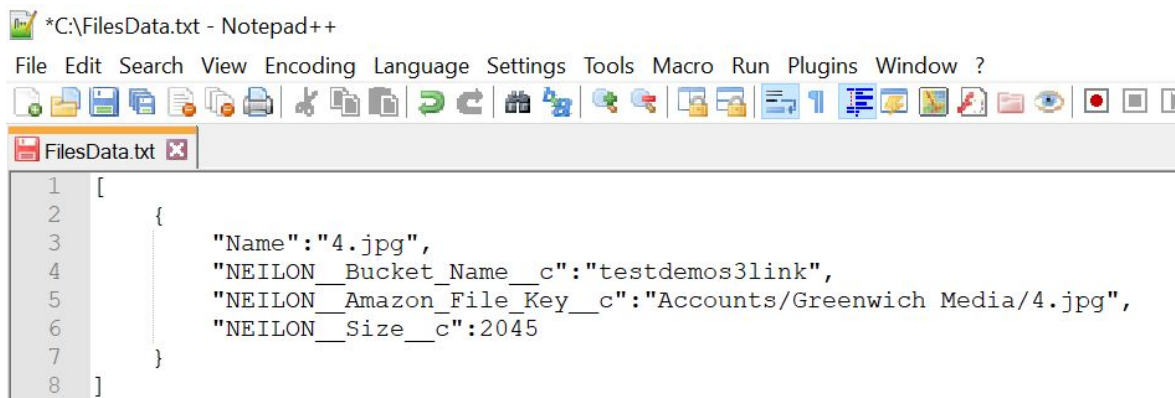
- **URI Request Parameters**

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



```
*C:\FilesData.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
FilesData.txt
1 [
2   {
3     "Name": "4.jpg",
4     "NEILON__Bucket_Name__c": "testdemos3link",
5     "NEILON__Amazon_File_Key__c": "Accounts/Greenwich Media/4.jpg",
6     "NEILON__Size__c": 2045
7   }
8 ]
```

- **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" -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/objects/NEILON__File__c/a090I00001NKSsEQAX"},
      "NEILON__Amazon_File_Key__c": "Accounts/GreenwichMedia/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": "a090I00001NKSsEQAX"}
  ],
  "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 that 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/>

Authorization: Bearer *sessionId*

Body

Notes

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

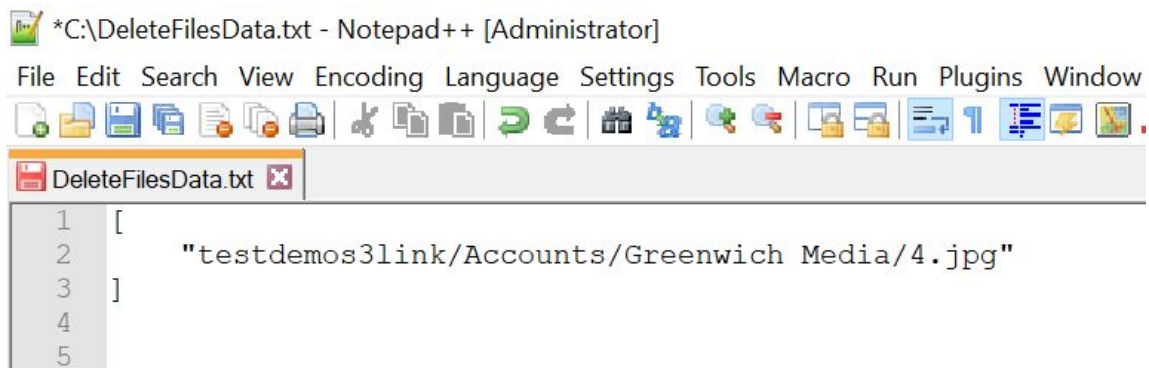
- **URI Request Parameters**

The request doesn't require any URI parameters.

- **Request Body**

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

Body



The screenshot shows a Notepad++ window titled "*C:\DeleteFilesData.txt - Notepad++ [Administrator]". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations. The active window is "DeleteFilesData.txt" and the content is a JSON array:

```
1 [
2   "testdemos3link/Accounts/Greenwich Media/4.jpg"
3 ]
4
5
```

- **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
```

00D2800001F3JL!ARUAQD9_gSnqoLyMhpEhifW0ajibDXDjUO6MFYFAMF0zBsX7uku5KT8UP11
t4J.I5Y9Zp2zYFMSUi8_BSVTLFptfW14gNLP" -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"}
```