Amazon Web Services CreateLinkedAccount API

Account creation and linking is currently a multi-step manual process. Using the CreateLinkedAccount API enables customers to create new accounts programmatically and associate them with a paying account. This API is used when there is a need to create new linked accounts frequently or in large numbers. The CreateLinkedAccount API is offered at no additional charge.

Overview

There are two parts to creating and linking a new account. The synchronous API called by the client and an asynchronous workflow that is started by the API.

The API is called using Signature v4 (<http://docs.aws.amazon.com/general/latest/gr/signature-version-4.html>) signing with valid AWS IAM (<https://aws.amazon.com/iam/>) user credentials and takes an email address, name, mailing address, phone number, access role name, and an AWS SNS topic ARN (<http://aws.amazon.com/sns/>) as input. The API does some shallow validation of the parameters. It then posts a start message to the SNS topic and starts the workflow.

The workflow creates a new linked account using the email passed into the call as the new linked account's email. It populates the rest of the account with the other information specified. When the workflow completes it will post a message to the SNS topic describing the results. If the workflow completes successfully, the new account will be linked to the payer account. The new linked account will be subscribed to the standard AWS product package that any new account receives.

**Important**

* The API assigns a random password to the new account and does NOT provide that password.
  + In order to get root login access, the account owner will need to go to <https://aws.amazon.com/> and begin a password recovery by attempting login and selecting “I forgot my password.”
  + The API will also create a role in the new account that grants admin access to the account. The role will be set up so that the payer account has permission to access this role.
* This API only creates new accounts. It cannot be used to link already existing accounts.

Requirements

* An eligible paying account that is signed up for Consolidated Billing.
* The payer account must not be at the maximum number of allowed linked accounts
* An IAM user in the payer account with an access policy allowing the user to use the API
* A SNS topic you control in the US East region
  + This service’s account (710052904269) needs permission to publish to this topic

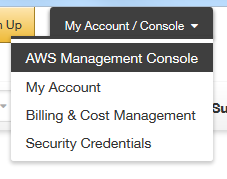
Getting Started

* [Create an SNS topic and whitelist our service account](#SNS)
* [Use our Java SDK](#SDK) or [write your own script with Python or AWS CLI](#CLI)

Whitelisting CreateLinkedAccount to post to an SNS topic

This section shows how to whitelist the account used by this API so that it can post to an SNS topic. This SNS topic does not have to be owned by the payer account the new linked account will be associated with.

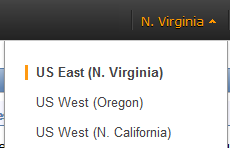
* Go to [https://aws.amazon.com](https://aws.amazon.com/) and log in the AWS Management Console in the upper right



* Click on SNS Push Notification Service



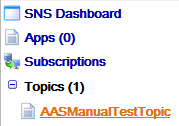
* Switch to the US East region in the upper right



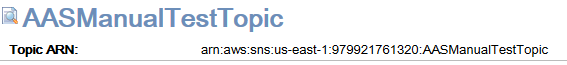
* If you need a new topic, click the Create New Topic button and fill in the form

A description...

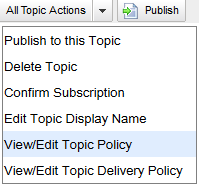
* In the left hand window, expand the Topics category and click your topic



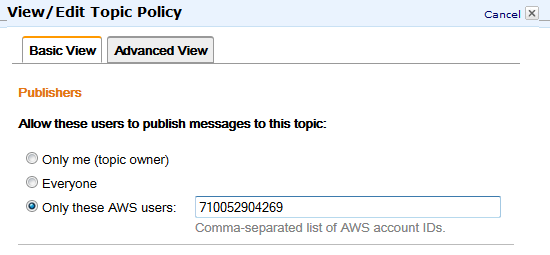
* In the right hand window it will show the ARN for your topic. This is the parameter you will provide to the CreateLinkedAccount API.



* Above that, click the “All Topic Actions” dropdown and select “View/Edit Topic Policy”



* In the Publishers area, select the radio button for “Only these AWS users:” and fill in the field with account 710052904269



* Click the “Update Policy” button and your SNS topic is ready to be used by the API

A description...

Using the AWS Command Line Interface (CLI) or Python SDK (Boto3)

To be able to use CreateLinkedAccount API through the AWS CLI or Python SDK, you need to first setup the AWS CLI or Python SDK on your machine and add CreateLinkedAccount API model into your CLI environment.

**Installing the AWS CLI/Python SDK:** If you don’t have AWS CLI installed, follow the procedure as below to install version 1.9.12 or higher.

<http://docs.aws.amazon.com/cli/latest/userguide/installing.html>

If you need to call the CreateLinkedAccount API with Python SDK, you also need to install the Python SDK by following the procedure as below:

<https://aws.amazon.com/sdk-for-python/>

**Adding CreateLinkedAccount model into your CLI environment:** Put the CreateLinkedAccount model JSON file under the current directory, and run the following CLI command to add CreateLinkedAccount API model to your CLI environment:

|  |
| --- |
| $ **aws configure** add-model --service-mode file://awsaccountsservice-2016-01-07.normal.json --service-name AWSAccountsService |

You will then have a command “AWSAccountsService” that is associated with the CreateLinkedAccount service model.

**Configuring the AWS CLI with your payer account secret key:** Run “aws configure” command to setup your AWS CLI with your payer account access Key ID and secret access key.

|  |
| --- |
| $ **aws configure**  AWS Access Key ID [None]: ***AKIXZOSFODNN7EXAMPLE***  AWS Secret Access Key [None]: ***wJalsXUdnFEMI/K7MDENG/bPxPfiCZEXAMRLEKEY***  Default region name [None]: ***us-east-1***  Default output format [None]: ***json*** |

**Calling CreateLinkedAccount with the AWS CLI:**

The easiest way to call CreateLinkedAccount with the AWS CLI is to put all the input parameters of the API into a JSON file:

|  |
| --- |
| {"email": "newLinkedAccountEmail@myDomain.com",  "address":  {"firstName": "John",  "lastName": "Smith",  "mailingAddress":  {"addressLine1": "123 4th Ave",  "addressLine2": "thisLineOptional",  "addressLine3": "thisLineOptional",  "city": "Seattle",  "stateOrRegion": "Washington",  "postalCode": "98101-1210",  "countryCode": "US"},  "phoneNumber": "1234567890"},  "snsArn": "arn:aws:sns:us-east-1:453272712372:ExampleSNSTopic",  "roleName": "PayerAccountAccessRole"} |

Then specify its location it in your command line:

|  |
| --- |
| $ **aws** AWSAccountsService create-linked-account --cli-input-json file://inp.json |

**Calling CreateLinkedAccount with the AWS Python SDK (Boto3):**

Below is sample code showing how to use the Python SDK to call the CreateLinkedAccount API.

|  |
| --- |
| **import** boto3  *# Setup AWSAccountsService client*  client = boto3.client(service\_name='AWSAccountsService')  *# Email address for the new AWS account*  newAccountEmail='newLinkedAccountEmail@myDomain.com'  *# Remember that the service's account needs permission to post to this topic*  notificationSNSArn='arn:aws:sns:us-east-1:453272712372:ExampleSNSTopic'  *# Set the name of the IAM role the payer will use to access the account*  IAMRoleName='PayerAccountAccessRole'  newAccountAddress={  "firstName": "John",  "lastName": "Smith",  "mailingAddress": {  "addressLine1": "1918 8th Ave",  "addressLine2": "thisLineOptional",  "addressLine3": "thisLineOptional",  "city": "Seattle",  "stateOrRegion": "Washington",  "postalCode": "98101-1210",  "countryCode": "US"  },  "phoneNumber": "1234567890"  }  res=client.create\_linked\_account(email=newAccountEmail, address=newAccountAddress, snsArn=notificationSNSArn, roleName=IAMRoleName)  **print**(res) |

Using the Java SDK

The easiest way to start with the SDK is to use the AWS Toolkit for Eclipse:

<https://aws.amazon.com/sdk-for-java/>

Once this is installed, you can start a new AWS project which will automatically include the public AWS SDK and all dependencies. You should then add the CreateLinkedAccount SDK to your build path. Below is sample code showing how to use the SDK to call the CreateLinkedAccount API.

|  |
| --- |
| //All undeclared parameters are String objects that are  //initialized before this example  //First we instantiate an object to hold our IAM user credentials  BasicAWSCredentials creds = **new** BasicAWSCredentials(*ACCESS\_KEY*, *SECRET\_KEY*);  //Now we create our handle to the service and give it the credentials  //we want to use  AmazonAccountsServiceClient aas = **new** AmazonAccountsServiceClient(creds);  aas.setServiceNameIntern("AWSAccountsService");  //Now we need to create the parameters we are going to call the API with  //We instantiate a MailingAddress object and set the data in it  MailingAddress mailingAddress = **new** MailingAddress();  //You can also call .setAddressLine2 and .setAddressLine3 if you need to  mailingAddress.setAddressLine1(*ADDRESS\_LINE\_1*);  mailingAddress.setCity(*CITY*);  mailingAddress.setStateOrRegion(*STATE*);  mailingAddress.setCountryCode(*COUNTRY*);  //Remember that the postal code must be the full code (5+4)  mailingAddress.setPostalCode(*POSTAL\_CODE*);    //Now we instantiate our contact address which includes the  //mailing address, our name, and our contact phone number  Address address = **new** Address();  address.setFirstName(*FIRST\_NAME*);  address.setLastName(*LAST\_NAME*);  address.setMailingAddress(mailingAddress);  address.setPhoneNumber(*PHONE\_NUMBER*);    //Now let's create the request we are going to pass to the service  CreateLinkedAccountRequest ClaRequest = **new** CreateLinkedAccountRequest();  //And populate the data in that request  ClaRequest.setAddress(address);  ClaRequest.setEmail(*EMAIL*);  //Remember that the service's account needs permission to post to this topic  ClaRequest.setSnsArn(*SNS\_TOPIC*);  // Set the name of the IAM role the payer will use to access the account  ClaRequest.setRoleName(*ROLE\_NAME*);  //And finally let's make the call to the service. This will throw  //an exception if the API is not successful, otherwise you can  //assume the workflow was started  aas.createLinkedAccount(ClaRequest); |

SDK Exceptions

**AmazonServiceException**

<http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/AmazonServiceException.html>

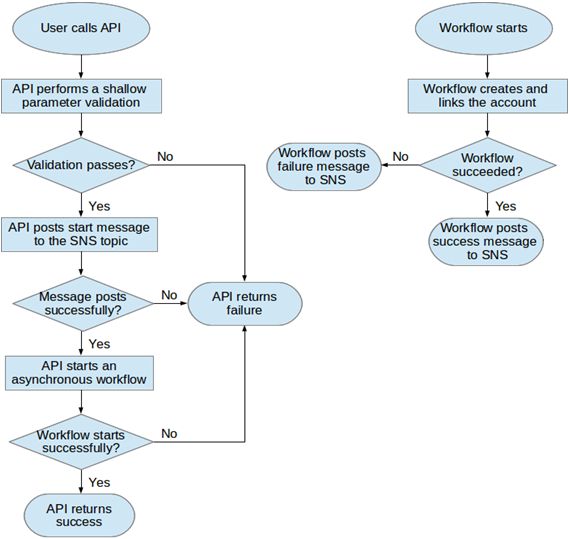
This is a catch-all for all service-side exceptions (e.g. the API exceptions).

**AmazonClientException**

<http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/AmazonClientException.html>

This is a catch-all for any AWS exception that could be thrown including client-side issues such as no available network connection.

Process Flow for API and Workflow



Inputs

**email**

This should be the email you would normally use to create a new account. The new account will be bound to this email. The provided email must not already be associated with an AWS or Amazon retail account. No longer than 64 characters. You will need access to this email address in order to recover the password for the new linked account and use it.

**firstName/lastName**

The first and last name to be associated with the new linked account. Each name cannot be longer than 256 characters.

**mailingAddress**

Mailing/contact address to be associated with the new linked account. The second and third address lines are optional. The postal code must be a nine-digit zip code (#####-####). All country codes are two characters. If your address is in the United States, you should use ‘US’ as your country code. One way to check if the address is valid before calling the API is to log in to an Amazon.com retail account and add an address. The resulting address added is an entry our system should accept.

**phoneNumber**

Phone number for the new linked account. This should be the raw string of numbers with no additional formatting or special characters.

**snsArn**

This should be the ARN for an SNS topic created in US East that you control. The API and workflow will post messages to this topic about the status of the account creation and linking process. Posts to the topic are made using the services AWS account 710052904269. This account needs permission to publish to the SNS topic. There is a section in this document that describes granting the required permission.

**roleName (optional)**

The API will create a role in the new linked account that has admin access. This role will be accessible by the payer. The role will be given this parameter as its name if this is provided. If not provided, the role will be given “PayerAccountAccessRole” as its default name.

Authorization/Authentication

This API requires HTTPS Signing v4. You should use IAM credentials from the paying account. The IAM access policy for using this API is below.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "AWSAccountsService:CreateLinkedAccount",

"Resource": "\*"

}

]

}

Output

The API will respond according to Hypertext Transfer Protocol (HTTP) response definitions. It will give an HTTP 200 response if your request was valid. Otherwise it will response with an HTTP error code and an exception descriptive of the cause. If you got an HTTP 200, then a start message has been posted to your SNS topic and the workflow has been started. Upon workflow completion, one of two messages will be posted to the SNS topic indicating if the workflow was successful.

API Exceptions

**UnauthorizedRequestException**

The credentials you used to call the API are not tied to a payer account. These credentials belong to a regular account.

**InvalidParameterException**

Indicates that a request parameter does not comply with the associated constraints.

**LinkedAccountLimitExceededException**

Your payer account has reached the maximum number of accounts that can be linked to it. You can contact customer service to request a limit increase.

**InternalErrorException**

Indicates an internal AWS error.

SNS Messages

There are three types of messages posted to the SNS topic: starting the workflow, workflow failure, and workflow success. Successful calls to the API will have a start message and one result message. The body of these messages is in JSON format.

**API starting the workflow**

Subject: CreateLinkedAccount - Starting

Body:

{"email":"yourEmailParameter","info":"LinkedAccountCreationStarted"}

**Workflow succeeded**

Subject: CreateLinkedAccount – Success

Body: {"linkedAccountEmail":"yourEmailParameter", "linkedAccountId":"############", "accessRoleName":"nameOfTheCreatedRole"}

**Workflow failed**

Subject: CreateLinkedAccount - Failed

Body:

{“linkedAccountEmail”:”yourEmailParameter”,”exception”:”exceptionThatCausedFailure”}

Workflow exceptions

The following are the possible exceptions that can cause a workflow failure.

**UnauthorizedRequestException**

The account your IAM credentials are under is not marked as a payer account. These credentials belong to a regular account.Call customer service if you are certain the IAM credentials are tied to a payer account.

**LinkedAccountLimitExceededException**

You have reached the maximum number of accounts that can be linked to your payer account. You can call customer service to increase this limit.

**DuplicateAccountException**

The email you are using is already registered with Amazon.

**InternalErrorException**

Indicates an internal AWS error.

FAQ

**Why do I get an error that says it cannot find my SNS topic even though I can see it in my console?**

Make sure you have created your topic in the us-east-1 region. The API cannot find topics in any other region.

**Should I retry calls?**

No. All of the error cases will result in the same error case if you retry. Correct your input first before retrying. If there is a network problem and you did not get a response, it is safe to retry.

**I always get an error saying “Unable to validate Address”. How can I fix this?**

This means that the mailing address you are providing is not specific enough. We can't offer you suggestions to narrow down your address so this API needs a very specific address that has no ambiguity. You should fix this problem by creating one account manually with the address you want to use and using one of the suggestions provided by the AWS Portal.

**I need to create more linked accounts but my payer account has reached its limit. What should I do?**

You can call Customer Service and request to have your limit increased.

**When is my new linked account ready to be used?**

As soon as you receive the success message from the workflow the new linked account is ready. The can take varying amounts of time depending on the current load our systems are experiencing. The workflow will also retry operations so failing workflows will take longer to resolve. Normally, the workflow should take only a few minutes.

**How can I maintain control over the linked account?**

It is recommended that you do not give the root account credentials to your customer. Instead, you should keep those and create IAM users for your customers to use. You can apply policies to these IAM users that restrict what resources, actions, and services they have access to. IAM is part of the standard AWS SDK which allows you to set up these users programmatically.

Overview of IAM: <http://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html>

IAM API Reference: <http://docs.aws.amazon.com/IAM/latest/APIReference/Welcome.html>