

## Uploading Files to S3 Using API Gateway via Lambda Function:

[] Create a s3 bucket.

### Steps to Create to Lambda Function:

[] Navigate to lambda console

[] Create an AWS Lambda function with Python 3.10.

[] Once lambda is created. Add Below Code and Deploy.

[] On the Configuration tab, choose General configuration, and then choose Edit. Set Timeout to 10 seconds and then choose Save

```
import json

import boto3
import base64
import uuid
from datetime import datetime

def lambda_handler(event, context):
    # Initialize S3 client
    S3 = boto3.client("s3")

    # Get file content from event
    get_file_content = event["content"]

    # Decode the base64 encoded content
    decode_content = base64.b64decode(get_file_content)

    # Generate a unique file name using UUID and timestamp
    unique_file_name = f"data_{uuid.uuid4()}.txt"

    # Upload the file to S3 with the unique name
    s3_upload = S3.put_object(Bucket="filesuploding", Key=unique_file_name,
Body=decode_content)

    # Return success message with the new file name
    return {
        'statusCode': 200,
        'body': json.dumps(f'The Object {unique_file_name} is uploaded successfully!')}
```

```
}
```

**[] delete the previous one and update to a new one.**

```
import json
import boto3
import base64

def lambda_handler(event, context):
    S3 = boto3.client("s3")
    get_file_content = event["content"]
    decode_content = base64.b64decode(get_file_content)
    s3_upload =
S3.put_object(Bucket="filesuploding",Key="data.txt",Body=decode_content)

    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('The Object is Uploaded successfully!')
    }
```

[] add permission to lambda role, i.e. s3 full access and API gateway (go to IAM, select role, choose which we create lambda add the permissions)

## **Steps to Create to Rest API in Api gateway:**

- [] Navigate to Api gateway console.
- [] Click on Create new Api.
- [] Choose Api type as Rest API and click on Build.

## REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:

Lambda, HTTP, AWS Services

Import

Build

[] Select as New Api, Enter Api name, Description(optional), Api Endpoint type as “Regional” and Click on Create Api.

### API details

☒ **New API**

Create a new REST API.

☐ **Clone existing API**

Create a copy of an API in this AWS account.

☐ **Import API**

Import an API from an OpenAPI definition.

☐ **Example API**

Learn about API Gateway with an example API.

API name

myRESTAPI |

Description - *optional*

API endpoint type

Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

Regional

Cancel

Create API

[] Once api is created successfully, Click on Create resource.

### Resources

API actions ▼

Deploy API

Create resource

#### Resource details

Update documentation

Enable CORS

Path

/

Resource ID

kyfx9gvsdd

[] Enter Resource name (eg: upload\_data) and Click on create resource.

## Create resource

### Resource details

☒ **Proxy resource** [Info](#)  
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path  
/

▼

Resource name  
upload\_data

☐ **CORS (Cross Origin Resource Sharing)** [Info](#)  
Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel

Create resource

[] Once a resource is created, Click on Create Method.

## Resources

Create resource

/

/upload\_data

**Resource details**

DeleteUpdate documentationEnable CORS

Path  
/upload\_data

Resource ID  
mynhuq

**Methods (0)**

DeleteCreate method

Method type	Integration type	Authorization	API key
No methods			
No methods defined.			

[] Select Method type as Put, Integration Type as “Lambda function”, Lambda Function as function created in Step and Click on Create Method.

### Method details


Method type

PUT

Integration type


☒ **Lambda function**

Integrate your API with a Lambda function.




☐ **HTTP**

Integrate with an existing HTTP endpoint.




☐ **Mock**

Generate a response based on API Gateway mappings and transformations.




☐ **AWS service**

Integrate with an AWS Service.



☐ **VPC link**

Integrate with a resource that isn't accessible over the public internet.




☐ **Lambda proxy integration**


Send the request to your Lambda function as a structured event.

**Lambda function**

Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-west-2

 Choose a Lambda function or enter its ARN

 Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

**Integration timeout** | [Info](#)

By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms

29000

[] After method is created, in Integration Request tab, click on Edit

Create resource

/

/upload\_data

PUT

/upload\_data - PUT - Method execution

Update documentation

Delete

ARN

arn:aws:execute-api:us-west-2:921617140173:0vcus3y887/\*PUT/upload\_data

Resource ID

mynhuq

Client

Method request

Integration request

Method response

Integration response

Lambda integration

Method request

Integration request

Integration response

Method response

Test

Integration request settings

Edit

**[ ]** Select Request body passthrough as “When no template matches the request content-type header”. In Mapping Template, add mapping template, enter Content type as “application/pdf and Template body as below and Click on Save.

Method details

Integration type

☒ Lambda function

Integrate your API with a Lambda function.

☐ HTTP

Integrate with an existing HTTP endpoint.

☐ Mock

Generate a response based on API Gateway mappings and transformations.

☐ AWS service

Integrate with an AWS Service.

☐ VPC link

Integrate with a resource that isn't accessible over the public internet.

☒ Lambda proxy integration

Send the request to your Lambda function as a structured event.

Lambda function

Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-west-2

arn:aws:lambda:us-west-2:921617140173:function:s3fi

Execution role

arn:aws:iam::myAccount:role/myRole

Credential cache

Do not add caller credentials to cache key

Integration timeout

Info

By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms.

29000

Request body passthrough

☐ When there are no templates defined (recommended)

☒ When no template matches the request content-type header

☐ Never

⚠ If you set the request body passthrough to **When no templates matches** the request content-type header, API Gateway will pass all request payloads directly to the endpoint without transformation, and will transform any matches for the incoming content type. To secure your integration, select **When there are no templates defined (recommended)**.

► URL path parameters

► URL query string parameters

► URL request headers parameters

▼ Mapping templates

Content type

application/pdf

Remove

Generate template

Template body

```
1 {  
2   "content": "$input.body"  
3 }
```

```
{  
  "content": "$input.body"  
}
```

In the left panel choose Api Settings. In the Binary Media Types, select manage media type, section choose Add Binary Media Type. Add the string \*/\* and click Save changes.

**API Gateway**

APIs

Custom domain names

VPC links

API: myRESTAPI

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

API keys

Client certificates

Settings

Create resource

/

/upload\_data

PUT

Client

Method response

Integration response

Lambda integration on

Method request

**Integration request**

Integration response

Method response

Test

Integration request settings

Edit

Integration type [info](#)

Lambda

Region

us-west-2

Lambda proxy integration [info](#)

False

Lambda function

s3file

Input passthrough

When no template matches the request content-type header

Timeout

Default (29 seconds)

URL path parameters (0)

< 1 >

Name	Mapped from	Caching
No URL path parameters		

No URL path parameters defined

**Binary media types (0)**

Manage media types

Add or remove media types that contain binary data.

< 1 >

Media type

No media types

You don't have any media types.

Manage media types

## Manage binary media types

**Binary media types**

To configure binary support for your API, enter media types that your API should treat as binary data. API Gateway checks the Content-Type and Accept HTTP headers to determine how to handle the request body.

Binary media type

\*/\*

Remove

Add binary media type

Cancel

Save changes

[] Click on Deploy Api. We see a small pop up



Resources

API actions ▼ **Deploy API**

Create resource

/

/upload\_data  
PUT

**Resource details** Update documentation Enable CORS

Path / Resource ID kyfx9gvsdd

**Methods (0)** Delete Create method

Method type ▲	Integration type ▼	Authorization ▼	API key ▼
No methods			
No methods defined.			

[] If you are deploying Api for the first time, choose stage as *\*New stage\**, Enter Stage name and Click on Deploy.

**Deploy API** ✕

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)

Stage

\*New stage\* ▼

Stage name

test

**i** A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description

Cancel **Deploy**

[] Once its deployed, it redirects to stage page, where we can see invoke url. Copy the Invoke Url.

Stages Stage actions ▼ Create stage

test

Stage details [Info](#)

Stage name

test

Cache cluster [Info](#)

⊖ Inactive

Default method-level caching

⊖ Inactive

Rate [Info](#)

-

Burst [Info](#)

-

Web ACL

-

Client certificate

-

Invoke URL

<https://0vcus3y887.execute-api.us-west-2.amazonaws.com/test>

Edit

## Test the API:

[] We can use Postman to test the connectivity.

1. in new select HTTPS
2. Choose Put
3. paste the URL
4. select Header
5. Content-type
6. application/pdf
7. select body
8. select Binary
9. upload files and Click on Send.

New Import PUT https://b4zaht3rbl.exe... + ... No Environment ▼

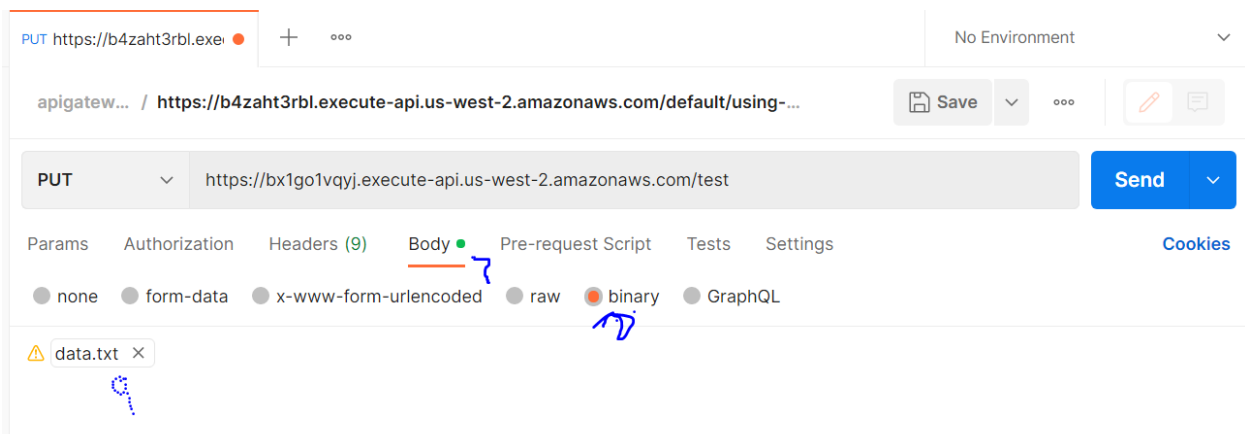
apigatew... / https://b4zaht3rbl.execute-api.us-west-2.amazonaws.com/default/using-...

PUT **2** https://bx1go1vqyj.execute-api.us-west-2.amazonaws.com/test **3** Send ▼

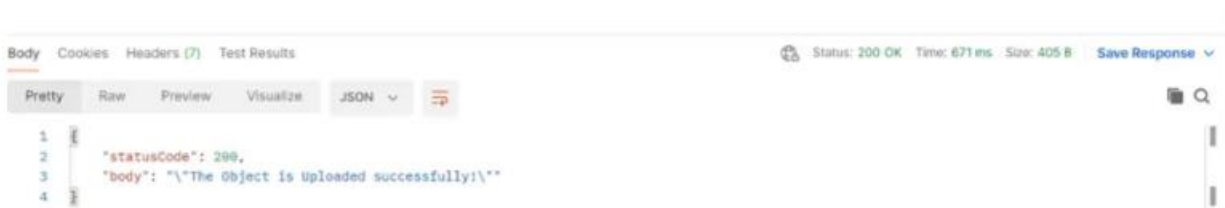
Params Authorization Headers (9) **4** Body ● Pre-request Script Tests Settings Cookies

Headers **5** 8 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/> Content-type <b>6</b>	application/pdf				
Key	Value	Description			



[] we will get output.



[] check the s3 bucket.