

DemoApi

/getPerson

GET

/createPerson

POST



DemoLambda

APIs

Custom domain names

VPC links

Amazon API Gateway

create, maintain, and secure APIs at any scale

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.

Choose an API type

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

[Import](#)[Build](#)

WebSocket API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

AWS Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

awssimplified ▾ N. Virginia ▾ Support

AWS Lambda X

Lambda > Functions

Functions (0)

Last fetched in 0 seconds ↻ Actions ▾ Create function

< 1 > ⚙️

Function name	Description	Package type	Runtime	Code size	Last modified
There is no data to display.					

Filter by tags and attributes or search by keyword

Dashboard

Applications

Functions

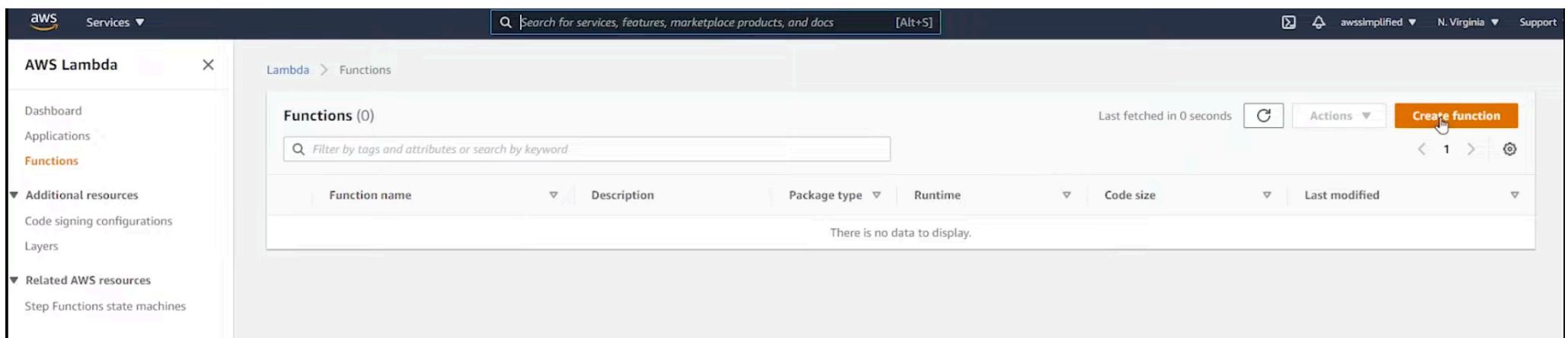
Additional resources

Code signing configurations

Layers

Related AWS resources

Step Functions state machines



AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] awssimplified N. Virginia Support

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- Author from scratch** Start with a simple Hello World example.
- Use a blueprint** Build a Lambda application from sample code and configuration presets for common use cases.
- Container image** Select a container image to deploy for your function.
- Browse serverless app repository** Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

▶ Advanced settings

Cancel **Create function**

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] awssimplified ▾ N. Virginia ▾ Support

arn:aws:lambda:us-east-1:755314965794:function:DemoAppFunction

Code Test Monitor Configuration Aliases Versions

Code source Info Upload from ▾

File Edit Find View Go Tools Window Test Deploy Changes deployed

Go to Anything (Ctrl-P) lambda_function +

DemoAppFunction λ lambda_function.py

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S]

API Gateway X

APIs Custom domain names VPC links

Step 1 Create an API

Step 2 Configure routes

Step 3 Define stages

Step 4 Review and create

Create an API

Create and configure integrations

Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.

Integrations Info

Lambda

AWS Region Lambda function Version Learn more.

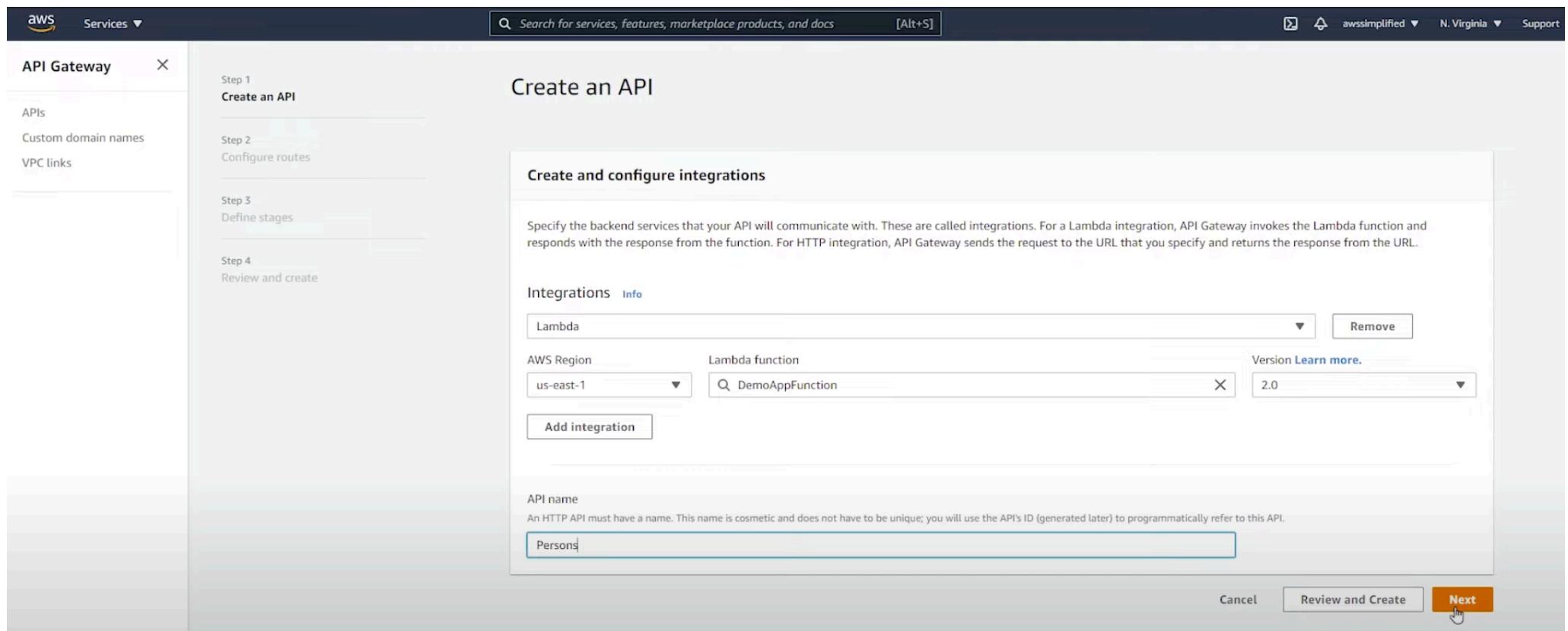
us-east-1 Q DemoAppFunction 2.0

Add integration

API name
An HTTP API must have a name. This name is cosmetic and does not have to be unique; you will use the API's ID (generated later) to programmatically refer to this API.

Persons

Cancel Review and Create Next



AWS Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

awssimplified ▾ N. Virginia ▾ Support

API Gateway X

Step 1 Create an API

Step 2 Configure routes

Step 3 Define stages

Step 4 Review and create

Configure routes

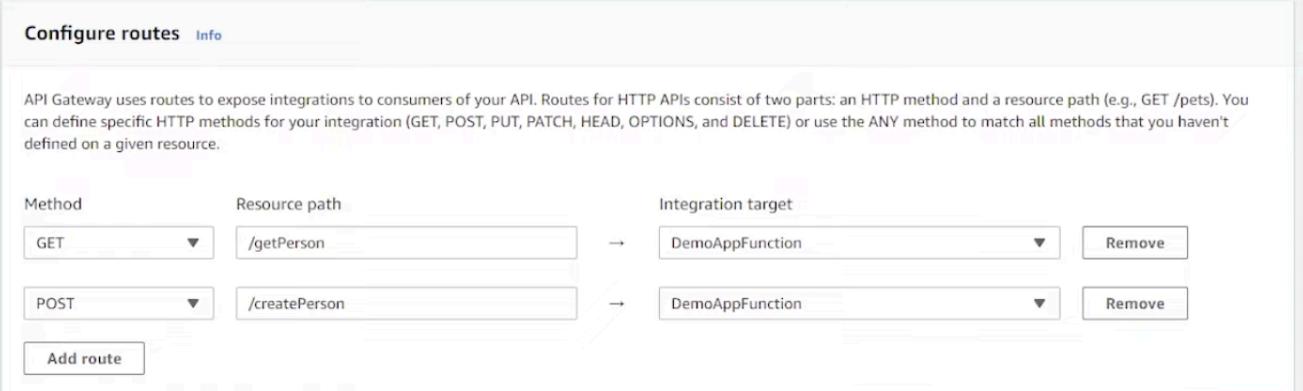
Configure routes Info

API Gateway uses routes to expose integrations to consumers of your API. Routes for HTTP APIs consist of two parts: an HTTP method and a resource path (e.g., GET /pets). You can define specific HTTP methods for your integration (GET, POST, PUT, PATCH, HEAD, OPTIONS, and DELETE) or use the ANY method to match all methods that you haven't defined on a given resource.

Method	Resource path	Integration target
GET	/getPerson	DemoAppFunction
POST	/createPerson	DemoAppFunction

Add route

Cancel Previous Next



AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S]

API Gateway X

Step 1 Create an API

Step 2 Configure routes

Step 3 Define stages

Step 4 Review and create

Define stages

Configure stages Info

Stages are independently configurable environments that your API can be deployed to. You must deploy to a stage for API configuration changes to take effect, unless that stage is configured to autodeploy. By default, all HTTP APIs created through the console have a default stage named \$default. All changes that you make to your API are autodeployed to that stage. You can add stages that represent environments such as development or production.

Stage name Auto-deploy Remove

Add stage

Cancel Previous Next

The screenshot shows the 'Define stages' step of creating an API in the AWS API Gateway console. On the left, a sidebar lists 'APIs', 'Custom domain names', and 'VPC links'. The main area has a header 'Step 3 Define stages'. Below it, a section titled 'Configure stages' contains a description of stages as independent environments. A table row shows a stage named '\$default' with an 'Auto-deploy' checkbox checked and a 'Remove' button. A 'Add stage' button is also present. At the bottom, there are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted by a mouse cursor.

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S]

awssimplified ▾ N. Virginia ▾ Support

API Gateway X

APIs
Custom domain names
VPC links

Step 1 Create an API

Step 2 Configure routes

Step 3 Define stages

Step 4 Review and create

Review and create

API name and integrations

API name Persons

Integrations DemoAppFunction (Lambda)

Routes

Routes

GET /getPerson → DemoAppFunction (Lambda)
POST /createPerson → DemoAppFunction (Lambda)

Stages

Stages \$default (Auto-deploy: enabled)

Cancel Previous Create

Searched for services, features, marketplace products, and docs [Alt+S]

Successfully created API Persons (prg8ewzsr4)

API Gateway Details Stage: - Deploy

APIs Custom domain names VPC links

API: Persons (prg8ewzsr4)

Develop Routes Authorization Integrations CORS Reimport Export

Deploy Stages

Protect Throttling

Monitor Metrics Logging

Persons

API details

API ID	prg8ewzsr4	Protocol	HTTP	Created	2021-05-29
Description	No Description	Default endpoint	Enabled		

Stages for Persons

Stage name	Invoke URL	Attached deployment	Auto deploy	Last updated
\$default	https://prg8ewzsr4.execute-api.us-east-1.amazonaws.com	hi7m87	enabled	2021-05-29

Tags (0)

Key	Value
No Tags	



The screenshot shows the AWS Lambda console interface. At the top, there's a search bar with the placeholder "Search for services, features, marketplace products, and docs" and a keyboard shortcut "[Alt+S]". To the right of the search bar are icons for notifications, user account, and support. Below the header, the URL "arn:aws:lambda:us-east-1:755314965794:function:DemoAppFunction" is displayed.

The main area is divided into tabs: "Code" (which is selected), "Test", "Monitor", "Configuration", "Aliases", and "Versions".

Under the "Code" tab, there's a "Code source" section with a "Info" link and an "Upload from" button. A toolbar above the code editor includes "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test" (with a dropdown arrow), "Deploy", and a green "Changes deployed" button.

The code editor displays the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

The screenshot shows the AWS Lambda code editor interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, search bar ('Search for services, features, marketplace products, and docs'), and keyboard shortcut '[Alt+S]'. To the right are icons for 'awsimplified' (a simplified UI), 'N. Virginia' (region), and 'Support'.

The main area has tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected. Below it, there's a toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (dropdown), 'Deploy' (button), and a status message 'Changes deployed'.

The central workspace displays a file tree under 'lambda_function':

- lambda_function (selected)
- DemoAppFunction (folder)
 - lambda_function.py

The code editor shows the contents of `lambda_function.py`:

```
1 import json
2
3 def lambda_handler(event, context):
4     print(event)
5 
```

A sidebar on the left is titled 'Environment' and contains a small icon of a hand pointing at a screen.

CloudWatch X

CloudWatch > Log groups > /aws/lambda/DemoAppFunction > 2021/05/29/[\$LATEST]4b3655591bf3492b9ab34828b88ca04e

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns [? Filter patterns](#)

View as text [Copy](#) Actions ▾ Create Metric Filter

Filter events Clear 1m 30m 1h 12h Custom [Copy](#)

▶	Timestamp	Message
		No older events at this moment. Retry
▶	2021-05-29T18:22:10.098-04:00	START RequestId: afc2a2fa-df9d-4d90-93c8-83f66821f98a Version: \$LATEST
▼	2021-05-29T18:22:10.100-04:00	{"version": "2.0", "routeKey": "GET /getPerson", "rawPath": "/getPerson", "rawQueryString": "personId=personId123", "headers": {"accept": "*/*", "accept-encoding": "gzip, deflate, br", "x-amzn-requestid": "afc2a2fa-df9d-4d90-93c8-83f66821f98a", "x-amzn-trace-id": "Root=1-60b2be91-47a4bf7549c607d510a9c469", "x-forwarded-for": "24.150.131.11", "x-forwarded-port": "443", "x-forwarded-proto": "https", "queryStringParameters": {"personId": "personId123"}, "requestContext": {"accountId": "755314965794", "apiId": "prgBewzs4", "domainName": "prgBewzs4.execute-api.us-east-1.amazonaws.com", "domainPrefix": "prgBewzs4", "http": {"method": "GET", "path": "/getPerson", "protocol": "HTTP/1.1", "sourceIp": "24.150.131.11", "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36"}, "requestId": "AHK2zj-OoAMEPig", "routeKey": "GET /getPerson", "stage": "\$default", "time": "29/May/2021:22:22:09 +0000", "timeEpoch": 1622326929781}, "isBase64Encoded": false}
▶	2021-05-29T18:22:10.102-04:00	END RequestId: afc2a2fa-df9d-4d90-93c8-83f66821f98a
▶	2021-05-29T18:22:10.102-04:00	REPORT RequestId: afc2a2fa-df9d-4d90-93c8-83f66821f98a Duration: 1.75 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 50 MB Init Duration: 162.26 ms
▶	2021-05-29T18:22:20.063-04:00	START RequestId: c79a86bb-77db-4703-a8e7-de1be89ecd78 Version: \$LATEST
▶	2021-05-29T18:22:20.066-04:00	{"version": "2.0", "routeKey": "GET /getPerson", "rawPath": "/getPerson", "rawQueryString": "personId=personId123", "headers": {"accept": "*/*", "accept-encoding": "gzip, deflate, br", "x-amzn-requestid": "c79a86bb-77db-4703-a8e7-de1be89ecd78", "x-amzn-trace-id": "Root=1-60b2be91-47a4bf7549c607d510a9c469", "x-forwarded-for": "24.150.131.11", "x-forwarded-port": "443", "x-forwarded-proto": "https", "queryStringParameters": {"personId": "personId123"}, "requestContext": {"accountId": "755314965794", "apiId": "prgBewzs4", "domainName": "prgBewzs4.execute-api.us-east-1.amazonaws.com", "domainPrefix": "prgBewzs4", "http": {"method": "GET", "path": "/getPerson", "protocol": "HTTP/1.1", "sourceIp": "24.150.131.11", "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36"}, "requestId": "AHK2zj-OoAMEPig", "routeKey": "GET /getPerson", "stage": "\$default", "time": "29/May/2021:22:22:09 +0000", "timeEpoch": 1622326929781}, "isBase64Encoded": false}
▶	2021-05-29T18:22:20.067-04:00	END RequestId: c79a86bb-77db-4703-a8e7-de1be89ecd78
▶	2021-05-29T18:22:20.067-04:00	REPORT RequestId: c79a86bb-77db-4703-a8e7-de1be89ecd78 Duration: 1.31 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 51 MB
▶	2021-05-29T18:22:20.659-04:00	START RequestId: 7e289025-62b2-4f83-9fba-2e213d7d548f Version: \$LATEST
▶	2021-05-29T18:22:20.664-04:00	{"version": "2.0", "routeKey": "GET /getPerson", "rawPath": "/getPerson", "rawQueryString": "personId=personId123", "headers": {"accept": "*/*", "accept-encoding": "gzip, deflate, br", "x-amzn-requestid": "7e289025-62b2-4f83-9fba-2e213d7d548f", "x-amzn-trace-id": "Root=1-60b2be91-47a4bf7549c607d510a9c469", "x-forwarded-for": "24.150.131.11", "x-forwarded-port": "443", "x-forwarded-proto": "https", "queryStringParameters": {"personId": "personId123"}, "requestContext": {"accountId": "755314965794", "apiId": "prgBewzs4", "domainName": "prgBewzs4.execute-api.us-east-1.amazonaws.com", "domainPrefix": "prgBewzs4", "http": {"method": "GET", "path": "/getPerson", "protocol": "HTTP/1.1", "sourceIp": "24.150.131.11", "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36"}, "requestId": "AHK2zj-OoAMEPig", "routeKey": "GET /getPerson", "stage": "\$default", "time": "29/May/2021:22:22:09 +0000", "timeEpoch": 1622326929781}, "isBase64Encoded": false}
▶	2021-05-29T18:22:20.665-04:00	END RequestId: 7e289025-62b2-4f83-9fba-2e213d7d548f
▶	2021-05-29T18:22:20.665-04:00	REPORT RequestId: 7e289025-62b2-4f83-9fba-2e213d7d548f Duration: 1.36 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 51 MB
		No newer events at this moment. Auto retry paused. Resume

API Gateway | DemoAppFunction - Lambda | CloudWatch Management Console | json curious concept - Bing | JSON Formatter & Validator

https://jsonformatter.curiousconcept.com/#

JSON FORMATTER & VALIDATOR

About Learn Bookmarklet FAQ Changelog Contact [Twitter](#)

 chrome
Switch to Google Chrome,
the secure way to browse.
[Yes, get Chrome now](#)

#1 May 29th 2021, 6:25:02 pm

VALID (RFC 8259)

Fixer/Validator Output

- Info:** Replaced incorrect quotes.
- Info:** Escaped unescaped character.
- Info:** Converted false to lowercase.

Formatted JSON Data

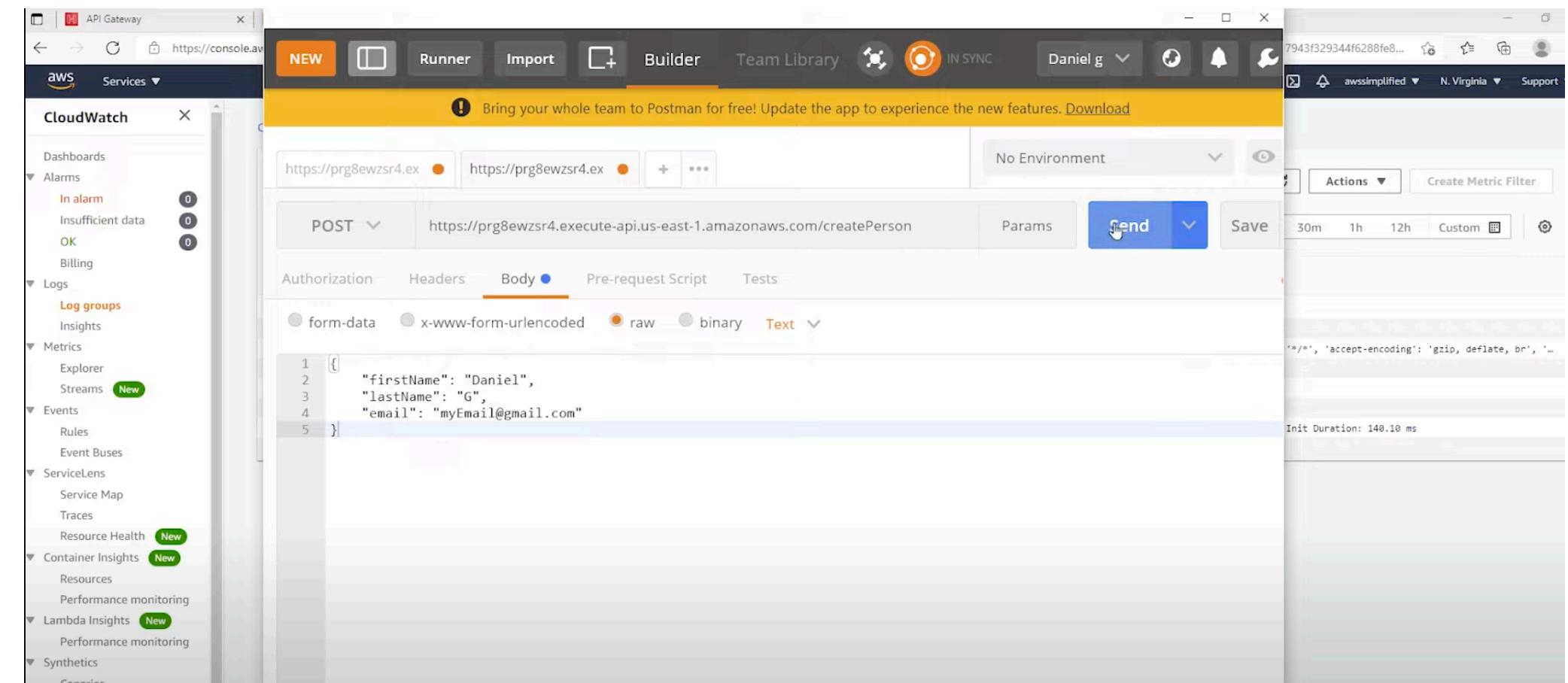
```
{ "version": "2.0", "routeKey": "GET /getPerson", "rawPath": "/getPerson", "rawQueryString": "personId=personId123", "headers": { "accept": "*/*", "accept-encoding": "gzip, deflate, br", "accept-language": "en-US,en;q=0.9", "cache-control": "no-cache", "content-length": "0", "host": "prg8ewzsr4.execute-api.us-east-1.amazonaws.com", "postman-token": "6259ec56-bcc7-2628-07b8-1e96c5d7a05b", "sec-ch-ua": "\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"90\", \"Google Chrome\";v=\"", "sec-ch-ua-mobile": "0" }}
```

```
{ □
  "version": "2.0",
  "routeKey": "GET /getPerson",
  "rawPath": "/getPerson",
  "rawQueryString": "personId=personId123",
  "headers": { □
    "accept": "*/*",
    "accept-encoding": "gzip, deflate, br",
    "accept-language": "en-US,en;q=0.9",
    "cache-control": "no-cache",
    "content-length": "0",
    "host": "prg8ewzsr4.execute-api.us-east-1.amazonaws.com",
    "postman-token": "6259ec56-bcc7-2628-07b8-1e96c5d7a05b",
    "sec-ch-ua": "\\" Not A;Brand\\";v=\\"99\\", \\"Chromium\\";v=\\"90\\", \\"Google Chrome\\";v=\\"90\\",
    "sec-ch-ua-mobile": "?0",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "none",
    "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36",
    "x-amzn-trace-id": "Root=1-60b2be91-47a48f7549c607d510a9c469",
    "x-forwarded-for": "24.150.131.11",
    "x-forwarded-port": "443",
    "x-forwarded-proto": "https"
  },
  "queryStringParameters": { □
    "personId": "personId123"
  },
  "requestContext": { □
    "accountId": "755314965794",
    "apiId": "prg8ewzsr4"
  }
}
```

The screenshot shows the AWS Lambda console interface. At the top, there are several tabs: API Gateway, DemoAppFunction - Lambda, CloudWatch Management Console, json curious concept - Bing, and JSON Formatter & Validator. Below the tabs, the AWS Services navigation bar includes links for Services, Search for services, features, marketplace products, and docs, and account information like awssimplified, N. Virginia, and Support.

The main content area is titled "Code source" and "Info". It features a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message "Changes not deployed". On the left, there's a sidebar for "Environment" with a dropdown menu showing "DemoAppFunction" and "lambda_function.py". The main code editor window displays the following Python code:

```
1 import json
2 import uuid
3
4 GET_RAW_PATH = "/getPerson"
5 CREATE_RAW_PATH = "/createPerson"
6
7 def lambda_handler(event, context):
8     print(event)
9     if event['rawPath'] == GET_RAW_PATH:
10         #GetPerson Path - call database
11         print("Start Request for GetPerson")
12         personId = event['queryStringParameters']['personId']
13         print("Received request with personId= " + personId)
14         return { "firstName": "Daniel", "lastName": "G", "email": "myEmail@gmail.com"}
15
16     elif event['rawPath'] == CREATE_RAW_PATH:
17         #CreatePerson Path - write to database
18         print("Start Request for CreatePerson")
19
```



```
{
  "version": "2.0",
  "routeKey": "POST /createPerson",
  "rawPath": "/createPerson",
  "rawQueryString": "",
  "headers": {
    "accept": "*/*",
    "accept-encoding": "gzip, deflate, br",
    "accept-language": "en-US,en;q=0.9",
    "cache-control": "no-cache",
    "content-length": "75",
    "content-type": "text/plain; charset=UTF-8",
    "host": "prg8ewzsr4.execute-api.us-east-1.amazonaws.com",
    "origin": "chrome-extension://fhbjgbiflinjbdggehcdcbnccccdomop",
    "postman-token": "3b679593-baa3-ba12-2b63-0cd77ea69903",
    "sec-ch-ua": "\\" Not A;Brand\\";v=\"99\", \\"Chromium\\";v=\"90\", \\"Google Chrome\\";v=\"90\"",
    "sec-ch-ua-mobile": "?0",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "none",
    "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36",
    "x-amzn-trace-id": "Root=1-60b2c11b-4868a72e5d63adfc09a8c658",
    "x-forwarded-for": "24.150.131.11",
    "x-forwarded-port": "443",
    "x-forwarded-proto": "https"
  },
  "requestContext": {
    "accountId": "755314965794",
    "apiId": "prg8ewzsr4",
    "domainName": "prg8ewzsr4.execute-api.us-east-1.amazonaws.com"
  }
}
```

```
"origin":"chrome-extension://tmbjgb1t1jnjbddgencaacbncaadomop",
"postman-token":"3b679593-baa3-ba12-2b63-0cd77ea69903",
"sec-ch-ua":"\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"90\", \"Google Chrome\";v=\"90\"",
"sec-ch-ua-mobile":"?0",
"sec-fetch-dest":"empty",
"sec-fetch-mode":"cors",
"sec-fetch-site":"none",
"user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36",
"x-amzn-trace-id":"Root=1-60b2c11b-4868a72e5d63adfc09a8c658",
"x-forwarded-for":"24.150.131.11",
"x-forwarded-port":"443",
"x-forwarded-proto":"https"
},
"requestContext":{█
  "accountId":"755314965794",
  "apiId":"prg8ewzsr4",
  "domainName":"prg8ewzsr4.execute-api.us-east-1.amazonaws.com",
  "domainPrefix":"prg8ewzsr4",
  "http":{█
    "method":"POST",
    "path":"/createPerson",
    "protocol":"HTTP/1.1",
    "sourceIp":"24.150.131.11",
    "userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36"
  },
  "requestId":"AHMcZh2_oAMEP3w=",
  "routeKey":"POST /createPerson",
  "stage":"$default",
  "time":"29/May/2021:22:32:59 +0000",
  "timeEpoch":1622327579979
},
"body":█
  "\n\t\"firstName\": \"Daniel\", \n\t\"lastName\": \"G\", \n\t\"email\": \"myEmail@gmail.com\"\n}",
"isBase64Encoded":false
```

arn:aws:lambda:us-east-1:755314965794:function:DemoAppFunction

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from ▾

File Edit Find View Go Tools Window

Test ▾

Deploy

Changes deployed

Your changes have been deployed



Go to Anything (Ctrl-P)

lambda_function x

Environment
DemoAppFunction
lambda_function.py

```
6
7 def lambda_handler(event, context):
8     print(event)
9     if event['rawPath'] == GET_RAW_PATH:
10         #GetPerson Path - call database
11         print("Start Request for GetPerson")
12         personId = event['queryStringParameters']['personId']
13         print("Received request with personId= " + personId)
14         return { "firstName": "Daniel", "lastName": "G", "email": "myEmail@gmail.com"}
15
16     elif event['rawPath'] == CREATE_RAW_PATH:
17         #CreatePerson Path - write to database
18         print("Start Request for CreatePerson")
19         decodedEvent = json.loads(event['body'])
20         firstName = decodedEvent['firstName']
21         print("Received request with firstName=" + firstName)
22         #call database
23         return { "personId": str(uuid.uuid1())}
24
25
26
```

API Gateway

NEW Runner Import Builder Team Library IN SYNC Daniel g No Environment

Bring your whole team to Postman for free! Update the app to experience the new features. [Download](#)

https://prg8ewzsr4.ex https://prg8ewzsr4.ex + ... No Environment

Code source Info

File Edit Find View

Go to Anything (Ctrl-P)

DemoAppFunction lambda_function.py

Environment

POST https://prg8ewzsr4.execute-api.us-east-1.amazonaws.com/createPerson Params Send Save

Upload from

```
1 {  
2   "firstName": "Daniel",  
3   "lastName": "G",  
4   "email": "myEmail@gmail.com"  
5 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 94ms

Pretty Raw Preview JSON

```
1 {  
2   "personId": "90d233dd-c0ce-11eb-a3a6-7fae390ef802"  
3 }
```

Code properties

This screenshot shows a Postman test environment for an AWS Lambda function named 'DemoAppFunction'. The 'Code source' tab is selected, displaying the Lambda function's code in Python:

```
1 {  
2   "firstName": "Daniel",  
3   "lastName": "G",  
4   "email": "myEmail@gmail.com"  
5 }
```

The 'Body' tab shows the JSON request body:

```
1 {  
2   "personId": "90d233dd-c0ce-11eb-a3a6-7fae390ef802"  
3 }
```

The 'Test Results' tab shows a successful response with status 200 OK and time 94ms.

CloudWatch	
Dashboards	
Alarms	
In alarm	0
Insufficient data	0
OK	0
Billing	
Logs	
Log groups	
Insights	
Metrics	
Explorer	
Streams	New
Events	
Rules	
Event Buses	
ServiceLens	
Service Map	
Traces	
Resource Health	New
Container Insights	New
Resources	
Performance monitoring	
Lambda Insights	New
Performance monitoring	
Synthetics	
Canaries	
Contributor Insights	
Settings	
Favorites	
+ Add a dashboard	
2021-05-29T18:38:19.038-04:00	
▶ 2021-05-29T18:38:19.042-04:00	
START RequestId: 6b239fb0-e740-46c6-933c-eac57b21790b Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:19.042-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:19.042-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:19.042-04:00	
END RequestId: 6b239fb0-e740-46c6-933c-eac57b21790b	
▶ 2021-05-29T18:38:19.042-04:00	
REPORT RequestId: 6b239fb0-e740-46c6-933c-eac57b21790b Duration: 1.40 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
▶ 2021-05-29T18:38:19.791-04:00	
START RequestId: e7c30971-4252-4ab1-becd-de9d075ca20f Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:19.796-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:19.796-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:19.797-04:00	
END RequestId: e7c30971-4252-4ab1-becd-de9d075ca20f	
▶ 2021-05-29T18:38:19.797-04:00	
REPORT RequestId: e7c30971-4252-4ab1-becd-de9d075ca20f Duration: 1.40 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
▶ 2021-05-29T18:38:20.529-04:00	
START RequestId: 5723560c-f3f5-4c6b-846d-15eea234f735 Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:20.535-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:20.535-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:20.536-04:00	
END RequestId: 5723560c-f3f5-4c6b-846d-15eea234f735	
▶ 2021-05-29T18:38:20.536-04:00	
REPORT RequestId: 5723560c-f3f5-4c6b-846d-15eea234f735 Duration: 1.53 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
▶ 2021-05-29T18:38:21.457-04:00	
START RequestId: 90484900-f72f-48d8-bc4e-edeb5b70761f8 Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:21.459-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:21.459-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:21.459-04:00	
END RequestId: 90484905-f72f-48d8-bc4e-edeb5b70761f8	
▶ 2021-05-29T18:38:21.460-04:00	
REPORT RequestId: 90484905-f72f-48d8-bc4e-edeb5b70761f8 Duration: 1.44 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
▶ 2021-05-29T18:38:22.509-04:00	
START RequestId: 8931fbfd-427e-4a18-ab62-a7cd0d8b8df0 Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:22.513-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:22.513-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:22.513-04:00	
END RequestId: 8931fbfd-427e-4a18-ab62-a7cd0d8b8df0	
▶ 2021-05-29T18:38:22.513-04:00	
REPORT RequestId: 8931fbfd-427e-4a18-ab62-a7cd0d8b8df0 Duration: 1.36 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
▶ 2021-05-29T18:38:23.300-04:00	
START RequestId: 110f319c-96e6-40d4-916f-8a8981332a8e Version: \$LATEST	
{'version': '2.0', 'routeKey': 'POST /createPerson', 'rawPath': '/createPerson', 'rawQueryString': '', 'headers': {'accept': '*/*', 'accept-encoding': 'gzip, deflate, br', 'accept-lan...	
▶ 2021-05-29T18:38:23.304-04:00	
Start Request for CreatePerson	
▶ 2021-05-29T18:38:23.304-04:00	
Received request with firstName=Daniel	
▶ 2021-05-29T18:38:23.305-04:00	
END RequestId: 110f319c-96e6-40d4-916f-8a8981332a8e	
▶ 2021-05-29T18:38:23.305-04:00	
REPORT RequestId: 110f319c-96e6-40d4-916f-8a8981332a8e Duration: 1.23 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 54 MB	
No newer events at this moment. Auto retry paused. Resume	