

Cloud and Serverless Computing

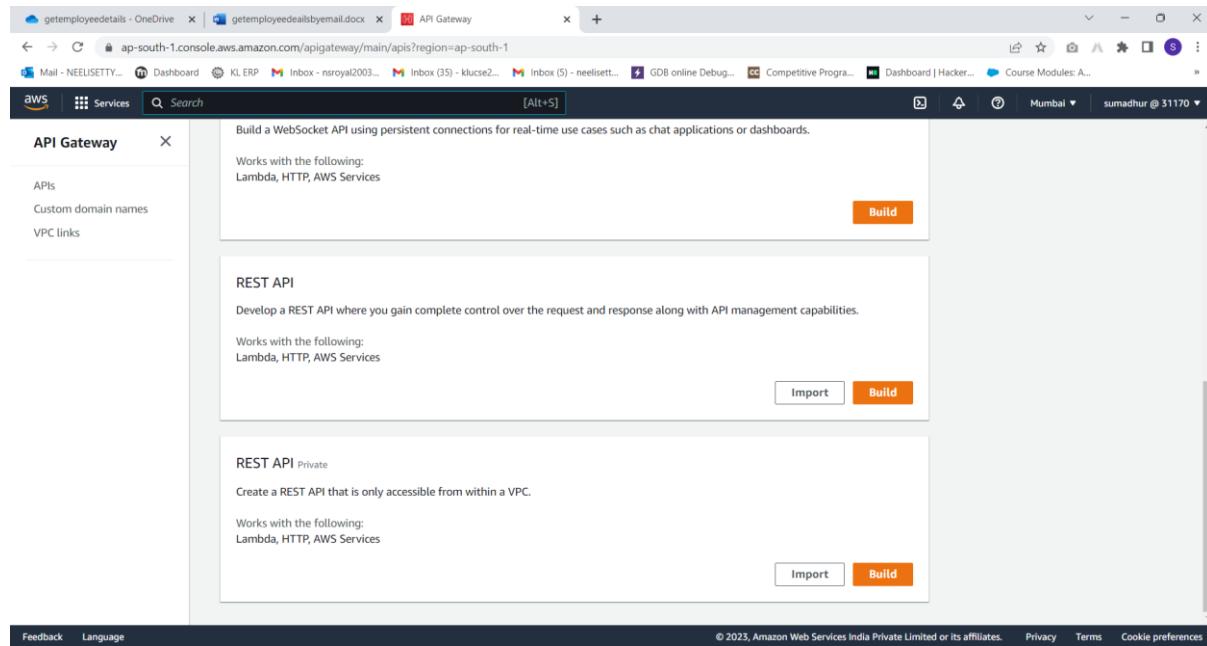
Develop Serverless WebApplication on AWS using GET-METHOD
getstudents by id (BUS)

Name: G.Tarun Datta

Kusumanth Reddy

ID: 2000030284

2000030539



Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

REST WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

New API Import from Swagger or Open API 3 Example API

Settings

Choose a friendly name and description for your API.

API name*

Description

Endpoint Type

* Required Create API

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

New Child Resource

Use this page to create a new child resource for your resource.

Configure as proxy resource

Resource Name*

Resource Path*

You can add path parameters using brackets. For example, the resource path `{username}` represents a path parameter called 'username'. Configuring `/proxy{}` as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to `/foo`. To handle requests to `/`, add a new ANY method on the `/` resource.

Enable API Gateway CORS

* Required Cancel Create Resource

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS API Gateway 'Create API' interface. On the left, a sidebar lists various API components: APIs, Custom Domain Names, VPC Links, API: GetEmployeeDet..., Resources, Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, Dashboard, Settings, and Usage Plans. The 'Resources' section is currently selected. In the main pane, the path is shown as: APIs > GetEmployeeDetailsByEmail (oxa45258ge) > Resources > /getcustomerdetailsbyemail (y41kui) > GET. A sub-menu titled 'Actions' is open, showing options like 'Edit', 'Test', 'Mock', 'AWS Lambda', 'HTTP', 'VPC Link', and 'Delete'. The 'Integration type' dropdown is set to 'Mock'. A 'Save' button is visible at the bottom right. The status bar at the bottom indicates: © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences.

This screenshot shows the 'Method Execution' interface for the 'getcustomerdetailsbyemail' resource. The left sidebar is identical to the previous screenshot. The main pane displays the execution flow: 'Client' (TEST button) → 'Method Request' (Auth: NONE, ARN: arn:aws:execute-api:ap-south-1:609971362146:oxa45258ge/*:GET/ge) → 'Integration Request' (Type: MOCK) → 'Mock Endpoint'. Below this, the 'Method Response' (HTTP Status: 200, Models: application/json => Empty) is shown on the left, and the 'Integration Response' (HTTP status pattern: [200-299], Output passthrough: No) is shown on the right. Arrows indicate the flow direction between each stage. The status bar at the bottom indicates: © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences.

The screenshot shows the AWS API Gateway interface. On the left, the navigation pane is visible with options like APIs, Resources, Stages, Authorizers, etc. The main area is titled 'Method Execution' for the endpoint '/getcustomerdetailsbyemail - GET'. It displays a table for mapping HTTP status codes to method response statuses. A row for status code 200 is selected, with 'HTTP status regex' set to 'default' and 'Content handling' set to 'Passthrough'. Below this, sections for 'Header Mappings' and 'Mapping Templates' are shown. Under 'Mapping Templates', a Content-Type entry is set to 'application/json'. A mapping template is defined with the following JSON:

```
1 {
2   "FirstName": "sumadur",
3   "LastName": "royal",
4   "Email": "neelisettysumadurroyal@gmail.com"
5 }
```

The screenshot shows the 'Deploy API' dialog box overlaid on the API configuration page. The dialog box has fields for 'Deployment stage' (set to 'New Stage'), 'Stage name*' (set to 'Dev'), and 'Stage description'. The 'Deploy' button is highlighted. In the background, the API configuration page for the '/getcustomerdetailsbyemail' endpoint is visible, showing the same settings as the previous screenshot.

The screenshot shows the AWS API Gateway Dev Stage Editor. On the left, a sidebar lists various API resources like Resources, Stages, Authorizers, and Models. The 'Stages' section is selected, showing 'Dev' as the active stage. The main area is titled 'Dev Stage Editor' and contains tabs for Settings, Logs/Tracing, Stage Variables, SDK Generation, Export, Deployment History, Documentation History, and Canary. Under the 'Settings' tab, there are sections for Cache Settings (with 'Enable API cache' checked), Default Method Throttling (with 'Rate' set to 10000 requests per second and 'Burst' set to 5000 requests), and Web Application Firewall (WAF) (with 'Web ACL' set to 'None'). A note at the bottom says 'Select the Web ACL to be applied to this stage.' At the bottom right of the editor, there are links for 'Feedback', 'Language', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences'.

The screenshot shows the Postman application interface. The left sidebar has sections for Home, Workspaces, API Network, Reports, and Explore. The 'APIs' section is selected, showing 'No APIs yet'. The main area shows an 'Overview' tab for the URL 'https://oxa45258ge.execute-api.ap-south-1.amazonaws.com/Dev'. A 'GET' request is selected with the URL 'https://oxa45258ge.execute-api.ap-south-1.amazonaws.com/Dev'. Below it, there are tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. A 'Query Params' table is shown with one row: KEY 'Key' and VALUE 'Value'. The 'Send' button is visible. Below the request details, there is a 'Response' section with a placeholder image of a rocket launching and the text 'Click Send to get a response'. At the bottom, there are links for 'Find and Replace', 'Console', 'Cookies', 'Capture requests', 'Bootcamp', 'Runner', 'Trash', and a progress bar indicating '67%'. A 'Start working with APIs' button is also present.

Screenshot of the AWS IAM Management Console showing the 'Create role' wizard - Step 1: Select trusted entity.

The page title is 'Select trusted entity'. It shows the 'Trusted entity type' section with the 'AWS service' option selected. Other options include 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is the 'Use case' section, which lists 'Common use cases' (EC2 and Lambda) and 'Use cases for other AWS services' (choose a service to view use case). The Lambda option is selected under common use cases.

Screenshot of the AWS IAM Management Console showing the 'Create role' wizard - Step 3: Name, review, and create.

The page title is 'Name, review, and create'. It shows the 'Role details' section where the 'Role name' is set to 'getEmployeeDetails'. The 'Description' field contains the text 'Allows Lambda functions to call AWS services on your behalf.' A preview of the JSON policy document is shown at the bottom:

```
1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Effect": "Allow",
5         "Action": [
```

Screenshot of the AWS IAM Management Console showing the "Step 2: Add permissions" step of creating a new role.

The page title is "Create function - Lambda".

Permissions policy summary:

Policy name	Type	Attached as
AmazonDynamoDBReadOnlyAccess	AWS managed	Permissions policy
AWSLambda_FullAccess	AWS managed	Permissions policy

Tags:

Add tags - optional / Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Buttons at the bottom:

- Cancel
- Previous
- Create role

Page footer:

- Feedback
- Language
- © 2023, Amazon Web Services India Private Limited or its affiliates.
- Privacy
- Terms
- Cookie preferences

Screenshot of the AWS Lambda 'Create function' wizard.

Author from scratch (selected): 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.

Basic information

Function name: getEmployeeDetailsByEmail

Runtime: Node.js 12.x

Architecture: x86_64

Permissions: 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

Execution role: Create a new role with basic Lambda permissions

Existing role: getEmployeeDetails

Advanced settings

Create function button

Screenshot of the AWS Lambda function configuration page.

The test event myEvent was successfully saved.

Code source

index.js

```
1 exports.handler = async (event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify(`{"Firstname": "sumadhur", "Lastname": "royal", "Email": "neelisettysumadhurroyal@gmail.com"}`)
6     };
7     return response;
8 }
9
```

Code, **Test**, **Monitor**, **Configuration**, **Aliases**, **Versions**

Upload from

Execution results

Feedback, **Language**

The test event myEvent was successfully saved.

Code | Test | Monitor | Configuration | Aliases | Versions

Code source info

File Edit Find View Go Tools Window Test Deploy

Execution result

Status: Succeeded Max memory used: 56 MB Time: 3.52 ms

Test Event Name myEvent

Response

```
"statusCode": 200,
"body": "{\"FirstName\":\"sumadhur\",\"LastName\":\"royal\",\"Email\":\"neelisettysumadhurroyal@gmail.com\"}"
```

Function Logs

```
START RequestId: d4f6dd6-2914-47c2-9dbf-f8ff9b539f1f Version: $LATEST
END RequestId: d4f6dd6-2914-47c2-9dbf-f8ff9b539f1f
REPORT RequestId: d4f6dd6-2914-47c2-9dbf-f8ff9b539f1f Duration: 3.52 ms Billed Duration: 4 ms Memory Size: 128 MB Max Memory Used: 56 MB Init Duration: 156.13 ms
```

Request ID

```
d4f6dd6-2914-47c2-9dbf-f8ff9b539f1f
```

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

APIs / Resources Actions Method Execution /getcustomerdetailsbyemail - GET - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type Lambda Function HTTP Mock AWS Service VPC Link

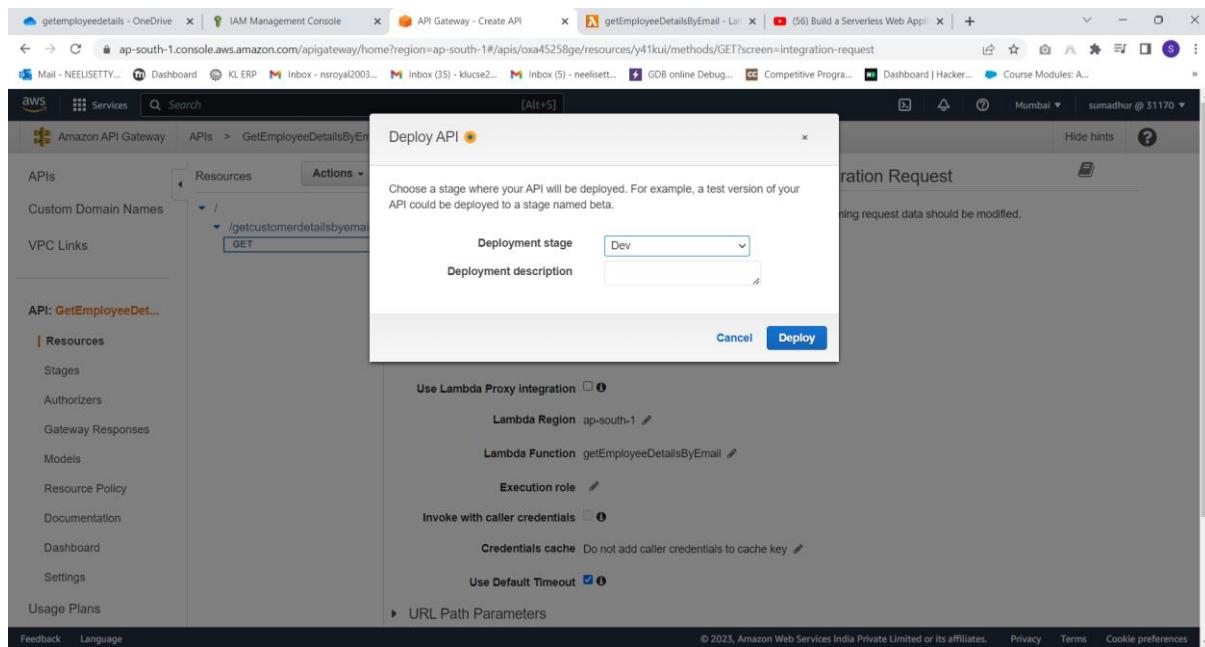
Use Lambda Proxy Integration

Lambda Region ap-south-1 Lambda Function getEmployeeDetailsByEmail Use Default Timeout

Save

Mapping Templates

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences



The screenshot shows the Postman application interface. The left sidebar shows 'My Workspace' with sections for 'Collections', 'Environments', 'Mock Servers', 'Flows', and 'History'. A central panel displays an API endpoint: `https://oxa45258ge.execute-api.ap-south-1.amazonaws.com/Dev/getcustomerdetailsbyemail`. The request method is 'GET'. The 'Params' tab shows a single parameter 'Key' with value 'Value'. The 'Body' tab shows a JSON response:

```
1:   "statusCode": 200,
2:   "body": "{\"FirstName\":\"sumadhur\", \"LastName\":\"royal\", \"Email\":\"neelisettysumadhurroyal@gmail.com\"}"
```

Screenshot of the AWS DynamoDB 'Create table' page.

Table details

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
 String
1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
 String
1 to 255 characters and case sensitive.

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS DynamoDB 'List tables' page showing the newly created table.

The EmployeeDetails table was created successfully.

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Total size	Table class
EmployeeDetails	Active	EmailID (\$)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)	0 bytes	Standard

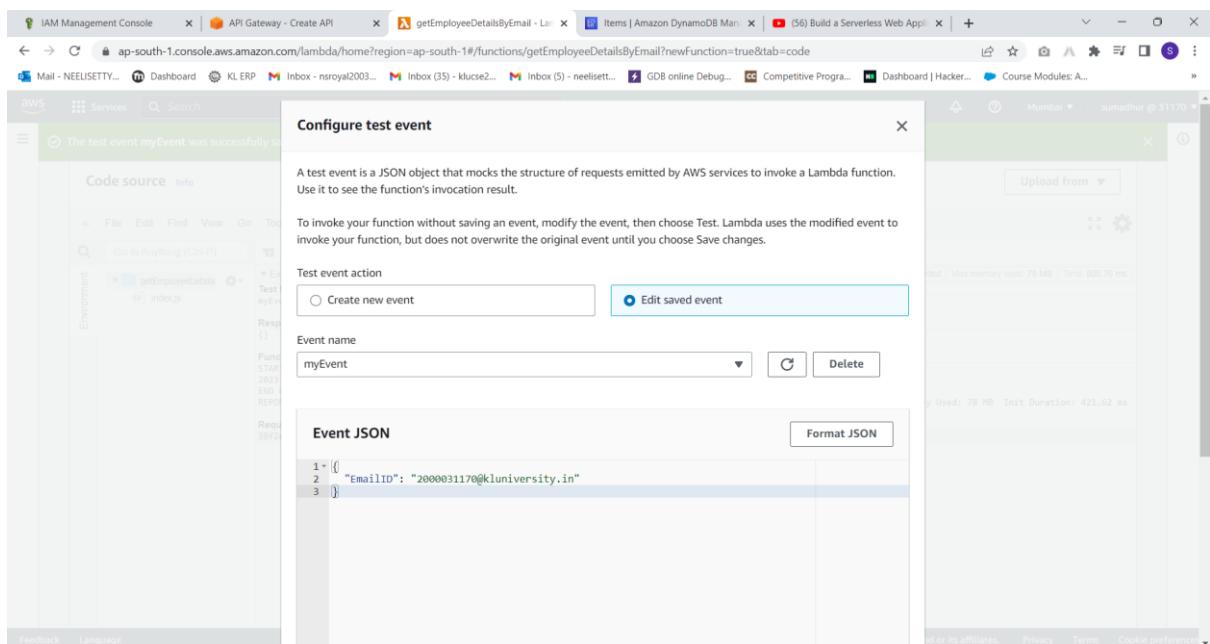
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS DynamoDB console with the 'Edit item | Amazon DynamoDB' tab selected. The URL is https://ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#edit-item?table=EmployeeDetails&route=ROUTE_ITEM_EXPLORER&itemMode=1. The browser window has multiple tabs open, including 'API Gateway - Create API', 'getEmployeeDetailsByEmail - Lambda', '(56) Build a Serverless Web App!', and '(56) Build a Serverless Web App!'. The main content area shows the 'Create item' dialog for the 'EmployeeDetails' table. It includes fields for 'Attribute name' (EmailID, FirstName, LastName) and 'Value' (2000031170@kluniversity.in, Sumadur, Royal). A 'Type' column indicates all values are String. Buttons for 'Cancel' and 'Create item' are at the bottom. The top navigation bar shows 'aws Services Search [Alt+S]' and the user 'sumadur @ 31170'.

The screenshot shows the AWS DynamoDB console with the 'Items | Amazon DynamoDB' tab selected. The URL is <https://ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#item-explorer?table=EmployeeDetails>. The browser window has the same tabs as the previous screenshot. The main content area shows a green success message: 'The item has been saved successfully.' Below it, the 'EmployeeDetails' table is displayed with one item returned. The table structure includes columns for EmailID, FirstName, and LastName. The single item shown is 2000031170@kluniv..., Sumadur, Royal. Buttons for 'Autopreview', 'View table details', 'Actions', and 'Create item' are visible. The top navigation bar shows 'aws Services Search [Alt+S]' and the user 'sumadur @ 31170'.

The screenshot shows the AWS Lambda function editor for a function named "getEmployeeDetailsByEmail". The code is written in JavaScript and uses the AWS SDK to interact with DynamoDB. The function logs the event object and retrieves employee details based on the provided EmailID.

```
const AWS = require('aws-sdk');
var docClient = new AWS.DynamoDB.DocumentClient();
var tablename = 'EmployeeDetails';
exports.handler = (event, context, callback) => {
  console.log(event.EmailID);
  var params = {
    TableName : tablename,
    Key:{
      "EmailID" : event.EmailID
    }
  };
  docClient.get(params, function(err,data){
    if(err)
      callback(err);
    else
      callback(null,data.Item);
  });
};
```



Screenshot of the AWS Lambda function configuration page for the 'getEmployeeDetailsByEmail' function.

The 'Code source' tab is selected. The code file 'index.js' contains the following code:

```
function handler(event) {
  const { EmailID } = event;
  const response = {
    "Item": {
      "EmailID": "2000031170@kluniversity.in",
      "FirstName": "Sumadhur",
      "LastName": "Royal"
    }
  };
  return response;
}
```

The 'Execution result' section shows the following details:

- Test Event Name: myEvent
- Status: Succeeded
- Max memory used: 78 MB
- Time: 931.98 ms

Function Logs:

```
START RequestId: 1ba72279-ca67-490a-a073-ff7ed1c110e1 Version: $LATEST
2023-02-22T11:05:31.100Z 1ba72279-ca67-490a-a073-ff7ed1c110e1 INFO 2000031170@kluniversity.in
END RequestId: 1ba72279-ca67-490a-a073-ff7ed1c110e1
REPORT RequestId: 1ba72279-ca67-490a-a073-ff7ed1c110e1 Duration: 931.98 ms Billed Duration: 932 ms Memory Size: 128 MB Max Memory Used: 78 MB Init Duration: 473.33 ms
1ba72279-ca67-490a-a073-ff7ed1c110e1
```

Request ID: 1ba72279-ca67-490a-a073-ff7ed1c110e1

Screenshot of the AWS API Gateway 'Method Execution' page for the 'GET /getcustomerdetailsbyemail' method.

The left sidebar shows the API structure:

- APIs
- Custom Domain Names
- VPC Links
- API: GetEmployeeDet...
- Resources
- Stages
- Authorizers
- Gateway Responses
- Models
- Resource Policy
- Documentation
- Dashboard
- Settings
- Usage Plans

The 'Actions' dropdown is set to 'Resources'. The 'Resources' section shows the path '/getcustomerdetailsbyemail' with a 'GET' method selected.

The main panel displays the 'Method Execution' configuration for the GET request:

- Method Request:
 - Authorization: NONE
 - Request Validator: NONE
 - API Key Required: false
- URL Query String Parameters:

Name	Required	Caching
EmailID	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add query string
- HTTP Request Headers
- Request Body

Screenshot of the AWS API Gateway - Create API interface showing the Method Execution configuration for the /getcustomerdetailsbyemail - GET method.

API: GetEmployeeDetailsByEmail

Resources / /getcustomerdetailsbyemail **Actions** [Alt+S]

Method Execution /getcustomerdetailsbyemail - GET - Method Request

Provide information about this method's authorization settings and the parameters it can receive.

Settings

Authorization NONE

Request Validator Validate query string parameters and headers

API Key Required false

URL Query String Parameters

Name	Required	Caching
EmailID	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add query string

HTTP Request Headers

Request Body

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS API Gateway - Create API interface showing the Method Execution configuration for the /getcustomerdetailsbyemail - GET method.

Resource Policy

Documentation

Dashboard

Settings

Usage Plans

API Keys

Client Certificates

Settings

Execution role

Invoke with caller credentials

Credentials cache Do not add caller credentials to cache key

Use Default Timeout

URL Path Parameters

URL Query String Parameters

HTTP Headers

Mapping Templates

Request body passthrough

- When no template matches the request Content-Type header
- When there are no templates defined (recommended)
- Never

Content-Type

application/json

Add mapping template

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS API Gateway Mapping Templates configuration for a specific API endpoint. The 'Request body passthrough' option is selected. A mapping template is defined for the 'Content-Type' header set to 'application/json'. The template content is:

```
1 {
2     "EmailID": "$input.params('EmailID')"
3 }
```

The screenshot shows the 'Deploy API' dialog box. The deployment stage is set to 'Dev'. The 'Deployment description' field is empty. The 'Deploy' button is visible at the bottom right. The background shows the API resources list, including the endpoint /getEmployeeDetailsByEmail with a GET method.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Overview POST https://api.telegram.o... GET https://oxa45258ge.ex... + ...

Search Postman

No Environment

Save Send

My Workspace

Collections +

APIs (selected)

Environments

Mock Servers

Monitors

Flows

History

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

Start working with APIs

67%

Next: Save a request. Show me

Find and Replace Console

Status: 400 Bad Request Time: 479 ms Size: 331 B Save Response

Pretty Raw Preview Visualize JSON

```
1: "message": "Missing required request parameters: [EmailID]"
```

Cookies Capture requests Bootcamp Runner Trash

This screenshot shows the Postman application interface. The left sidebar is titled 'My Workspace' and contains sections for Collections, APIs (which is selected), Environments, Mock Servers, Monitors, Flows, and History. The main workspace shows an API request for 'getcustomerdetailsbyemail'. The 'Params' tab is selected, showing a single query parameter 'EmailID' with the value 'Key'. The response status is '400 Bad Request' with a message: 'Missing required request parameters: [EmailID]'. The bottom status bar indicates a time of 479 ms and a size of 331 B.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Overview POST https://api.telegram.o... GET https://oxa45258ge.ex... + ...

Search Postman

No Environment

Save Send

My Workspace

Collections +

APIs (selected)

Environments

Mock Servers

Monitors

Flows

History

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

Start working with APIs

67%

Next: Save a request. Show me

Find and Replace Console

Status: 200 OK Time: 1395 ms Size: 382 B Save Response

Pretty Raw Preview Visualize JSON

```
1: "Item": {  
2:     "EmailID": "2000031170@kluniversity.in",  
3:     "FirstName": "Sumadhur",  
4:     "LastName": "Royal"  
5: }
```

Cookies Capture requests Bootcamp Runner Trash

This screenshot shows the Postman application interface. The left sidebar is identical to the first one. The main workspace shows the same API request for 'getcustomerdetailsbyemail', but now the 'Params' tab shows a query parameter 'EmailID' with the value '2000031170@kluniversity.in'. The response status is '200 OK' with a message: 'Item: { EmailID: "2000031170@kluniversity.in", FirstName: "Sumadhur", LastName: "Royal" }'. The bottom status bar indicates a time of 1395 ms and a size of 382 B.

The screenshot shows the AWS API Gateway console. On the left, the navigation pane is open with the path: APIs > GetEmployeeDetailsByEmail > Resources > /getcustomerdetailsbyemail. The main content area is titled "Enable CORS" and lists several successful steps taken to configure CORS for the GET method on the "/getcustomerdetailsbyemail" resource:

- Create OPTIONS method
- Add 200 Method Response with Empty Response Model to OPTIONS method
- Add Mock Integration to OPTIONS method
- Add 200 Integration Response to OPTIONS method
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Method Response Headers to OPTIONS method
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Integration Response Header Mappings to OPTIONS method
- Add Access-Control-Allow-Origin Method Response Header to GET method
- Add Access-Control-Allow-Origin Integration Response Header Mapping to GET method

At the bottom of the page, there are links for Feedback, Language, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS S3 console. The navigation pane indicates the user is at Amazon S3 > Buckets > Create bucket. The main form is titled "Create bucket" and includes the following fields:

- General configuration**:
 - Bucket name: sumadur-31170
 - AWS Region: Asia Pacific (Mumbai) ap-south-1
 - Copy settings from existing bucket - optional: Choose bucket (button)
- Object Ownership**:
 - Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
 - ACLs disabled (recommended) (radio button)
 - ACLs enabled (radio button, selected)

At the bottom of the page, there are links for Feedback, Language, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'sumadhur-31170'". Below it, the "Buckets" section displays an account snapshot and a list of buckets. The bucket "sumadhur-31170" is listed with the following details:

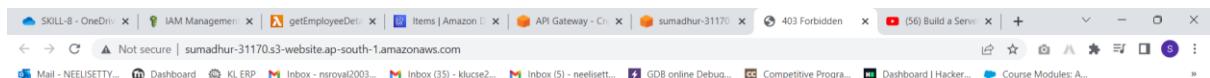
Name	AWS Region	Access	Creation date
sumadhur-31170	Asia Pacific (Mumbai) ap-south-1	Objects can be public	February 22, 2023, 17:03:12 (UTC+05:30)

The screenshot shows the AWS S3 console with the "Upload" interface for the "sumadhur-31170" bucket. The "Files and folders" section lists three files: "index.html", "jquery-3.1.1.min.js", and "knockout-3.4.2.js". The "Destination" section shows the target path as "s3://sumadhur-31170".

Name	Folder	Type	Size
index.html	-	text/html	2.3 KB
jquery-3.1.1.min.js	-	text/javascript	84.7 KB
knockout-3.4.2.js	-	text/javascript	59.1 KB

The screenshot shows the AWS S3 console interface. At the top, there's a green success message: "Upload succeeded" with a link to "View details below". Below this, the "Upload: status" section indicates "Succeeded" for 3 files (146.0 KB) and "Failed" for 0 files (0 B). The "Files and folders" tab is selected, showing a table with one item: "index.html" (text/html, 2.5 KB, Succeeded). The bottom of the page includes standard AWS footer links for Feedback, Language, Privacy, Terms, and Cookie preferences.

The screenshot shows the "Edit static website hosting" configuration page for the "sumadhur-31170" bucket. Under "Static website hosting", the "Enable" option is selected. Under "Hosting type", the "Host a static website" option is selected. A note explains that content must be publicly readable. The "Index document" field is set to "index.html". The left sidebar shows the "Amazon S3" navigation menu with options like Buckets, Storage Lens, and AWS Marketplace for S3.



403 Forbidden

- Code: AccessDenied
- Message: Access denied
- RequestId: TG6XYVW1WDZS6ESFG
- HostId: tirTsWCqaMOE0d5XgjdGZh4wkHJWeh4cp3C8aVFdh8LoV3WWJ4s+pviC/gORHealauYi671Nnc=

The screenshot shows the AWS S3 console with the 'Edit bucket policy' page for the 'sumadhur-31170' bucket. The policy is defined as follows:

```
1 * {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "Statement1",
6             "Action": ["s3:GetObject"],
7             "Effect": "Allow",
8             "Resource": "arn:aws:s3:::sumadhur-31170/*",
9             "Principal": "*"
10        }
11    ]
12 }
```

The right side of the screen shows a modal titled 'Edit statement' with the heading 'Select a statement'. It contains the instruction 'Select an existing statement in the policy or add a new statement.' and a button '+ Add new statement'.

