

2000031668

K Lohith

## SEC-13 SKILL-9

### POST DETAILS of EMPLOYEE

The screenshot shows the AWS API Gateway console. The top navigation bar includes links for ERP, LMS, OUTLOOK, GMAIL, I LOVE PDF, CERTIFICATIONS, RS, MSWD, Dell, Unschool, React App, and AI. The main interface displays a list of APIs under the 'APIs' tab. One API, 'GetEmployeeDetailsByEmail', is listed with the following details:

Name	ID	Protocol	Endpoint type	Created
GetEmployeeDetailsByEmail	goxl5wlgoh	REST	Regional	2023-03-02

Below this, there is a 'Create API' button.

The screenshot shows the 'Create API' wizard. The first step, 'Choose the protocol', is displayed. It asks whether you want to create a REST API or a WebSocket API. The 'REST' option is selected. Below this, there are sections for 'Create new API' (describing a REST API as a collection of resources and methods) and 'Settings' (where the API name is set to 'PostCustomerDetails', the description is empty, and the endpoint type is 'Edge optimized'). A 'Create API' button is at the bottom right.

The screenshot shows the second step of the 'Create API' wizard, 'Create new API'. It shows the 'PostCustomerDetails' API configuration. The 'API name\*' field is filled with 'PostCustomerDetails', the 'Description' field is empty, and the 'Endpoint Type' is set to 'Edge optimized'. A note at the bottom left says '\* Required'. A 'Create API' button is at the bottom right.

API Gateway - Create API postcuster.docx (552) Build a Serverless Web Ap... postcustomerdetails-part2 - One ...

https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#apis/d70fsbtvnd/resources/9v0wdgjipa/create

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI Other favorites

aws Services Search [Alt+S]

Amazon API Gateway APIs > PostCustomerDetails (d70fsbtvnd) > Resources > / (9v0wdgjipa) > Create

Hide hints ?

APIs Resources Actions New Child Resource

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

Configure as proxy resource  Resource Name\* postCustomerDetails

Resource Path\* /postcustomerdetails

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

Create Resource

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 5:49 PM 3/8/2023

API Gateway - Create API postcuster.docx (552) Build a Serverless Web Ap... postcustomerdetails-part2 - One ...

https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#apis/d70fsbtvnd/resources/babhfj/methods/POST

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI Other favorites

aws Services Search [Alt+S]

Amazon API Gateway APIs > PostCustomerDetails (d70fsbtvnd) > Resources > /postcustomerdetails (babhfj) > POST

Hide hints ?

APIs Resources Actions /postcustomerdetails - POST - Setup

Choose the integration point for your new method.

Integration type  Lambda Function  HTTP  Mock  AWS Service  VPC Link

Save

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 5:50 PM 3/8/2023

API Gateway - Create API postcuster.docx (552) Build a Serverless Web Ap... postcustomerdetails-part2 - One ...

https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#apis/d70fsbtvnd/resources/babhfj/methods/POST

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI Other favorites N. Virginia Lohith.K

Amazon API Gateway APIs > PostCustomerDetails (d70fsbtvnd) > Resources > /postcustomerdetails (babhfj) > POST

APIs Services Search [Alt+S]

Method Execution /postcustomerdetails - POST - Integration Response

First, declare response types using Method Response. Then, map the possible responses from the backend to this method's response types.

HTTP status regex	Method response status	Output model	Default mapping
-	200	-	Yes

Map the output from your HTTP endpoint to the headers and output model of the 200 method response.

HTTP status regex default Content handling Passthrough

Header Mappings Mapping Templates Add integration response

Cancel Save

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 5:50 PM 3/8/2023

API Gateway - Create API postcuster.docx (552) Build a Serverless Web Ap... postcustomerdetails-part2 - One ...

https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#apis/d70fsbtvnd/resources/babhfj/methods/POST

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI Other favorites N. Virginia Lohith.K

Amazon API Gateway APIs > PostCustomerDetails (d70fsbtvnd) > Resources > /postcustomerdetails (babhfj) > POST

APIs Services Search [Alt+S]

Method Execution /postcustomerdetails - POST - Integration Response

First, declare response types using Method Response. Then, map the possible responses from the backend to this method's response types.

HTTP status regex	Method response status	Output model	Default mapping
-	200	-	Yes

Map the output from your HTTP endpoint to the headers and output model of the 200 method response.

HTTP status regex default Content handling Passthrough

Header Mappings Mapping Templates

Content-Type application/json

application/json Generate template: 1 { "StatusCode" : "200" }

Add mapping template

Cancel Save

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 5:51 PM 3/8/2023

The screenshot shows two overlapping windows from the AWS API Gateway console.

**Top Window: Method Execution**

- Left Sidebar:** Shows the API named "PostCustomerDetails".
- Resources:** A tree view showing the root path "/" with a single child resource "postcustomerdetails". This resource has three methods: OPTIONS, POST, and another unnamed method.
- Actions:** A dropdown menu with options like "Test", "Edit", "Delete", etc.
- Method Execution /postcustomerdetails - POST - Method Test:**
  - Request:** Path: /postcustomerdetails, Status: 200, Latency: 10 ms, Response Body: {"Status": "Success"}
  - Response Headers:** Content-Type: application/json
  - Logs:** Displays the execution log for the request, showing the flow from the request path to the successful response.

**Bottom Window: Deploy API**

- Left Sidebar:** Same as the top window.
- Actions:** A dropdown menu with options like "Test", "Edit", "Delete", etc.
- Deploy API Dialog:**
  - Deployment stage:** [New Stage] (dropdown menu)
  - Stage name\***: dev
  - Stage description**: (text input)
  - Deployment description**: (text input)
  - Buttons:** Cancel, Deploy
- Logs:** Displays the deployment log for the request, showing the flow from the deployment stage to the successful response.

The screenshot displays two side-by-side browser windows. The top window is the AWS API Gateway 'Create API' interface, specifically the 'Stages' section for the 'dev' stage of an API named 'PostCustomerDetails'. It shows settings for API cache, method throttling (Rate: 10000 requests per second, Burst: 5000 requests), and a Web Application Firewall (WAF) configuration. The bottom window is the Postman application, which is running a POST request to the same endpoint. The Postman interface includes a 'Params' tab with a 'Query Params' table, a 'Body' tab showing a JSON response with a StatusCode of 200, and various other tabs like Cookies, Headers, and Test Results.

API Gateway - Create API | postcustomer.docx | (552) Build a Serverless Web Ap... | postcustomerdetails-part2 - One... | +

https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#apis/d70fsbtvnd/stages/dev

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI

aws Services Search [Alt+S]

Amazon API Gateway APIs > PostCustomerDetails (d70fsbtvnd) > Stages > dev

dev Stage Editor

Delete Stage Configure Tags

APIs Stages Create

Custom Domain Names VPC Links

API: PostCustomerDe...

Resources Stages

Authorizers Gateway Responses Models

Resource Policy Documentation Dashboard Settings

Usage Plans API Keys Client Certificates

Feedback Language 33°C Partly sunny

33°C Partly sunny

API Ga Home Workspaces API Network Explore

My Workspace New Import

https://d70fsbtvnd.execute-api.us-east-1.amazonaws.com/dev/postcustomerdetails

POST https://d70fsbtvnd.execute-api.us-east-1.amazonaws.com/dev/postcustomerdetails Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

1 "StatusCode": "200"

2

3

200 OK 1168 ms 443 B Save Response

Feedback Language 33°C Partly sunny

33°C Partly sunny

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

**Step 1: Select trusted entity**

**Trusted entity type**

- AWS service: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Common use cases**

- EC2: Allows EC2 instances to call AWS services on your behalf.
- Lambda: Allows Lambda functions to call AWS services on your behalf.

**Use cases for other AWS services:**  
Choose a service to view use case

**Next**

**Feedback Language**   **33°C Partly sunny**   **Q Search**   **5:54 PM 3/8/2023**

Screenshot of the AWS IAM Management Console showing the "Create role" wizard - Step 2: Name, review, and create.

**Name, review, and create**

**Role details**

**Role name:** postcustomerdetails

**Description:** Allows Lambda functions to call AWS services on your behalf.

**Step 1: Select trusted entities**

```

1+ [{
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole"
7     },
8     {
9       "Principal": [
10         "lambda.amazonaws.com"
11     ]
12   }
13 }
14 ]
  
```

**Feedback Language**   **33°C Partly sunny**   **Q Search**   **5:55 PM 3/8/2023**

**Step 2: Add permissions**

Permissions policy summary

Policy name	Type	Attached as
AWSLambda_FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	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.

**Create role**

**Create function - Lambda**

Function name: savecustomerdetails

Runtime: Node.js 14.x

Architecture: x86\_64

Permissions: postCustomerDetails

Change default execution role:

- Execution role: Create a new role with basic Lambda permissions (selected)
- Existing role: postCustomerDetails

The event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

```
1+ [
2 "EmailID": "kllohit2003@gmail.com",
3 "FirstName": "Kanthury",
4 "LastName": "Lohith"
5 ]
```

Format JSON

Code Test Monitor Configuration

Code source Info

File Edit Find View Go Tools Window

Go to Anything (Ctrl-P)

savecustomerdetails

index.js

Cancel Save

Environment

33°C Partly sunny

aws Services Search [Alt+S]

The test event event was successfully saved.

Function ARN

arn:aws:lambda:us-east-1:543828056217:function:savecustomerdetails

Function URL [Info](#)

Code Test Monitor Configuration Aliases Versions

Code source Info

File Edit Find View Go Tools Window

Test Deploy

Execution results

Test Event Name event

Response

```
{
  "statusCode": 200,
  "body": "\nHello from Lambda!\n"
}
```

Function Logs

START RequestId: 1412be27-82a6-4e95-96c0-ce97a5686f98 Version: \$LATEST

END RequestId: 1412be27-82a6-4e95-96c0-ce97a5686f98

REPORT RequestId: 1412be27-82a6-4e95-96c0-ce97a5686f98 Duration: 14.73 ms Billed Duration: 15 ms Memory Size: 128 MB Max Memory Used: 57 MB Init Duration: 164.92 ms

Request ID

1412be27-82a6-4e95-96c0-ce97a5686f98

Feedback Language

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

33°C Partly sunny

5:57 PM 3/8/2023

The screenshot shows two consecutive screenshots of a web browser displaying AWS services.

**Screenshot 1: Create table - DynamoDB**

This screenshot shows the "Create table" page in the AWS DynamoDB console. The URL is <https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table>.

**Table details:**

- Table name:** PostCustomerDetails (This will be used to identify your table. Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)).
- Partition key:** EmailID (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 type).
- Sort key - optional:** Enter the sort key name (String type). (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. 1 to 255 characters and case sensitive.)

**Table settings:**

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 33°C Partly sunny 6:06 PM 3/8/2023

**Screenshot 2: Lambda function configuration**

This screenshot shows the "Items | Amazon DynamoDB Man" page in the AWS Lambda console. The URL is <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/savecustomerdetails?newFunction=true&tab=code>.

**Code source:**

```

Code | Test | Monitor | Configuration | Aliases | Versions
Code source | Info
File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P) index.js Execution result: 
Test Event Name event
Environment Status: Failed | Max memory used: 78 MB | Time: 613.23 ms
Response
{
  "errorType": "AccessDeniedException",
  "errorMessage": "User: arn:aws:sts::543828856217:assumed-role/postCustomerDetails/savecustomerdetails is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:9266a086-2c46-4c51-a944-8046066874fb",
  "trace": [
    "AccessDeniedException: User: arn:aws:sts::543828856217:assumed-role/postCustomerDetails/savecustomerdetails is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:9266a086-2c46-4c51-a944-8046066874fb",
    "at Request.callListeners (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:106:20)",
    "at Request.emit (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:78:10)",
    "at Request.emit (/var/runtime/node_modules/aws-sdk/lib/request.js:688:14)",
    "at Request.transition (/var/runtime/node_modules/aws-sdk/lib/request.js:22:10)",
    "at AcceptorStateMachine.runTo (/var/runtime/node_modules/aws-sdk/lib/state_machine.js:14:12)",
    "at /var/runtime/node_modules/aws-sdk/lib/state_machine.js:26:10",
    "at Request.transition (/var/runtime/node_modules/aws-sdk/lib/request.js:38:9)",
    "at Request._execute (/var/runtime/node_modules/aws-sdk/lib/request.js:688:12)",
    "at Request.callListeners (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:116:18)"
  ]
}
Function Logs
START RequestId: 9266a086-2c46-4c51-a944-8046066874fb Version: $LATEST
2023-03-08T12:41:16.510Z 9266a086-2c46-4c51-a944-8046066874fb ERROR Invoke Error {"errorType": "AccessDeniedException", "errorMessage": "User: arn:aws:sts::543828856217:assumed-role/postCustomerDetails/savecustomerdetails is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:9266a086-2c46-4c51-a944-8046066874fb", "trace": [{}]}
END RequestId: 9266a086-2c46-4c51-a944-8046066874fb
REPORT RequestId: 9266a086-2c46-4c51-a944-8046066874fb Duration: 4 ms Memory Size: 128 MB Max Memory Used: 78 MB Init Duration: 450.42 ms
Request ID: 9266a086-2c46-4c51-a944-8046066874fb
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 33°C Partly sunny 6:11 PM 3/8/2023

```

Screenshot of the AWS Lambda function configuration page showing the successful update of the 'savecustomerdetails' function.

The Lambda function code (index.js) is displayed:

```
var AWS = require("aws-sdk");
var docClient = new AWS.DynamoDB.DocumentClient();
exports.handler = (event, context, callback) => {
  var tablename = "CustomerDetails";
  var params = {
    TableName : tableName,
    Item: {
      "EmailID" : event.EmailID,
      "Firstname" : event.Firstname,
      "Lastname" : event.Lastname
    }
  };
  docClient.put(params, function(err,data){
    if(err){
      callback(err, data);
    }else{
      callback(null,"successfully updated data");
    }
  });
};
```

The screenshot shows the Lambda function successfully updated message at the top.

Screenshot of the Amazon DynamoDB Items page showing the successful save of a customer detail item.

The item details are as follows:

EmailID	Firstname	Lastname
kloith2005@gmail.com	kloith	lohit

Screenshot of the AWS Lambda console showing the successful update of the function "savecustomerdetails".

The Lambda function has been updated successfully. A test event named "event" has been created, containing the following JSON payload:

```
1 < [{"EmailID": "klohit200@gmail.com",  
2   "FirstName": "klohit",  
3   "LastName": "lohit"},  
4 ]]
```

Screenshot of the IAM Management Console showing the successful attachment of a policy to a role.

The "postCustomerDetails" role now has the "Policy was successfully attached to role." message displayed. The role summary shows the following details:

- Creation date: February 23, 2023, 10:21 (UTC+05:30)
- Last activity: 13 days ago
- ARN: arn:aws:iam:543828056217:role/postCustomerDetails
- Maximum session duration: 1 hour

The "Permissions" tab is selected, showing three managed policies attached to the role:

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS M...
AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB via t...
AWSLambda_FullAccess	AWS managed	Grants full access to AWS Lambda service. AWS...

The screenshot shows the AWS Lambda console interface for creating a new function named "savescustomerdetails".

**Basic information:**

- Function name:** savescustomerdetails
- Runtime:** Node.js 16.x
- Architecture:** x86\_64
- Permissions:** postCustomerDetails role

**Execution role:**

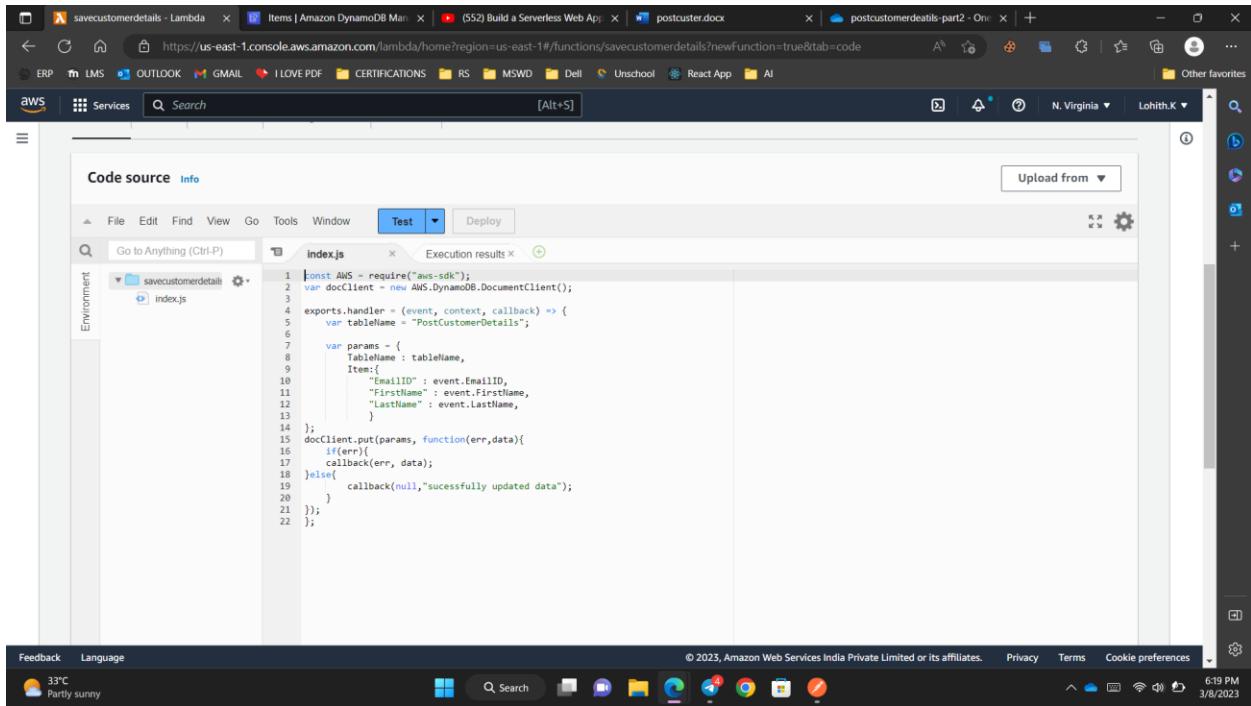
- Use an existing role: Use an existing role

**Existing role:** postCustomerDetails role

**Code source:**

- File: index.js
- Execution results:
- Test Event Name: event
- Response: "Successfully updated the data"
- Request ID: 3cc6619e-7124-40e7-8c07-01b98721f7f8
- Function Log:

```
START RequestId: 3cc6619e-7124-40e7-8c07-01b98721f7f8 Version: $LATEST
END RequestId: 3cc6619e-7124-40e7-8c07-01b98721f7f8
REPORT RequestId: 3cc6619e-7124-40e7-8c07-01b98721f7f8 Duration: 973.88 ms Billed Duration: 974 ms Memory Size: 128 MB Max Memory Used: 83 MB Init Duration: 441.04 ms
Request ID: 3cc6619e-7124-40e7-8c07-01b98721f7f8
```



```
const AWS = require('aws-sdk');
```

```
var docClient = new AWS.DynamoDB.DocumentClient();
```

```
var tableName = "CustomerDetails";
```

```
exports.handler = (event, context, callback) => {
```

```
  console.log(event.EmailID)
```

```
  var params = {
```

```
    TableName : tableName,
```

```
  Key:{
```

```
    "EmailID" : event.EmailID
```

```
 }
```

```
}
```

```
docClient.get(params, function(err,data){
```

```
    callback(err, data);
```

```
})
```

```
};
```

The screenshot shows the AWS Lambda console interface for testing a function. The top navigation bar includes tabs for 'Lambda' and 'Services'. The main area is titled 'Test event' with a 'Info' tab selected. A note states: 'To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.' Below this, there are two options for 'Test event action': 'Create new event' (unchecked) and 'Edit saved event' (checked). The 'Event name' field contains 'event'. The 'Event JSON' section displays the following JSON code:

```
1 {  
2   "EmailID": "sai@gmail.com",  
3   "Firstname": "sai",  
4   "LastName": "SAI"  
5 }
```

At the bottom of the interface, there are links for 'Feedback', 'Language', and 'Cookie preferences', along with system status icons like weather and battery level. The footer includes copyright information for Amazon Web Services and navigation links for 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar menu for 'DynamoDB' is visible with options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, and Settings. Under 'Explore items', 'PostCustomerDetails' is selected. The main panel has a search bar at the top and a section titled 'Scan or query items' with 'Scan' selected. It shows 'Completed. Read capacity units consumed: 0.5'. Below this, a table titled 'Items returned (1)' displays one item:

EmailID	FirstName	LastName
klohit2003@gmail.c...	klohit	lohith

At the bottom of the browser window, there is a Windows taskbar with various pinned icons and the system clock showing 6:20 PM on 3/8/2023.

The second part of the screenshot shows the same setup after another item has been added to the table. The 'Items returned' count is now 2, and the table now shows two items:

EmailID	FirstName	LastName
sai@gmail.com	sai	SAI
klohit2003@gmail.c...	klohit	lohith

The system clock at the bottom now shows 6:21 PM on 3/8/2023.

The screenshot shows two consecutive screenshots of the AWS API Gateway Console interface, illustrating the configuration of a POST method for a Lambda function.

**Screenshot 1: Method Execution Configuration**

This screenshot shows the configuration of the POST method for the '/postcustomerdetails' resource. The integration type is set to 'Lambda Function' (selected via radio buttons), and the Lambda function is 'savecustomerdetails'. The Lambda Region is 'us-east-1'. The 'Save' button is visible at the bottom right.

**Screenshot 2: Method Execution Details**

This screenshot shows the detailed execution configuration for the POST method. It displays the Method Request (Auth: NONE, ARN: arn:aws:execute-api:us-east-1:543828056217:d70fsbtvnd/\*/POST/), Integration Request (Type: LAMBDA, Region: us-east-1), Method Response (HTTP Status: 200, Models: application/json => Empty), and Integration Response (HTTP status pattern: -, Output passthrough: No). The Lambda function name 'savecustomerdetails' is also visible on the right side of the interface.

The screenshot shows the AWS API Gateway Console interface. On the left, a sidebar navigation bar includes links for APIs, Custom Domain Names, VPC Links, and various API management sections like Resources, Stages, Authorizers, and Models. The main content area is titled "Enable CORS" for the "/postcustomerdetails" resource under the "PostCustomerDetails API". It shows the "Gateway Responses for PostCustomerDetails API" section with options for DEFAULT 4XX and DEFAULT 5XX. Under the "Methods" section, "POST" is selected, and the "Access-Control-Allow-Methods" header is set to "OPTIONS, POST". The "Access-Control-Allow-Headers" header is set to "Content-Type,X-Amz-Date,Authorization". The "Access-Control-Allow-Origin" header is set to "\*". A blue button at the bottom right says "Enable CORS and replace existing CORS headers".

The screenshot shows the Postman application interface. The left sidebar lists collections, environments, and other workspace items. The main panel shows a POST request to the URL "https://d70fsbtvnd.execute-api.us-east-1.amazonaws.com/dev/postcustomerdetails". The "Body" tab is selected, showing a JSON payload:

```
1 {"EmailID": "123@gmail.com",  
2 "FirstName": "sai",  
3 "LastName": "SAI"}
```

The "Headers" tab shows the following headers:

Header	Value
Content-Type	application/json
Accept	application/json

The "Test Results" tab shows a successful response with status 200 OK, 1852 ms, and 546 B.

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar menu for 'DynamoDB' includes options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, Clusters, Subnet groups, Parameter groups, and Events. The 'Explore items' section is currently selected. In the main area, a table named 'PostCustomerDetails' is displayed. The table has columns: EmailID, FirstName, and LastName. Three items are listed:

EmailID	FirstName	LastName
sai@gmail.com	sai	SAI
123@gmail.com	sai	SAI
klohit2003@gmail.c...	lohit	lohit

A green notification bar at the bottom indicates: 'Completed. Read capacity units consumed: 0.5'. Below the table, there are 'Actions' and 'Create item' buttons.

The screenshot shows the AWS S3 Bucket Properties page for a bucket named "postcustomer". The "Block All Public Access" section is open, displaying four options under "Block all public access":

- Block public access to buckets and objects granted through new access control lists (ACLs)**: Describes how S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects.
- Block public access to buckets and objects granted through any access control lists (ACLS)**: Describes how S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**: Describes how S3 will block new bucket and access point policies that grant public access to buckets and objects.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: Describes how S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

A warning message states: "Turning off block all public access might result in this bucket and the objects within becoming public". A checkbox is checked, acknowledging this risk.

**Upcoming permission changes to disable any Block Public Access setting**

The screenshot then transitions to the AWS S3 Management Console, specifically the "Upload" page for a bucket named "31668postcustomerdetails". The "Upload" tab is selected. A large input field is present for dragging and dropping files. Below it, a table lists three files:

Name	Type	Size
index.html	text/html	2.2 KB
jquery-3.1.1.min.js	text/javascript	84.7 KB
knockout-3.4.2.js	text/javascript	59.1 KB

The "Destination" section shows the path "s3://31668postcustomerdetails".

The screenshot shows two stacked dialog boxes from the AWS S3 Management Console.

The top dialog is titled "Grant basic read/write permissions to other AWS accounts". It contains a note about using S3 bucket policies or IAM policies for access control, followed by a section for "Access control list (ACL)". The "Choose from predefined ACLs" option is selected. A warning message states: "Granting public-read access is not recommended. Anyone in the world will be able to access the specified objects." A checkbox at the bottom is checked, stating: "I understand the risk of granting public-read access to the specified objects."

The bottom dialog is titled "Static website hosting". It has a note: "Use this bucket to host a website or redirect requests." Under "Hosting type", the "Host a static website" option is selected. A note explains: "For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access." Below this, there are fields for "Index document" (set to "index.html") and "Error document - optional" (also set to "index.html").

## Index.html

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8" />
<script src="jquery-3.1.1.min.js"></script>
```

```

<script src="knockout-3.4.2.js"></script>
<script type="text/javascript">

    jQuery.support.cors = true;

    $(document).ready(function() {

        var customerDetailsViewModel = function() {
            var self = this;

            self.EmailID = ko.observable("");
            self.FirstName = ko.observable("");
            self.LastName = ko.observable("");
            self.SuccessMessage = ko.observable("");
            self.savecustomerdetails = function () {

                var CustomerDetail = {

                    EmailID: self.EmailID(),
                    FirstName: self.FirstName(),
                    LastName: self.LastName()

                }

                $.ajax({
                    crossDomain: true,

                    url: 'https://d70fsbtvnd.execute-api.us-east-1.amazonaws.com/dev/postcustomerdetails',

                    cache: false,
                    type: 'POST',
                    data: ko.toJSON(CustomerDetail),
                    success: function (data) {
                        self.SuccessMessage(data)
                            self.EmailID("");
                            self.FirstName("");
                            self.LastName("");
                    }
                }).fail(
                    function(xhr, textStatus, err){
                        alert("Error happened "+err);
                    }
                );
            }
        };
    });

```

```

        });
    };
}

var viewModel = new customerDetailsViewModel();
ko.applyBindings(viewModel);
});

</script>
</head>
<body>
<table>
<tr>
<td>
<div>
<p>
    EmailID:
    <input data-bind='value: EmailID' />

    </p>
    <p>
        FirstName:
        <input data-bind='value: FirstName' />
    </p>
    <p>
        LastName:
        <input data-bind='value: LastName' />
    </p>

    </div>
    <span style ="color: darkolivegreen" data-bind='text : SuccessMessage' />

</td>
</tr>
<tr>
<td>
<div>
    <button data-bind="click:
        $root.savecustomerdetails" > Save </button>
</div>
</td>
</tr>

```

```
</table>
```

```
</body>  
</html>
```

The screenshot shows a Microsoft Edge browser window with two tabs open. The top tab is titled "Edit bucket policy" and is located in the AWS S3 console. It displays a JSON policy document:

```
1  {  
2   "Id": "Policy1678281944848",  
3   "Version": "2012-10-17",  
4   "Statement": [  
5     {  
6       "Sid": "Stmt1678281943353",  
7       "Action": [  
8         "s3:GetObject"  
9       ],  
10      "Effect": "Allow",  
11      "Resource": "arn:aws:s3:::31668postcustomerdetails/*",  
12      "Principal": "*"  
13    }  
14  ]  
15 }
```

The bottom tab shows a form with three fields: EmailID, FirstName, and LastName, all containing the value "hi". Below the form is a "Save" button.

The browser status bar at the bottom of the screen shows the date as 3/8/2023 and the time as 6:56 PM.

The screenshot displays two side-by-side application windows. The left window is Postman, a tool for testing APIs. It shows a collection named 'My first collection' containing two folders: 'First folder inside collection' and 'Second folder inside collection'. A sub-request titled 'index.h' is selected, with its URL being `https://d70fsbtvnd.execute-api.us-east-1.amazonaws.com/dev/postcustomerdetails`. The 'Body' tab shows a JSON payload:

```
1 "EmailID": "123@gmail.com",
2 "FirstName": "sai",
3 "LastName": "SAI"
```

The response status is 200 OK, with a timestamp of 7:02 PM on 3/8/2023. The right window is the AWS DynamoDB console, specifically the 'CustomerDetails' table. It shows the results of a query with the message: 'Completed. Read capacity units consumed: 0.5'. The table data is as follows:

EmailID	FirstName	LastName
sai@gmail.com	sai	SAI
123@gmail.com	sai	SAI
klohith2003@gmail.c...	klohith	lohit

The screenshot displays two browser windows side-by-side. The left window shows the AWS DynamoDB console with the 'PostCustomerDetails' table selected. The right window shows a browser displaying the saved customer details.

**AWS DynamoDB Console (Left Window):**

- Left Sidebar:** Shows navigation links like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events.
- Main Area:** Shows the 'Tables (2)' section with 'CustomerDetails' and 'PostCustomerDetails'. The 'PostCustomerDetails' table is selected. The 'Scan or query items' section is open, showing 'Scan' selected, 'Table - PostCustomerDetails' selected, and 'All attributes' for attribute projection. A message indicates 'Completed. Read capacity units consumed: 0.5'. Below this, a list titled 'Items returned (3)' shows three items.

**Browser Window (Right Window):**

- Address Bar:** https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#item-explorer?table=PostCustomerDetails&maximize=true
- Content:** Displays the saved customer details. The form fields are:
  - EmailID: hihelo@gmail.com
  - FirstName: Hi
  - LastName: HeloA 'Save' button is at the bottom.

Not secure | 31668postcustomerdetails.s3-website-us-east-1.amazonaws.com

EmailID:

FirstName:

LastName:

26°C Partly sunny

AWS Management Console | 31668postcustomerdetails | Items | Amazon DynamoDB | savecustomerdetails - Lambda | API Gateway Console

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI

Search

9:29 AM 3/9/2023

Not secure | 31668postcustomerdetails.s3-website-us-east-1.amazonaws.com

EmailID:  2000031668cse@gmail.com

FirstName:  Kanthury

LastName:  Lohith

26°C Partly sunny

AWS Management Console | 31668postcustomerdetails | Items | Amazon DynamoDB | savecustomerdetails - Lambda | API Gateway Console

ERP LMS OUTLOOK GMAIL I LOVE PDF CERTIFICATIONS RS MSWD Dell Unschool React App AI

Search

9:29 AM 3/9/2023

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Items | Amazon DynamoDB" and displays a form for saving customer details to a table named "PostCustomerDetails". The form fields are:

- EmailID: [Text input]
- FirstName: [Text input]
- LastName: [Text input]

Below the form, a message says "successfully updated data" and there is a "Save" button.

At the bottom of the browser window, the AWS Management Console sidebar is visible, showing the "DynamoDB" service selected. The "Tables" section lists "CustomerDetails" and "PostCustomerDetails", with "PostCustomerDetails" currently selected. The main pane shows the results of a scan or query, with a message indicating "Completed. Read capacity units consumed: 0.5". The results table contains the following data:

EmailID	FirstName	LastName
sai@gmail.com	sai	SAI
31170@gmail.com	sumadhur	royal
12@gmail.com	lohit	kanthury
2000031668cse@gm...	Kanthury	Lohith
123@gmail.com	sai	SAI
klohit2003@gmail.c...	lohit	lohit

