# Serverless Backend API – Troubleshooting Scenarios & Fixes

This document captures **real-world troubleshooting scenarios** for the Serverless Backend API built using **Lambda, DynamoDB, API Gateway, IAM, and CloudFormation**.

Each scenario includes: - Symptom - Root cause - How to identify - Step-by-step fix - Interview explanation

---

## Scenario 1️⃣ API returns "Internal Server Error" (500)

### Symptom

```
{
  "message": "Internal server error"
}
```

### Root Cause

Lambda function failed during execution.

### How to Identify

1. Go to **Lambda → Monitor → CloudWatch Logs**
2. Open latest log stream

### Fix

- Read error message
- Fix code / permissions
- Redeploy API Gateway

### Interview Explanation

"A 500 error usually indicates a Lambda runtime failure. I debug it using CloudWatch logs to identify the exact exception."

---

## Scenario 2️⃣ DynamoDB Decimal Serialization Error

### Symptom

```
TypeError: Object of type Decimal is not JSON serializable
```

### Root Cause

DynamoDB returns numbers as `Decimal`, which cannot be directly converted to JSON.

### How to Identify

CloudWatch log shows `TypeError` during `json.dumps()`.

### Fix

Convert Decimal values to native Python types before returning response.

```python
from decimal import Decimal

def decimal_to_native(obj):
    if isinstance(obj, Decimal):
        return int(obj)
    if isinstance(obj, dict):
        return {k: decimal_to_native(v) for k, v in obj.items()}
    if isinstance(obj, list):
        return [decimal_to_native(i) for i in obj]
    return obj
```

### Interview Explanation

"DynamoDB uses Decimal to preserve precision, so I convert it before JSON serialization."

---

## Scenario 3️⃣ 403 Forbidden Error

### Symptom

```
{"message": "Forbidden"}
```

### Root Cause

API Gateway stage not deployed or missing Lambda invoke permission.

### How to Identify

- API Gateway test works
- External request fails

### Fix

1. Redeploy API Gateway stage
2. Verify `AWS::Lambda::Permission` exists

### Interview Explanation

"403 errors often indicate missing API Gateway deployment or Lambda invoke permissions."

---

## Scenario 4️⃣ Empty Response from GET /product/{id}

### Symptom

```
{}
```

### Root Cause

Incorrect product ID or item does not exist in DynamoDB.

### How to Identify

Check DynamoDB table for matching ID.

### Fix

- Use correct UUID
- Add validation and 404 handling (optional)

### Interview Explanation

"An empty response means the item was not found. I verify IDs against DynamoDB."

---

## Scenario 5️⃣ Lambda Timeout

### Symptom

```
Task timed out after X seconds
```

### Root Cause

Lambda timeout too low or inefficient code.

### How to Identify

CloudWatch logs show timeout error.

### Fix

- Increase timeout (e.g., 5 sec)
- Optimize DynamoDB operations

### Interview Explanation

"Timeouts indicate slow execution. I optimize code or adjust timeout accordingly."

---

## Scenario 6️⃣ AccessDeniedException (DynamoDB)

### Symptom

```
AccessDeniedException: User is not authorized to perform dynamodb:Scan
```

### Root Cause

IAM role missing required DynamoDB permissions.

### How to Identify

CloudWatch logs show `AccessDeniedException`.

### Fix

Attach required DynamoDB permissions to Lambda execution role.

### Interview Explanation

"I resolved it by applying least-privilege IAM permissions for DynamoDB actions."

---

## Scenario 7️⃣ API Works in Console but Not in Browser

### Symptom

API works via Postman but fails in browser.

### Root Cause

CORS headers missing in Lambda response.

### How to Identify

Browser console shows CORS error.

### Fix

Add headers to Lambda response:

```
{
  "Access-Control-Allow-Origin": "*"
}
```

### Interview Explanation

"Browsers enforce CORS, so I added appropriate headers in Lambda responses."

---

## Scenario 8️⃣ API Changes Not Reflected

### Symptom

Old behavior persists after code update.

### Root Cause

API Gateway stage not redeployed.

### Fix

Redeploy API Gateway to the stage.

### Interview Explanation

"API Gateway requires redeployment to reflect backend changes."

---

## Scenario 9️⃣CloudFormation Stack Rollback Failed

### Symptom

Stack stuck in `ROLLBACK_FAILED`.

### Root Cause

Resource dependency or manual modification.

### Fix

- Identify failed resource
- Manually delete it
- Retry stack deletion

### Interview Explanation

"I resolve rollback issues by removing blocking resources and retrying deletion."

---

## Scenario 🔟Unexpected AWS Charges

### Symptom

Billing shows unexpected cost.

### Root Cause

High API traffic or retained resources (logs).

### Fix

- Delete CloudWatch log groups
- Enable AWS Budgets

### Interview Explanation

"I monitor costs using AWS Budgets and clean unused resources."

---

## ✅Final Note

These troubleshooting scenarios reflect **real issues encountered during implementation** and demonstrate strong **debugging, operational, and AWS service knowledge**, which are critical in interviews and production environments.