|  |
| --- |
| """Lambda function to persist RFID scan events to DynamoDB.""" |
|  |
| import os |
| from datetime import datetime |
| from typing import Any, Dict |
|  |
| import boto3 |
|  |
| TABLE\_NAME = os.environ.get("TABLE\_NAME", "Inventory") |
| REQUIRED\_FIELDS = [ |
| "item\_id", |
| "expiry\_date", |
| "location", |
| "device\_id", |
| "timestamp", |
| ] |
|  |
|  |
| dynamodb = boto3.resource("dynamodb") |
| table = dynamodb.Table(TABLE\_NAME) |
|  |
|  |
| def lambda\_handler(event: Dict[str, Any], context: Any) -> Dict[str, Any]: |
| """Validate required fields and store the scan event.""" |
| try: |
| for field in REQUIRED\_FIELDS: |
| if field not in event: |
| raise KeyError(f"Missing required field: {field}") |
|  |
| # Validate timestamp format |
| datetime.fromisoformat(event["timestamp"].replace("Z", "+00:00")) |
|  |
| item = {field: event[field] for field in REQUIRED\_FIELDS} |
| table.put\_item(Item=item) |
| return {"statusCode": 200, "body": "Scan recorded"} |
| except (KeyError, ValueError) as e: |
| return {"statusCode": 400, "body": str(e)} |
| except Exception as e: # pragma: no cover - fallback |
| return {"statusCode": 500, "body": str(e)} |