**OIDC compatibility with Cypress and Serverless DynamoDB**

Demo: https://www.youtube.com/watch?v=t18YB3xDfXI

Today, I will be giving a short talk on the Open ID Connect protocol and how we can make Cypress compatible with it and DynamoDB in the AWS world, with additional configuration

This will save you time, if you find yourself using Cypress with AWS DynamoDB, and OIDC in the future

If you have not come across DynamoDB before, Amazon DynamoDB is a fully managed NoSQL (non-relational) database service. Fully managed means AWS handles the infrastructure, including server provisioning, patching, and backups

* OAuth 2.0 is primarily an authorization protocol framework. It enables a third-party application to obtain limited access to a user's resources on another service without sharing the user's credentials
* OAuth 2.0 handles authorization (granting access)

In the context of APIs, particularly when discussing OAUTH2

* Client ID = A public identifier for an application that’s used to tell the API server which application is making the request i.e. the username of the app
* Client Secret = A confidential key that only the application and the API server should know. It’s used to authorize the application, proving that its legitimate, and associated with the Client ID
* OpenID Connect handles authentication (verifying identity) and is frequently used to obtain access tokens
* OIDC is an authentication protocol built on top of the OAuth 2.0 framework which means it utilizes OAuth 2.0

Cypress doesn’t natively support complex authentication flows like OIDC which require token exchanges and validations

OIDC compatibility with Cypress is not inherently a generic feature. Integrating AWS Dynamo Database adds another layer of complexity, as AWS SDK requires proper credential management controlled by Identity Access Management policies. Cypress struggles with replicating the OIDC authentication flow to obtain the OIDC token.

**GitHub Actions YAML Script Elements**

permissions:  
 id-token: write # Grants GH Actions workflow permission to request OIDC JWT token  
 contents: read # Grants the workflow permission to access files, code, and other data in repo

|  |  |
| --- | --- |
| aws-actions/configure-aws-credentials@v4 | Standard script designed to configure AWS credentials for subsequent steps in the workflow |
| role-to-assume: | Grants a set of temporary permissions based on the policies in the role, in this case access to DynamoDB |
| role-session-name: | Used for logging purposes |
| aws sts get-caller-identity | Verifies and confirms the AWS identity being used within the workflow i.e. the arn role |