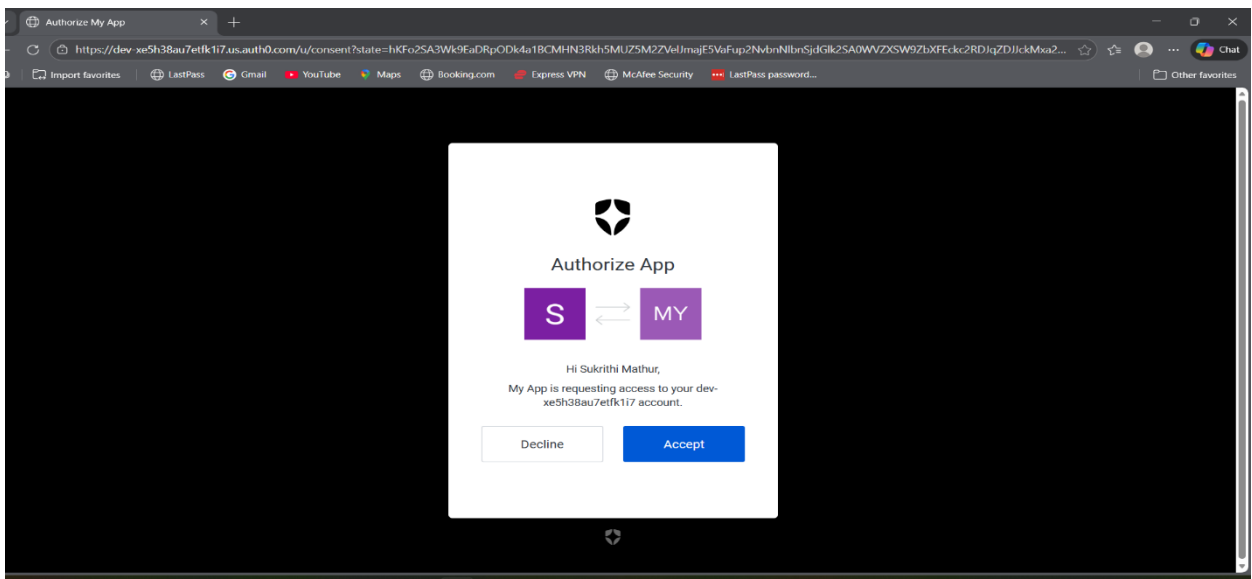
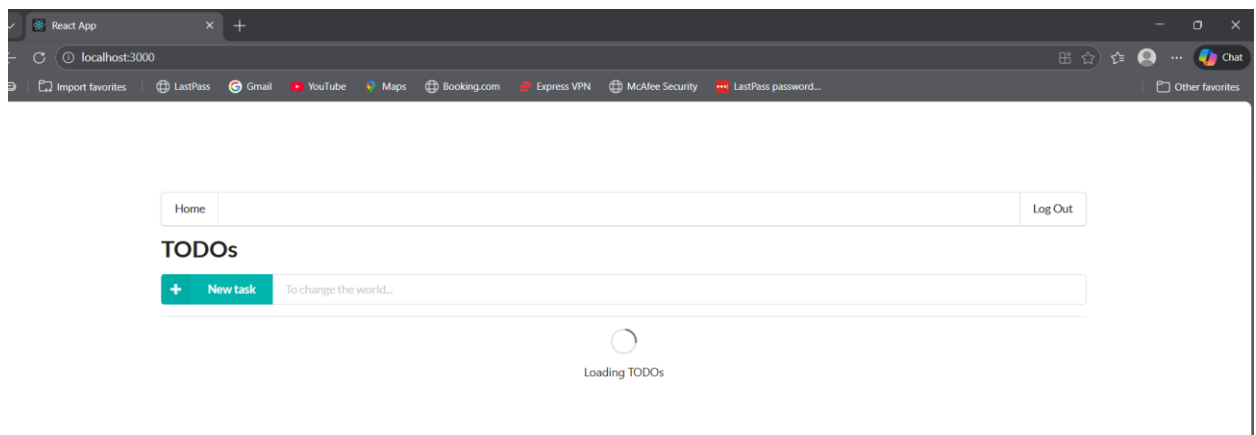


## OUTPUTS SCREENSHOTS of REACT APP



## Screenshots of successfully compiled Backend codes

The screenshot shows the VS Code interface with the Explorer on the left displaying the project structure. The main editor shows the code for `createTodo.js`. The code imports `middy`, `cors`, and other modules, and defines a handler for the `createTodo` endpoint. The terminal at the bottom shows the command `start > backend > src > lambda > http > JS createTodo.js > handler > handler() callback` being executed, followed by the AWS CLI command `aws configure` and the `Remove-Item` command.

The screenshot shows the VS Code interface with the Explorer on the left displaying the project structure. The main editor shows the code for `deleteTodo.js`. The code imports `middy`, `cors`, and other modules, and defines a handler for the `deleteTodo` endpoint. The terminal at the bottom shows the command `start > backend > src > lambda > http > JS deleteTodo.js > handler > handler() callback` being executed, followed by the AWS CLI command `aws configure` and the `Remove-Item` command.

## Frontend (Client side codes )- successfull URL

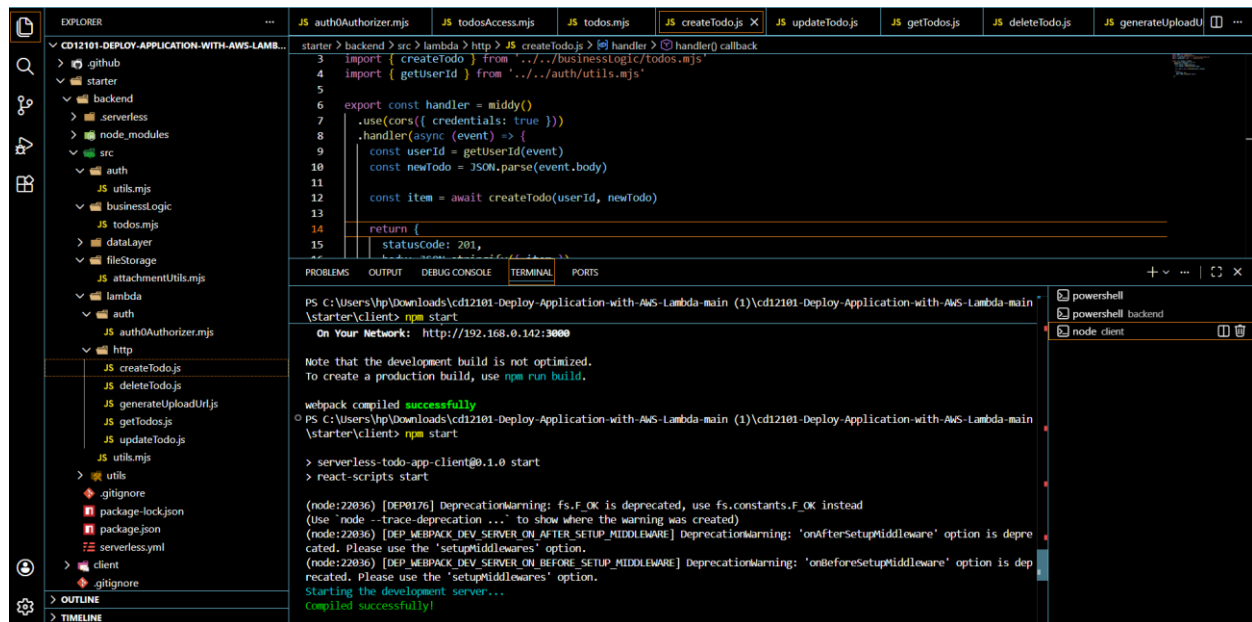
### .env (client side ) URL'S

REACT\_APP\_AUTH0\_DOMAIN=dev-xe5h38au7etfk1i7.us.auth0.com

REACT\_APP\_AUTH0\_CLIENT\_ID=RxKs84QnayvWDUMHJrPxn2ECrdn2mx89

REACT\_APP\_API\_ENDPOINT=https://jm8ghi66x0.execute-api.us-east-1.amazonaws.com/dev

Compiled successfully!



### App (local URL – React App)

You can now view serverless-todo-app-client in the browser

Local: <http://localhost:3000>

On Your Network: <http://192.168.0.142:3000>

webpack compiled successfully

serverless.yml code

service: todo-app

frameworkVersion: '3'

provider:

name: aws

runtime: nodejs18.x

stage: \${opt:stage, 'dev'}

region: us-east-1

environment:

TODOS\_TABLE: Todos-\${self:provider.stage}-\${opt:uniqueId, '1234'}

TODOS\_CREATED\_AT\_INDEX: CreatedAtIndex

ATTACHMENT\_S3\_BUCKET: todo-attachments-\${self:provider.stage}-\${opt:uniqueId, '1234'}

iam:

role:

statements:

- Effect: Allow

Action:

- dynamodb:Query
- dynamodb:PutItem
- dynamodb:UpdateItem
- dynamodb:DeleteItem

Resource:

- arn:aws:dynamodb:\*:\*:table/\${self:provider.environment.TODOS\_TABLE}

-

arn:aws:dynamodb:\*:\*:table/\${self:provider.environment.TODOS\_TABLE}/index/\${self:provider.environment.TODOS\_CREATED\_AT\_INDEX}

- Effect: Allow

Action:

- s3:PutObject

Resource:

- arn:aws:s3:::\${self:provider.environment.ATTACHMENT\_S3\_BUCKET}/\*

functions:

Auth:

handler: src/lambda/auth/auth0Authorizer.handler

GetTodos:

handler: src/lambda/http/getTodos.handler

events:

- http:

  - method: get

  - path: todos

  - cors: true

  - authorizer: Auth

CreateTodo:

handler: src/lambda/http/createTodo.handler

events:

- http:

  - method: post

  - path: todos

  - cors: true

  - authorizer: Auth

UpdateTodo:

handler: src/lambda/http/updateTodo.handler

events:

- http:

  - method: patch

  - path: todos/{todoId}

  - cors: true

  - authorizer: Auth

DeleteTodo:

handler: src/lambda/http/deleteTodo.handler

events:

- http:

method: delete

path: todos/{todold}

cors: true

authorizer: Auth

GenerateUploadUrl:

handler: src/lambda/http/generateUploadUrl.handler

events:

- http:

method: post

path: todos/{todold}/attachment

cors: true

authorizer: Auth

resources:

Resources:

TodosTable:

Type: AWS::DynamoDB::Table

Properties:

AttributeDefinitions:

- AttributeName: userId

AttributeType: S

- AttributeName: todold

AttributeType: S

- AttributeName: createdAt

AttributeType: S

KeySchema:

- AttributeName: userId

KeyType: HASH

- AttributeName: todoId

KeyType: RANGE

BillingMode: PAY\_PER\_REQUEST

TableName: \${self:provider.environment.TODOS\_TABLE}

LocalSecondaryIndexes:

- IndexName: \${self:provider.environment.TODOS\_CREATED\_AT\_INDEX}

KeySchema:

- AttributeName: userId

KeyType: HASH

- AttributeName: createdAt

KeyType: RANGE

Projection:

ProjectionType: ALL

AttachmentsBucket:

Type: AWS::S3::Bucket

Properties:

BucketName: \${self:provider.environment.ATTACHMENT\_S3\_BUCKET}

CorsConfiguration:

CorsRules:

- AllowedOrigins:

\_ '\*'

AllowedHeaders:

\_ '\*'

AllowedMethods:

- PUT

- GET

EXPLORER

CD12101-DEPLOY-APPLICATION...

- .github
- starter
  - backend
    - serverless
    - node\_modules
    - src
      - auth
        - JS utils.mjs
      - businessLogic
        - JS todos.mjs
      - dataLayer
        - JS todosAccess.mjs
      - fileStorage
        - JS attachmentUtils.mjs
      - lambda
        - auth
          - JS authOAuthorizer.mjs
        - http
          - JS createTodos.js
          - JS deleteTodos.js
          - JS generateUploadUrls.js
          - JS getTodos.js
          - JS updateTodos.js
          - JS utils.mjs
      - utils
        - .gitignore
        - package-lock.json
        - package.json
        - serverless.yml
      - client

JS todosAccess.mjs

```
1 import { decode } from 'jsonwebtoken'
2 import { createLogger } from '../utils/logger.mjs'
3
4 const logger = createLogger('utils')
5
6 /**
7  * Parse a JWT token and return a user id
8  * @param jwtToken JWT token to parse
9  * @returns a user id from the JWT token
10 */
11 export function parseUserId(jwtToken) {
12   const decodedJwt = decode(jwtToken)
13   return decodedJwt.sub
14 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\Downloads\cd12101-Deploy-Application-with-AWS-Lambda-main (1)\cd12101-Deploy-Application-with-AWS-Lambda-main> \starter\client> npm start

recated. Please use the 'setupMiddlewares' option.  
Starting the development server...  
Compiled successfully!

You can now view **serverless-todo-app-client** in the browser.

Compiled successfully!

You can now view **serverless-todo-app-client** in the browser.

Local: http://localhost:3000  
On Your Network: http://192.168.0.142:3000

EXPLORER

CD12101-DEPLOY-APPLICATION...

- .github
- starter
  - backend
    - serverless
    - node\_modules
    - src
      - auth
        - JS utils.mjs
      - businessLogic
        - JS todos.mjs
      - dataLayer
        - JS todosAccess.mjs
      - fileStorage
        - JS attachmentUtils.mjs
      - lambda
        - auth
          - JS authOAuthorizer.mjs
        - http
          - JS createTodos.js
          - JS deleteTodos.js
          - JS generateUploadUrls.js
          - JS getTodos.js
          - JS updateTodos.js
          - JS utils.mjs
      - utils
        - .gitignore
        - package-lock.json
        - package.json
        - serverless.yml
      - client
        - .gitignore

JS todosAccess.mjs

```
1 import AWS from 'aws-sdk'
2
3 const docClient = new AWS.DynamoDB.DocumentClient()
4 const TODOS_TABLE = process.env.TODOS_TABLE
5 const INDEX = process.env.TODOS_CREATED_AT_INDEX
6
7 export class TodosAccess {
8   async getTodos(userId) {
9     const result = await docClient.query({
10       TableName: TODOS_TABLE,
11       IndexName: INDEX,
12       KeyConditionExpression: 'userId = :uid',
13       ExpressionAttributeValues: {
14         ':uid': userId
15       }
16     }).promise()
17
18     return result.Items
19   }
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\Downloads\cd12101-Deploy-Application-with-AWS-Lambda-main (1)\cd12101-Deploy-Application-with-AWS-Lambda-main> \starter\client> npm start

recated. Please use the 'setupMiddlewares' option.  
Starting the development server...  
Compiled successfully!

You can now view **serverless-todo-app-client** in the browser.

Compiled successfully!

You can now view **serverless-todo-app-client** in the browser.

Local: http://localhost:3000  
On Your Network: http://192.168.0.142:3000

EXPLORER

CD12101-DEPLOY-APPLICATION...

- .github
- starter
  - backend
    - serverless
    - node\_modules
    - src
      - auth
        - JS utils.mjs
      - businessLogic
        - JS todos.mjs
      - dataLayer
        - JS todosAccess.mjs
      - fileStorage
        - JS attachmentUtils.mjs
      - lambda
        - auth
          - JS authOAuthorizer.mjs
        - http
          - JS createTodos.js
          - JS deleteTodos.js
          - JS generateUploadUrls.js
          - JS getTodos.js
          - JS updateTodos.js
          - JS utils.mjs
      - utils
        - .gitignore

JS createTodos.js

```
1 import middy from '@middy/core'
2 import cors from '@middy/http-cors'
3 import { createTodo } from '../businessLogic/todos.mjs'
4 import { getUserId } from '../auth/utils.mjs'
5
6 export const handler = middy()
7   .use(cors({ credentials: true }))
8   .handler(async (event) => {
9     const userId = getUserId(event)
10     const newTodo = JSON.parse(event.body)
11
12     const item = await createTodo(userId, newTodo)
13
14     return {
15       statusCode: 201,
16       body: JSON.stringify({ item })
17     }
18   })
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\Downloads\cd12101-Deploy-Application-with-AWS-Lambda-main (1)\cd12101-Deploy-Application-with-AWS-Lambda-main> \starter\client> npm start

recated. Please use the 'setupMiddlewares' option.  
Starting the development server...  
Compiled successfully!

dev-xeSh38au7etfk1i7

DEVELOPMENT

SearchDiscuss your needsDocumentation

Getting Started

AI AgentsNEW

Activity

Applications

Applications

APIs

SSO Integrations

Authentication

Organizations

User Management

Branding

Security

Actions

Event StreamsEARLY

Monitoring

Marketplace

Extensions

Settings

QuickstartSettingsAPIsAddonsConnectionsLogin ExperienceOkta Integration NetworkNEW

Basic Information

Name \*

My App

Domain

dev-xeSh38au7etfk1i7.us.auth0.com

Client ID

RxKs84QnayvWDUMHJrPxn2ECrdn2mx89

Client Secret

.....

The Client Secret is not base64 encoded.

Description

Add a description in less than 140 characters

A free text description of the application. Max character count is 140.

Get support

EXPLORER

CD12101-DEPLOY-APPLICATION-WITH-AWS-LAMBDA

github

starter

backend

serverless

node\_modules

src

auth

businessLogic

todos.mjs

dataLayer

todosAccess.mjs

fileStorage

attachmentUtils.mjs

lambda

auth

auth0Authorizer.mjs

http

createTodo.js

deleteTodo.js

generateUploadUrl.js

getTodos.js

updateTodo.js

utils.mjs

utils

gitignore

package-lock.json

package.json

serverless.yml

client

gitignore

OUTLINE

TIMELINE

start

backend

src

lambda

http

updateTodo.js

handler

1

import middy from '@middy/core'

2

import cors from '@middy/http-cors'

3

import { updateTodo } from '../businessLogic/todos.mjs'

4

import { getUserId } from '../auth/utils.mjs'

5

6

export const handler = middy()

7

.use(cors({ credentials: true })))

8

.handler(async (event) => {

9

const userId = getUserId(event)

10

const todoId = event.pathParameters.todoId

11

const updatedTodo = JSON.parse(event.body)

12

13

await updateTodo(userId, todoId, updatedTodo)

14

15

return {

16

statusCode: 200,

17

body: ''

18

}

19

})

20

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS c:\Users\hp\Downloads\cd12101-Deploy-Application-with-AWS-Lambda-main (1)\cd12101-Deploy-Application-with-AWS-Lambda-main

starter\client> npm start

Local: http://localhost:3000

You can now view serverless-todo-app-client in the browser.

You can now view serverless-todo-app-client in the browser.

Local: http://localhost:3000

On Your Network: http://192.168.0.142:3000

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

powershell

powershell backend

node client

Created of codes (folder as well as files )

