A

<https://aws.amazon.com/ko/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>

B

<https://aws.amazon.com/ko/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/>

변경할 리소스

* join.html
* cognito-auth.js

서버리스 웹 -A실습 기본 웹 -B실습

1. B 1단계 버킷생성(권한 모두 체크x), 파일 업로드 -> A 1단계 정책추가
2. A 2단계 사용자관리
3. B2단계 함수- lambda.txt 추가 - 테스트 안해도됨
4. B 3단계 배포까지
5. A 4단계 2,5만, invoke url은 직전에 만든 API에서 복사한 것 사용
6. B4단계 테이블명 ProjectFunction , 기본키이름 ProjectFunctinEmail

* 생성 후 테이블항목에서 항목만들기 - string 으로 Password, Name 항목 추가
* 값도 아무거나 추가해야 생성됨
* IAM정책 추가, 람다 수정x
* 테스트

1. B5 join.html 에서 API url로 변경 후 업데이트

* 클라이언트 테스트

-> 회원가입시 이메일,패스워드,이름 입력하고 버튼 누르면 db에 저장 & cognito 사용자 풀에 저장

새 함수 연결 B 3-2번째칸

API url 해당 페이지 fetch로

index.html - method, API url 변경

<https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/GettingStarted.Python.html>

람다 - 파이썬 설명서

람다 인라인 정책 추가

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1428510662000",

"Effect": "Allow",

"Action": [

   "DynamoDB:PutItem",

"DynamoDB:GetItem",

"DynamoDB:Scan",

"DynamoDB:Query"

],

"Resource": [

"arn:aws:dynamodb:us-east-1:270161363667:table/ProjectFunction"

]

}

]

}

{

"Sid": "Stmt1428510662000",

"Effect": "Allow",

"Action": [

"lambda:InvokeFunction",

"lambda:GetFunctionConfiguration"

],

"Resource": "*FunctionARN*"

}

파일이름 +( 밑 코드에서 def lambda\_handler(event, context): )함수이름 = 핸들러 이름

= lambda\_function.lambda\_handler

해당 행 전체 반환

from \_\_future\_\_ import print\_function

import boto3

import json

from boto3.dynamodb.conditions import Key, Attr

def lambda\_handler(event, context):

dynamo = boto3.resource('dynamodb').Table('ProjectFunction')

email = event["email"]

response = dynamo.query(

KeyConditionExpression=Key('ProjectFunctionEmail').eq(email)

)

items = response['Items']

return items

이벤트테스트

{

"email": "test2@naver.com"

}

-> 테스트 안의 변수 email이 코드의 event[‘email’]

테스트 이메일도 실제 db에 있는 값이어야 불러올 수 있음(테스트 결과 확인 가능)

여행정보로 바꾼다면

여행지변수 = event[‘변수’]

(쿼리 속 코드) =Key(‘여행지’).eq(여행지변수)

>해당 이메일 사용자의 이름만 반환

from \_\_future\_\_ import print\_function

import boto3

import json

from boto3.dynamodb.conditions import Key, Attr

def lambda\_handler(event, context):

dynamo = boto3.resource('dynamodb').Table('ProjectFunction')

email = event["email"]

response = dynamo.query(

KeyConditionExpression=Key('ProjectFunctionEmail').eq(email)

)

lst = []

for i in response['Items']:

lst.append(i['Name'])

#3개중 1개

return lst -> 값 그대로

return {

'statusCode': 200,

'headers': {'Access-Control-Allow-Origin': '\*'},

'body': json.dumps(lst)

}

return json.dumps(lst)

이벤트테스트

{

"email": "test2@naver.com"

}

from \_\_future\_\_ import print\_function

해당 행 전체 2

import boto3

import json

from boto3.dynamodb.conditions import Key, Attr

def lambda\_handler(event, context):

dynamo = boto3.resource('dynamodb').Table('ProjectFunction')

response = dynamo.query(

KeyConditionExpression=Key('ProjectFunctionEmail').eq('ccccc@naver.com')

)

lst = []

for i in response['Items']:

lst.append(i['Name'])

lst.append(i['ProjectFunctionEmail'])

lst.append(i['Password'])

print('\n')

return lst

<https://airpage.org/xe/language_data/25191>

function sendData() {

var url = "<https://amazon-lambda-api-address.ap-northeast-1.amazonaws.com/prod/demo/someset>";

var mycallback = function(data) {

if (data['result'] == "success") alert("success");

};

realSend(url, {'param1' : '1234', 'param2' : '' + '5678'}, mycallback);

}

function realSend(url, param, callback) {

$.ajax({

type:"POST",

dataType:"JSON",

url : url,

data : JSON.stringify(param),

headers : {"x-api-key" : "[API KEY]"},

crossDomain: true,

contentType: "application/json",

cache: false,

success : function(data) {

callback(data);

},

complete : function(data) {

},

error : function(xhr, status, error) {

}

});

}

# **https 사이트에서 jQuery .ajax() 메서드를 통해 get 또는 post로 비동기통신 할 때 mixed content 오류 발생시 대처법**

<https://devonaws.com/front-end/jquery/https-%EC%82%AC%EC%9D%B4%ED%8A%B8%EC%97%90%EC%84%9C-jquery-ajax-%EB%A9%94%EC%84%9C%EB%93%9C%EB%A5%BC-%ED%86%B5%ED%95%B4-get-%EB%98%90%EB%8A%94-post%EB%A1%9C-%EB%B9%84%EB%8F%99%EA%B8%B0%ED%86%B5/>

> db내용 출력하기

lambda -> html

<https://www.youtube.com/watch?v=wSRP4e0y5tQ>

4번 영상 입력 -> 검색

{'Access-Control-Allow-Origin': '\*'}

<https://forum.jquery.com/topic/script5009-validation-is-undefined>

API 설정

<https://docs.aws.amazon.com/ko_kr/apigateway/latest/developerguide/simple-calc-lambda-api.html>

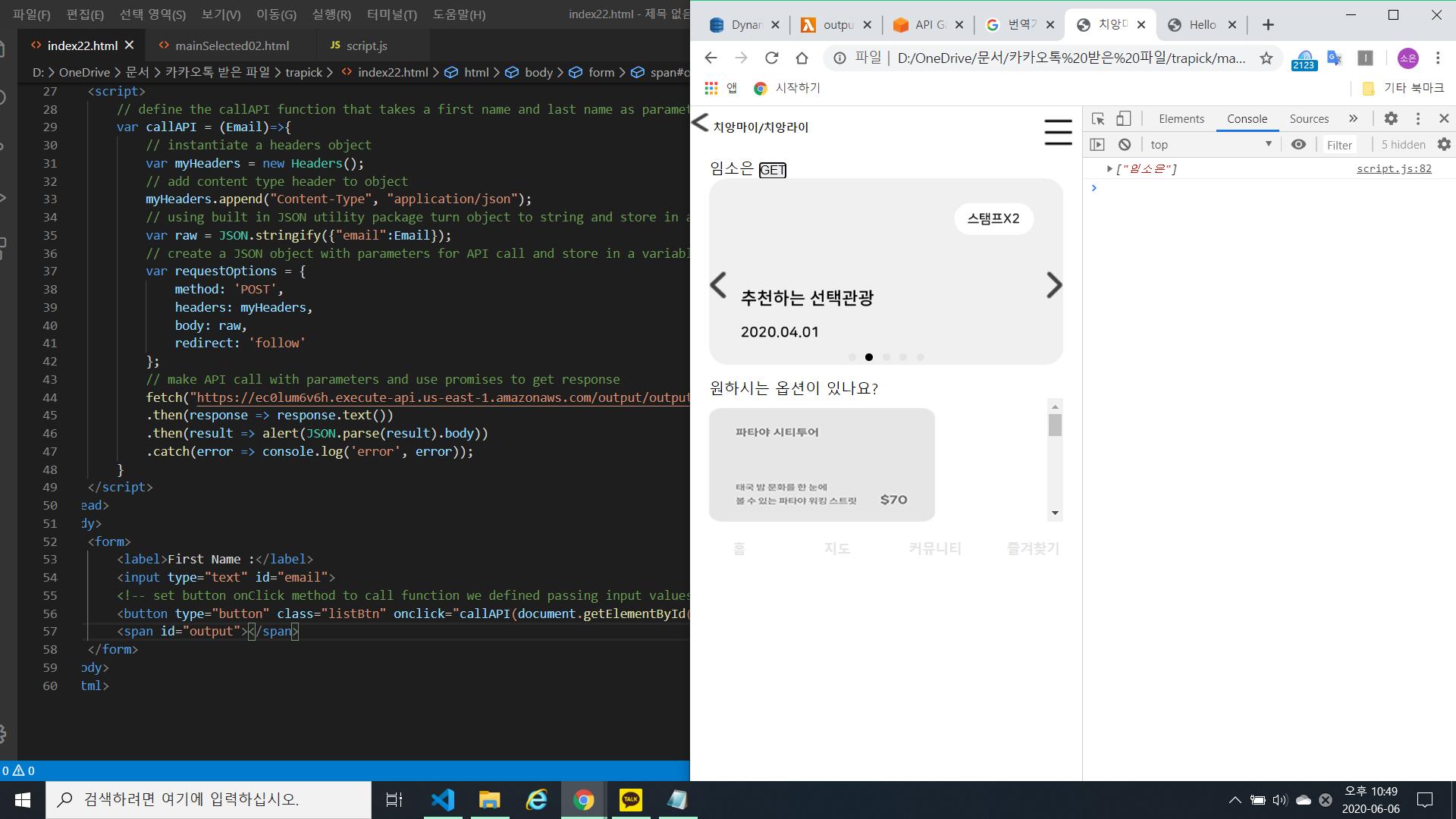
<https://docs.aws.amazon.com/ko_kr/apigateway/latest/developerguide/integrating-api-with-aws-services-lambda.html>

메서드 테스트 예시

#### 쿼리 문자열

**{output}**

**email=test2@naver.com**



>

window.location.href = 'mainSelected02.html';

<a href="#" class="listBtn">GET</a>

db리스트 출력

<https://blog.opid.kr/431>

<https://blog.naver.com/cosmosjs/221715332150>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>동남아 지역선택</title>

<link rel="stylesheet" href="css/base.css"/>

<link rel="stylesheet" href="css/layout.css"/>

<link rel="stylesheet" href="css/main.css"/>

<link rel="stylesheet" href="css/sub.css"/>

<script src="https://code.jquery.com/jquery-1.11.3.js"></script>

<script src="js/slick.js"></script>

<script src="js/script.js"></script>

<script>

function setChildValue(index) {

window.location.href = "mainSelected02.html?index=" + index;

}

</script>

</head>

<body>

<div class="skip-navigation">

<p><a href="#container">본문 바로가기</a></p>

</div>

<div class="wrap">

<div class="header">

<h3 class="MSTitle"><a href="index.html"><img src="img/sub/prev.png" alt="홈으로"/>동남아</a></h3>

<form action="#">

<fieldset>

<legend>통합검색</legend>

<input type="text" title="검색단어입력" placeholder="검색단어입력" class="SearchWordInput"/>

<input type="image" src="img/layout/search.png" alt="검색" class="Search"/>

</fieldset>

</form>

<div class="util">

<a href="#" class="allMenuBtn"><img src="img/layout/allMenuBtn.png" alt="로그인페이지"/></a>

<div class="menuArea">

<p class="myPage">

<a href="#" class="closeBtn">

<img src="img/layout/closeBtn.png" alt="로그인닫기"/>

</a>

로그인

</p>

<div class="logLogo">

<!-- <img src="#" class="logLogo" alt="trapick로고"/>-->

</div>

<form action="">

<fieldset>

<legend>로그인</legend>

<table class="login">

<caption></caption>

<tbody>

<tr>

<th scope="row"><label for="id">아이디</label></th>

<td><input type="text" id="id" placeholder="아이디" title="아이디입력"/></td>

</tr>

<tr>

<th scope="row"><label for="pw">비밀번호</label></th>

<td><input type="password" id="pw" placeholder="비밀번호" title="비밀번호입력"/></td>

</tr>

</tbody>

</table>

<p>

<button class="logBtn">로그인</button>

<a href="#" class="lookBox">아이디/비밀번호찾기</a>

<a href="join.html" class="member">회원가입</a>

</p>

</fieldset>

</form>

</div>

</div>

</div>

<div class="container" id="container">

<div class="sectionMS01">

<input type="button" value="치앙마이/치앙라이" class="listBtn" onclick="setChildValue(document.querySelector('#list'))"><span class="city"></span><span class="cityNumber">32</span></input>

<a href="#"><span class="city">방콕/파타야</span><span class="cityNumber">32</span></a>

<a href="#"><span class="city">싱가포르/브루나이</span><span class="cityNumber">32</span></a>

<a href="#"><span class="city">다낭/호이안.후에</span><span class="cityNumber">32</span></a>

<a href="#"><span class="city">하노이/하롱베이</span><span class="cityNumber">32</span></a>

<a href="#"><span class="city">보라카이</span><span class="cityNumber">32</span></a>

<a href="#"><span class="city">그 외 관광지</span><span class="cityNumber">32</span></a>

</div>

</div>

<ul class="mainGnb">

<li><a href="index.html">홈</a></li>

<li><a href="sub01.html">지도</a></li>

<li><a href="sub02.html">커뮤니티</a></li>

<li><a href="sub03.html">즐겨찾기</a></li>

</ul>

</div>

</body>

</html>

$("#list").html(data);

<a href="mainSelected02.html" value="GET" class="listBtn city">치앙마이/치앙라이</a>

onclick="

fetch('css.html').then(function (response) {

return response.text();

}).then(function (text) {

console.log(text);

<Lambda : function-tourlist>

from \_\_future\_\_ import print\_function

import json

import os

import boto3

import decimal

from boto3.dynamodb.conditions import Key, Attr

#Helper class toconvert a DynamoDB item to JSON

class DecimalEncoder(json.JSONEncoder):

def default(self, o):

if isinstance(o, deciaml.Deciaml):

if o % 1 > 0:

return float(o)

else:

return int(o)

return super(DecimalEncoder, self).default(o)

def lambda\_handler(event, context):

dynamodb=boto3.resource('dynamodb')

table=dynamodb.Table("TourDetail")

print("TourList in Country")

response=table.query(

KeyConditionExpression=Key('Country').eq('Thailand')

)

lst=[]

for i in response['Items']:

lst.append(i['TourName'])

lst.append(i['Country'])

lst.append(i['Area'])

lst.append(i['Price'])

return lst

<test event>

{

"TourName": "No.1",

"Country": "Thailand",

"Area": "Bangkok",

"Price": "$1,000"

}

입력값 출력

<https://medium.com/@mystar09070907/%EC%9B%B9-%ED%8E%98%EC%9D%B4%EC%A7%80-client-%EC%97%90%EC%84%9C-%EC%A0%95%EB%B3%B4-%EB%B3%B4%EB%82%B4%EA%B8%B0-bf3aff952d3d>