

# 리워드 관리자 고객출석현황조회, 고객리워드수령조회

👤 생성자	태환 김
🕒 생성 일시	@March 17, 2023 7:53 PM
🏷 태그	
👤 최종 편집자	태환 김
🕒 최종 편집 일시	@March 27, 2023 12:14 PM

DynamoDB > 항목 > Attendance

테이블 (4) ×

모든 테이블 태그

테이블 이름으로 테이블 찾기

< 1 > ⚙

- ☒ Attendance
- ☐ Products
- ☐ Rewards
- ☐ Users

## Attendance

☒ 자동 미리 보기 테이블 세부 정보 보기

▼ 항목 스캔 또는 쿼리

☒ 스캔
☐ 쿼리

테이블 또는 인덱스 선택  
테이블 - Attendance
속성 프로젝션 선택  
모든 속성

▶ 필터

실행 재설정

✓ 완료. 사용된 읽기 용량 단위: 0.5
×

반환된 항목 (4) ↺ 작업 ▼ 항목 생성


< 1 > ⚙

<input type="checkbox"/>	id	date	user_id
<input type="checkbox"/>	4	5	4
<input type="checkbox"/>	2	3	3
<input type="checkbox"/>	3	5	2
<input type="checkbox"/>	1	3	1

- 고객의 출석 현황을 알아야 합니다. 을 구현해야합니다.


DynamoDB 테이블 중 Attendance 의 user\_id를 받아온다.

## ▼ 함수 개요 정보

 attendance\_status

 Layers

(0)

 API 게이트웨이

+

+ 트리거 추가

코드

테스트

모니터링

구성

별칭

버전

## 코드 소스 정보


File Edit Find View Go Tools Window

Test

Deploy

Go to Anything (Ctrl-P)

Environment

attendance\_status -   
index.js

```
1 const AWS = require('aws-sdk');
2 const dynamoDB = new AWS.DynamoDB.DocumentClient();
3
4 async function getAttendanceData(user_id) {
5   const params = {
6     TableName: 'Attendance',
7     FilterExpression: 'user_id = :user_id',
8     ExpressionAttributeValues: {
9       ':user_id': user_id,
10    },
11  };
12
13  try {
14    const result = await dynamoDB.scan(params).promise();
15    return result.Items;
16  } catch (error) {
17    console.error(error);
18    throw error;
19  }
20 }
21
22 exports.handler = async (event) => {
23   if (!event.pathParameters || !event.pathParameters.user_id) {
24     return {
25       statusCode: 400,
26       body: JSON.stringify({ message: 'user_id를 찾을 수 없습니다.' }),
27     };
28   }
29
30   const user_id = event.pathParameters.user_id;
31
32   try {
33     const attendanceData = await getAttendanceData(user_id);
34     return {
35       statusCode: 200,
36       body: JSON.stringify(attendanceData),
37     };
38   } catch (error) {
39     return {
40       statusCode: 500,
41       body: JSON.stringify({ message: '출석 데이터를 가져오는 동안 오류가 발생했습니다.' }),
42     };
43   }
44 }
```

## 트리거 (1) 정보

🔍 트리거 찾기

☐ 트리거



**API 게이트웨이:** admin-api

arn:aws:execute-api:ap-northeast-2:180693256225:ay5gusxtn6/\*/\*/Attendance/{user\_id}

☐

API 엔드포인트: [https://ay5gusxtn6.execute-api.ap-northeast-2.amazonaws.com/Attendance/{user\\_id}](https://ay5gusxtn6.execute-api.ap-northeast-2.amazonaws.com/Attendance/{user_id})

▶ 세부 정보

출석 조회를 위해서 Lambda 를 만들고 코드를 작성한 뒤 API gateway에서 HTTP API를 생성 후 API gateway를 Lambda에 통신 연결합니다.

## 리소스 요약

역할 문서 보기



Amazon DynamoDB  
1 actions, 1 resources

함수에 액세스 권한이 있는 리소스 및 작업을 보려면 서비스를 선택합니다.

작업별

리소스별

리소스

작업

모든 리소스

Allow: dynamodb:\*

① 다음 정책 문은 이 정보를 Lambda 연습니다.

- 관리형 정책 AmazonDynamoDBFullAccess, 문의 0

DB 안에 데이터를 받아오기 위해서 권한(role)을 추가해줍니다.

https://ay5gusxtn6.execute-api.ap-northeast-2.amazonaws.com/Attendance/3

GET https://ay5gusxtn6.execute-api.ap-northeast-2.amazonaws.com/Attendance/3

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 1252 ms Size: 209 B Save Response


Pretty Raw Preview Visualize Text


```
1 [{"date":3,"user_id":"3","id":2}]
```


포스트맨으로 위에 api 엔드포인트로 user\_id를 끝점에 요청하면 DB안의 attendace 데이터 응답을 받기까지 완성.


- 어떤 고객이 어떤 리워드를 받아가는지 알아야 합니다.


▼ 함수 개요 정보

 reward\_status

 Layers (0)

 API 게이트웨이

 트리거 추가

 대상 추가

코드 | 테스트 | 모니터링 | 구성 | 별칭 | 버전

코드 소스 정보

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

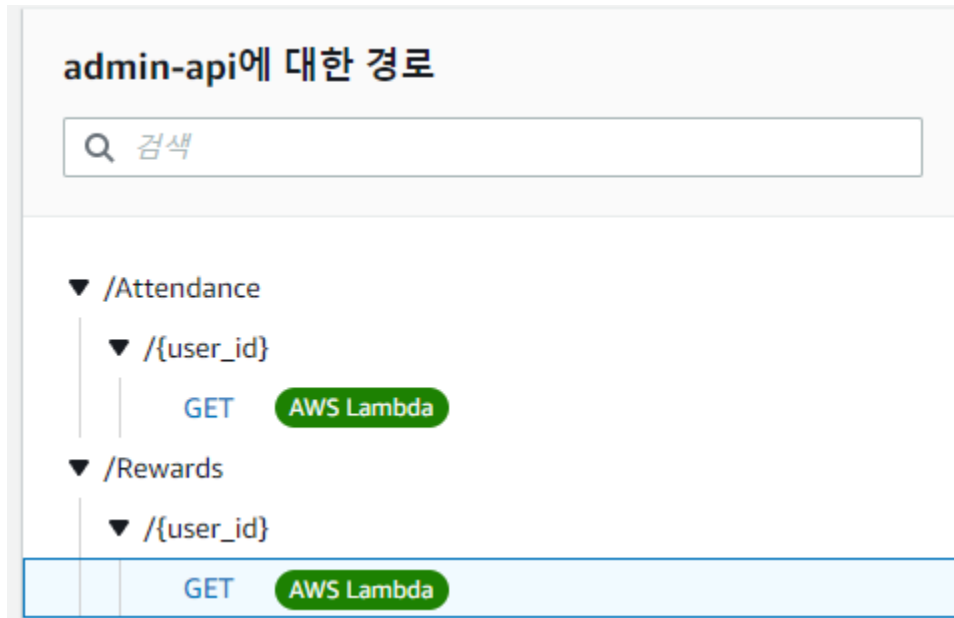
- reward\_status - /
  - index.js

```

1  const AWS = require('aws-sdk');
2  const dynamoDB = new AWS.DynamoDB.DocumentClient();
3
4  async function getRewardsData(user_id) {
5    const params = {
6      TableName: 'Rewards',
7      FilterExpression: 'user_id = :user_id',
8      ExpressionAttributeValues: {
9        ':user_id': user_id,
10     },
11   };
12
13   try {
14     const result = await dynamoDB.scan(params).promise();
15     return result.Items;
16   } catch (error) {
17     console.error(error);
18     throw error;
19   }
20 }
21
22 exports.handler = async (event) => {
23   if (!event.pathParameters || !event.pathParameters.user_id) {
24     return {
25       statusCode: 400,
26       body: JSON.stringify({ message: 'user_id를 찾을 수 없습니다.' }),
27     };
28   }
29
30   const user_id = event.pathParameters.user_id;
31
32   try {
33     const RewardsData = await getRewardsData(user_id);
34     return {
35       statusCode: 200,
36       body: JSON.stringify(RewardsData),
37     };
38   } catch (error) {
39     return {
40       statusCode: 500,
41       body: JSON.stringify({ message: '리워드 데이터를 가져오는 동안 오류가 발생했습니다.' }),
42     };
43   }
44 }

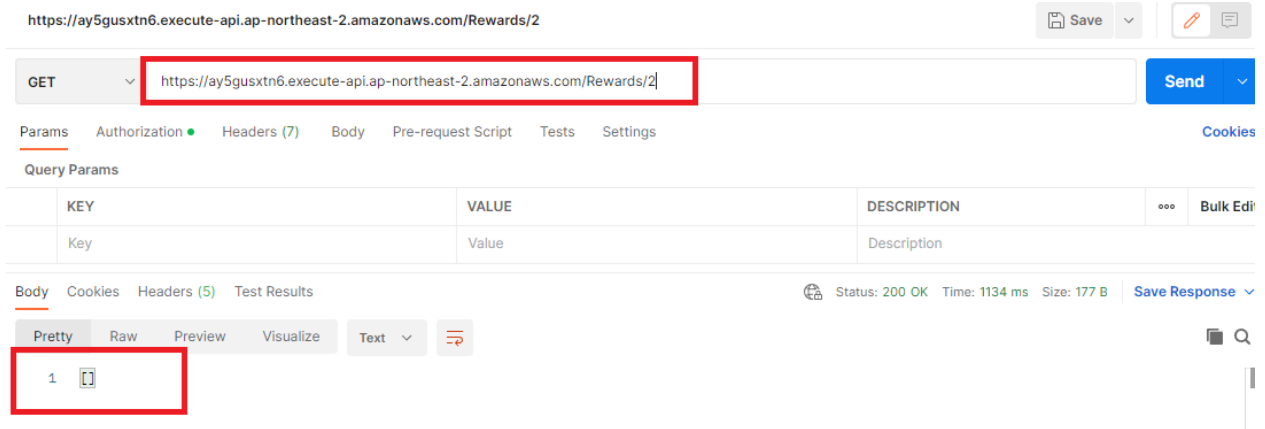
```

위에 attendace 처럼 같은 GET 요청이기때문에 코드자체는 크게 달라질 필요가 없습니다.



이 전에 attendace 출석 조회를 구현하기 위해 생성한 HTTP API를 그대로 쓰면 되고, 엔드포인트만 새로 생성해서 reward 람다를 통신 연결했습니다.

하지만,

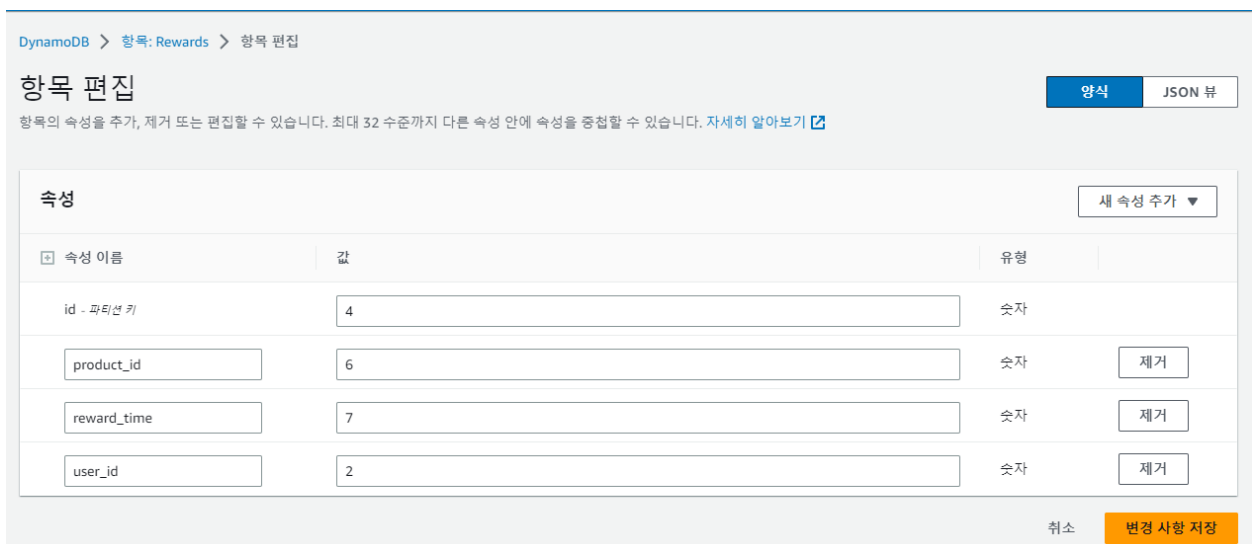


위와 같이 포스트맨으로 엔드포인트 끝점에 값을 넣고 확인을 해봤지만 요청은 200 ok가 떴지만 값이 나오지않았습니다.

에러 로그는 user\_id를 찾을 수 없다고 뜨는데, 이 문제를 고치기위해서 매개변수 매핑을 해야 하나, 통신이 잘못되었나, 권한 문제인가, DB 트리를 잡아야하는가, DB 데이터와 값이 오타난 부분이 있나, 함수 코드에 오타가 있는가 다 살펴보았지만 문제는 없어서 고난을 겪고 있었는데

결론적으로 말하면 정말 간단한 슈팅트러블이었습니다.

문제가 뭐였냐면,



rewards 테이블의 user\_id 속성의 유형을 숫자로 처리했기때문에, 문자열로만 받을 수 있게 만든 람다 함수 코드가 user\_id (숫자)를 인식하지 못한 거 였습니다..



'user\_id'가 숫자인 경우 이전 예제와 유사한 방식으로 DynamoDB 테이블을 쿼리할 수 있습니다. 정수인지 부동 소수점 숫자인지에 따라 'user\_id' 문자열을 'parseInt()' 또는 'parseFloat()'를 사용하여 숫자로 변환해야 합니다. 다음은 'parseInt()'를 사용하는 예입니다.

javascript

Copy code

```
const AWS = require('aws-sdk');
const dynamoDB = new AWS.DynamoDB.DocumentClient();

async function getUserData(user_id) {
  const params = {
    TableName: 'YourTableName',
    FilterExpression: 'user_id = :user_id',
    ExpressionAttributeValues: {
      ':user_id': user_id,
    },
  };
};

try {
  const result = await dynamoDB.scan(params).promise();
  return result.Items;
} catch (error) {
  console.error(error);
  throw error;
}

exports.handler = async (event) => {
  if (!event.pathParameters || !event.pathParameters.user_id) {
    return {
      statusCode: 400,
      body: JSON.stringify({ message: 'User ID not provided in the path.' }),
    };
  }

  const user_id = parseInt(event.pathParameters.user_id, 10);

  try {
    const userData = await getUserData(user_id);
    return {
      statusCode: 200,
      body: JSON.stringify(userData),
    };
  } catch (error) {
    return {
      statusCode: 500,
      body: JSON.stringify({ message: 'Error fetching user data.' }),
    };
  }
};
```

이 예제에서는 먼저 'event' 객체에 'pathParameters' 객체가 있는지, 그리고 'user\_id' 속성이 있는지 확인합니다. 그런 다음 'user\_id' 문자열을 'parseInt()'를 사용하여 정수로 변환합니다. 그런 다음 정수 값으로 DynamoDB 테이블을 쿼리합니다.

'YourTableName'을 DynamoDB 테이블의 실제 이름으로 바꾸는 것을 잊지 마십시오.

검색을 해본 결과, parseInt()를 사용해서 숫자를 받아오는 방식을 쓰면 된다는 걸 배웠습니다. 하지만, user\_id은 문자열로 쓰는 게 마땅하다고 생각했기때문에 DB 테이블 안에 있는 user\_id를 숫자에서 문자열로 변경하고 다시 시도해봤습니다.

DynamoDB > 항목: Rewards > 항목 편집

### 항목 편집

항목의 속성을 추가, 제거 또는 편집할 수 있습니다. 최대 32 수준까지 다른 속성 안에 속성을 중첩할 수 있습니다. 자세히 알아보기

양식
JSON 뷰

속성 이름	값	유형	
id - 파티션 키	3	숫자	
product_id	1	숫자	제거
reward_time	3	숫자	제거
user_id	1	문자열	제거

취소 변경 사항 저장

https://c068m9qap5.execute-api-northeast-2.amazonaws.com/Rewards/1

GET https://c068m9qap5.execute-api-northeast-2.amazonaws.com/Rewards/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 118 ms Size: 231 B Save Response

Pretty Raw Preview Visualize Text

```
1 [{"user_id": "1", "id": 3, "reward_time": 3, "product_id": 1}]
```

다행히 값을 응답 받을 수 있었습니다.

이 하나의 트러블 슈팅으로 인해 몇 시간을 허비했다고도 생각했지만, 이런 에러도 존재할 수 있다는 하나의 견문을 넓히는 데 도움이 됐다고 생각합니다.