

API gateway 실습

AWS REST API gateway 실습

목차

- AWS REST API 생성
- DynamoDB에 연결하여 API 전송 테스트

AWS REST API 생성

AWS API gateway 종류

- Private REST API는 퍼블릭 인터넷에서 접근 못함
- 우리 프로젝트에는 public REST API를 사용

API 유형 선택

HTTP API

OIDC 및 OAuth2와 같은 기능과 기본 CORS 지원이 내장된, 지연 시간이 짧고 비용 효율적인 REST API를 구축합니다.

다음과 호환됩니다.
Lambda, HTTP 백엔드

[가져오기](#)[구축](#)

WebSocket API

채팅 애플리케이션 또는 대시보드와 같은 실시간 사용 사례를 위해 지속적 연결을 사용하여 WebSocket API를 구축합니다.

다음과 호환됩니다.
Lambda, HTTP, AWS 서비스

[구축](#)

REST API

API 관리 기능과 함께 요청 및 응답을 완벽하게 제어할 수 있는 REST API를 개발합니다.

다음과 호환됩니다.
Lambda, HTTP, AWS 서비스

[가져오기](#)[구축](#)

REST API 프라이빗

VPC 내에서만 액세스할 수 있는 REST API를 생성합니다.

다음과 호환됩니다.
Lambda, HTTP, AWS 서비스

[가져오기](#)[구축](#)

API 생성 방법

- API 생성할 때 예제 코드를 넣어서 생성할 수도 있다.
- 하지만 new API를 사용 -> 커스텀 설정할 수 있도록 생성함.

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API ☐ Import from Swagger or Open API 3 ☐ Example API

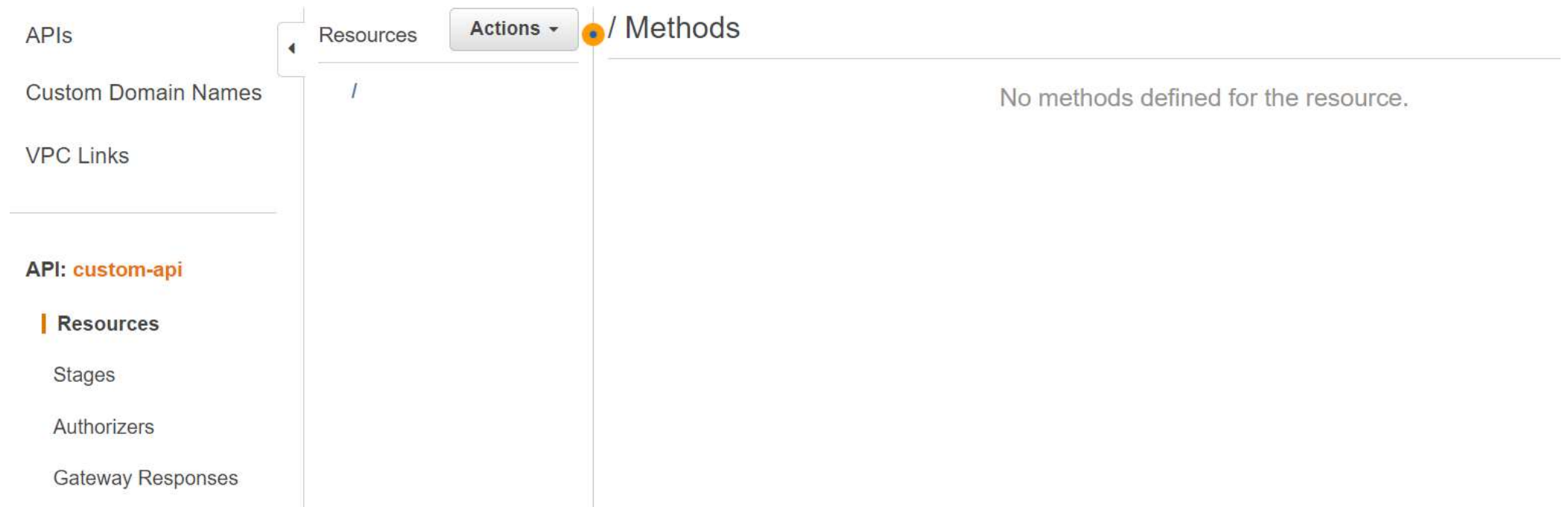
API 생성 방법 - setting

- Regional: 생성하는 지역에 국한됨.
- Edge optimized: 고객이 위치한 곳에서 가장 가까운 edge location 을 찾아서 요청해준다.
- Private: VPC 안에 구축할 때 사용하는 것

API name*	<input type="text" value="custom-api"/>
Description	<input type="text"/>
Endpoint Type	<div><div>Regional ▼ ⓘ</div><div><div>Regional</div><div>Edge optimized</div><div>Private</div></div></div>

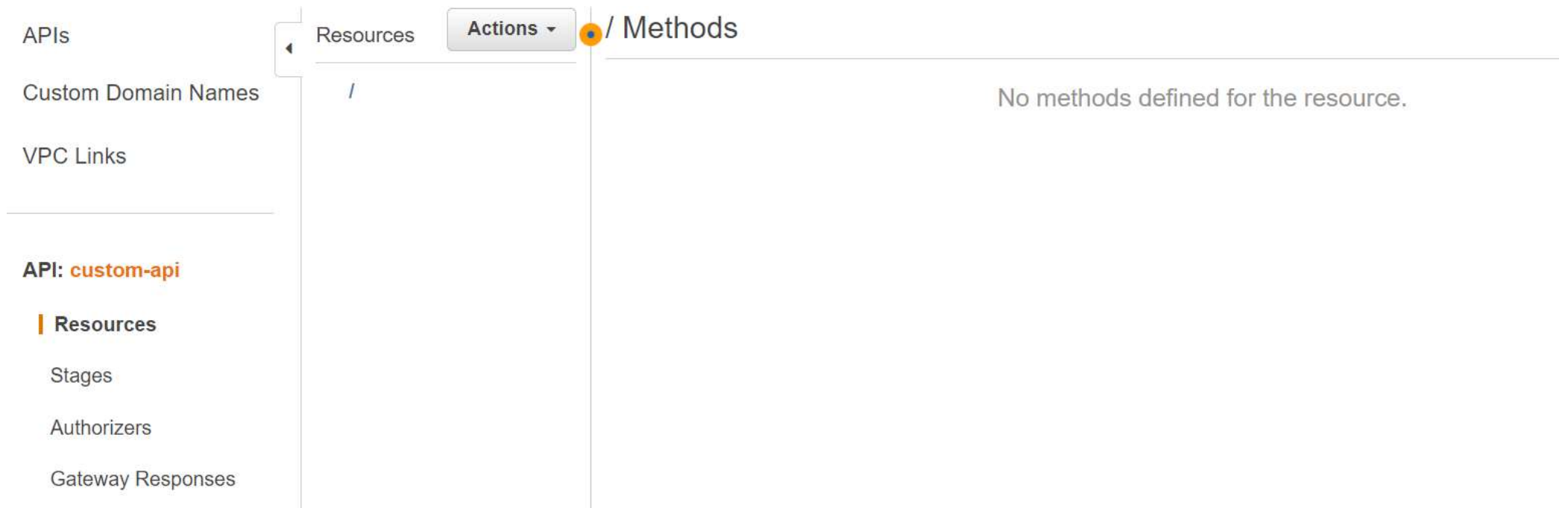
API 생성 완료

- API gateway 생성 완료 화면



API 생성 완료

- API gateway 생성 완료 화면



DynamoDB에 API 전송 테스트

API 리소스 등록

- Actions를 눌러서 새 리소스 생성
- 리소스: API 머리에 해당하는 부분. 다양한 메소드와 결합

New Child Resource

Use this page to create a new child resource for your resource. 

Configure as [proxy resource](#)

☐ 

Resource Name*

customer

Resource Path*

/ customer

You can add path parameters using brackets. For example, the resource path `{username}` represents a path parameter called 'username'. Configuring `/ {proxy+}` as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to `/foo`. To handle requests to `/`, add a new ANY method on the `/` resource.

Enable API Gateway CORS

☐ 

* Required

[Cancel](#)

[Create Resource](#)

API 메서드 등록

- 메서드: GET, POST, PUT, DELETE
- 메서드를 등록할 때 통신 방법을 선택할 수 있다. (Lambda, HTTP 등)

/customer - POST - Setup

Choose the integration point for your new method.

Integration type ☒ Lambda Function ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

Use Lambda Proxy integration ☐ ⓘ

Lambda Region ▼

Lambda Function ⓘ

Use Default Timeout ☒ ⓘ

Lambda 함수 생성

- 기존에 만들어 둔 DynamoDB와 연결을 하는 Lambda 함수를 생성
- DynamoDB에 접근할 수 있는 role을 선택

기본 정보

함수 이름

함수의 용도를 설명하는 이름을 입력합니다.

addCustomer

공백 없이 문자, 숫자, 하이픈 또는 밑줄만 사용합니다.

런타임 정보

함수를 작성하는 데 사용할 언어를 선택합니다. 콘솔 코드 편집

Python 3.9

아키텍처 정보

함수 코드에 대해 원하는 명령 세트 아키텍처를 선택합니다.

☒ x86_64

☐ arm64

▼ 기본 실행 역할 변경

실행 역할

Choose a role that defines the permissions of your function. To create a custom

☐ 기본 Lambda 권한을 가진 새 역할 생성

☒ 기존 역할 사용

☐ AWS 정책 템플릿에서 새 역할 생성

기존 역할

생성한 기존 역할 중에 이 Lambda 함수와 함께 사용할 역할을 선택합니다. 이 역할

aws-learner-dynamodb-role

[View the aws-learner-dynamodb-role role](#) on the IAM console.

Lambda 함수 생성 후 메서드에 장착

- DynamoDB에 접근하는 Lambda 함수를 API gateway 메서드에 장착한다.

Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:ap-northeast-2:453043152051:function:addCustomer

/customer - POST - Setup

Choose the integration point for your new method.

Integration type ☒ Lambda Function ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

Use Lambda Proxy integration ☐ ⓘ

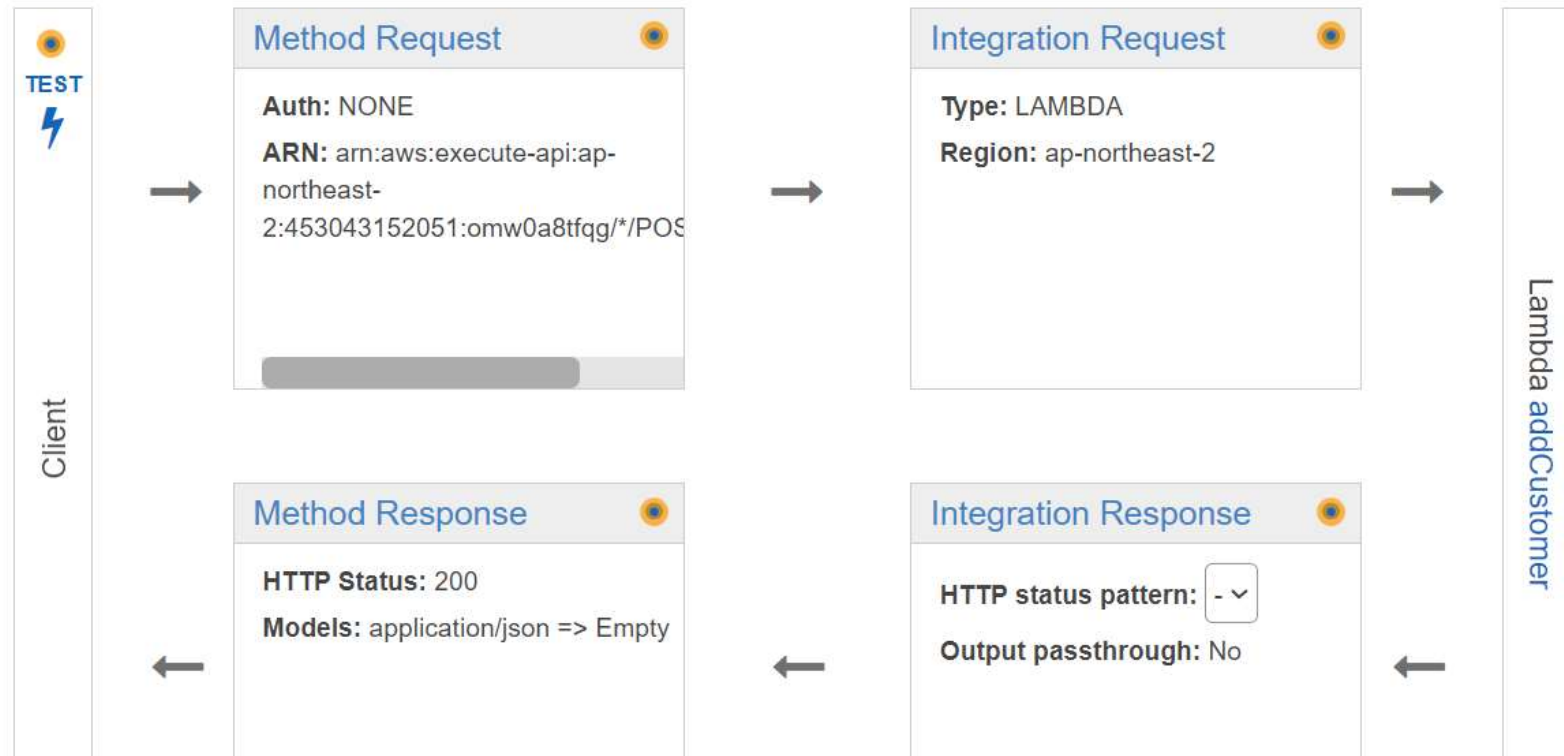
Lambda Region

Lambda Function ⓘ

Use Default Timeout ☒ ⓘ

Lambda 함수 생성 후 메서드에 장착 결과

- 메서드 요청과 응답에 대한 다이어그램 보임



API gateway 테스트

- Test 버튼 클릭
- 아래로 쪽 내려오면 요청 본문이 있음.
- Request Body에 JSON 형태로 입력하여 Test 보낼 수 있다.



Request Body

```
1 {  
2   "customer_id": "38NME",  
3   "customer_name": "John",  
4   "product_name": "chair",  
5   "price": 250000  
6 }
```

DynamoDB에 데이터 삽입

- API gateway가 미리 만들어 둔 "customer" dynamoDB에 데이터 삽입

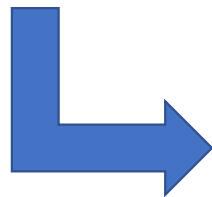
Request Body

```
1 {  
2   "customer_id": "38NME",  
3   "customer_name": "John",  
4   "product_name": "chair",  
5   "price": 258000  
6 }
```

Request: /customer
Status: 200
Latency: 217 ms

Response Body

```
{"code": 200, "message": "Data Successfully Inserted!"}
```



customers

자동 미리 보기 테이블 세부 정보 보기

▶ 항목 스캔 또는 쿼리
확장하여 항목을 쿼리하거나 스캔합니다.

반환된 항목 (1)

🔄 작업 ▼ 항목 생성

< 1 > ⚙️ ✖️

<input type="checkbox"/>	customer_id ▼	customer_na... ▼	price ▼	product_name ▼
<input type="checkbox"/>	38NME	John	258000	chair