جامعة بيرزيت
BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Computer Science Department
COMP 4382 Web Services Technologies

Final Project

Lina Anqawi 1153280

Thursday, 30 April 2020

1.High Level Technical Design :

System Overview

- **Short Description**
  This system used to store all information about renting cars

Project Members and Roles

Student Name	Role
Lina	Restful Service Design
Lina	Middle Tier Design
Lina	Backend and Database
Lina	Front End

Logical Data Model

Customers
Id : int
FirstName : String
LastName : String
LicenceNumber : string

Car
Id: int
brand:string
model:string
pricePerDay:string
available: Boolean;

Booking
Id:int
Customer_Id:int
Car_Id:int
ReservationDate:string
ReturningCarsDate:string
Invoices:string

API Design

1. Customers API

Description

This API is used to manage customer records….

Operation / Function	**HTTP Verb**	**Path**	**Description**
Register New Customer	PUT	/Customers/addnewcustomers	This call creates a new Customer record or updates an existing
List Customers	GET	/Customers /getallcustomers	Get Customers records
Update	patch	/Customers /updatelicencenumber	Update Licence number
delete	Delete	/Customers / deletcustomers	Delete the customers

2. Cars API:

Description

This API is used to manage cars records….

Operation / Function	HTTP Verb	Path	Description
Register New Car	PUT	/Cars/addnewcar	This call creates a new Cars record
List Cars	GET	/Cars /getallcars	Get Cars records
Update price per day		/Cars /updatepriceperdayincars	Update price per day number
delete	GET Update Delete	/cars /deletcars	Delete the cars

3. BOOKING API:

Description

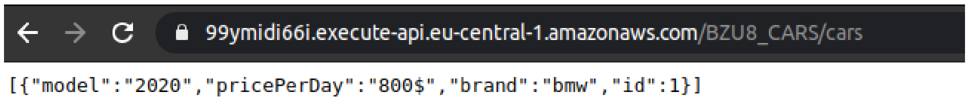
This API is used to manage booking records….

Operation / Function	HTTP Verb	Path	Description
Register New booking	PUT	_/booking/addnewbooking	This call creates a new Cars record
List bookings	GET	/booking /getallcars	Get booking records
Update Returning Cars Date		/booking /updatebooking	Update Returning Cars Date
delete	GET Update Delete	/booking /deletebooking	Delete the booking

2. Sample/ Proof of Concept

1.DOMAIN >> https://99ymidi66i.execute-api.eu-central-1.amazonaws.com/BZU8_CARS

CARS ENTITY:https://99ymidi66i.execute-api.eu-central-1.amazonaws.com/BZU8_CARS/cars

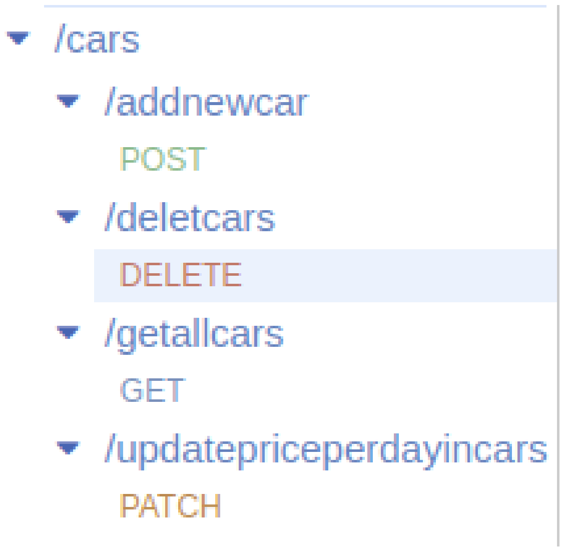

99ymidi66i.execute-api.eu-central-1.amazonaws.com/BZU8_CARS/cars

```
[{"model":"2020","pricePerDay":"800$","brand":"bmw","id":1}]
```

3. Final Application Demo

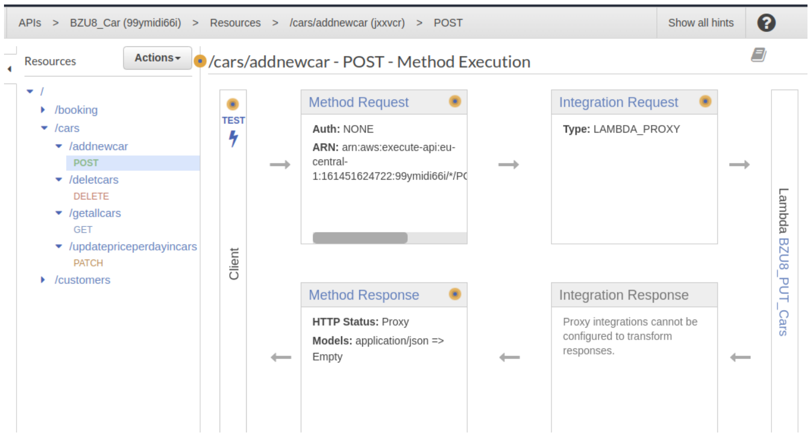
1. Car:

API gate way for car entity


/cars
/addnewcar
POST
/deletcars
DELETE
/getallcars
GET
/updatepriceperdayincars
PATCH

A.BZU8_PUT_Cars :

1. /addNewCar resource connect with BZU8_PUT_Cars Lambda function



APIs > BZU8_Car (99ymidi66i) > Resources > /cars/addnewcar (jxxvcr) > POST

/cars/addnewcar - POST - Method Execution

Resources: / ; /booking ; /cars ; /addnewcar POST ; /deletcars DELETE ; /getallcars GET ; /updatepriceperdayincars PATCH ; /customers

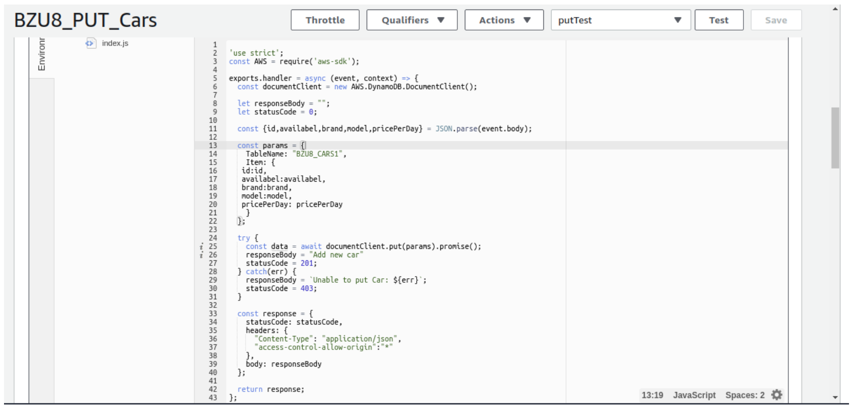
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/PC

Integration Request
Type: LAMBDA_PROXY

Method Response
HTTP Status: Proxy
Models: application/json => Empty

Integration Response
Proxy integrations cannot be configured to transform responses.

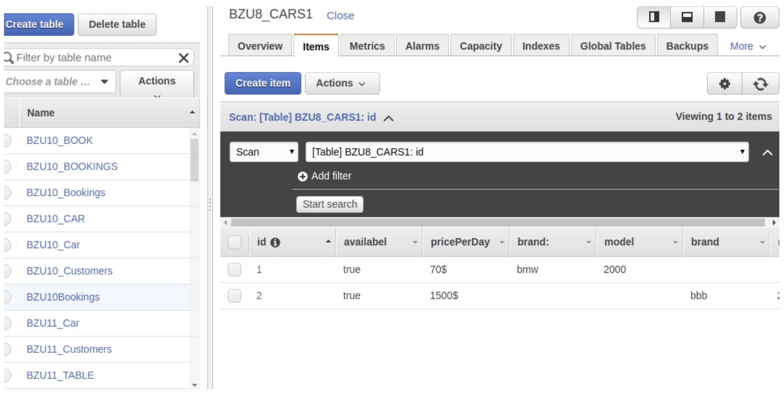
2. Lambda function file wrote by node.js



BZU8_PUT_Cars

```javascript

'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id,availabel,brand,model,pricePerDay} = JSON.parse(event.body);

  const params = {
    TableName: "BZU8_CARS1",
    Item: {
   id:id,
   availabel:availabel,
   brand:brand,
   model:model,
   pricePerDay: pricePerDay
    }
  };

  try {
    const data = await documentClient.put(params).promise();
    responseBody = "Add new car"
    statusCode = 201;
  } catch(err) {
    responseBody = `Unable to put Car: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

3. Cars Table



Name
BZU10_BOOK
BZU10_BOOKINGS
BZU10_Bookings
BZU10_CAR
BZU10_Car
BZU10_Customers
BZU10Bookings
BZU11_Car
BZU11_Customers
BZU11_TABLE

BZU8_CARS1

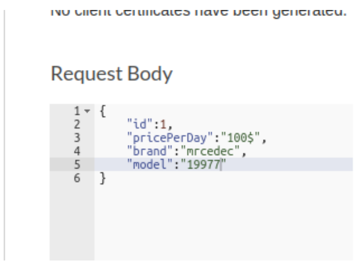
Scan: [Table] BZU8_CARS1: id — Viewing 1 to 2 items

Scan [Table] BZU8_CARS1: id

id	availabel	pricePerDay	brand:	model	brand
1	true	70$	bmw	2000	
2	true	1500$			bbb

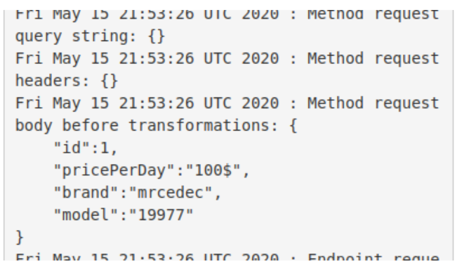
4. Test and result

Request body


No client certificates have been generated.

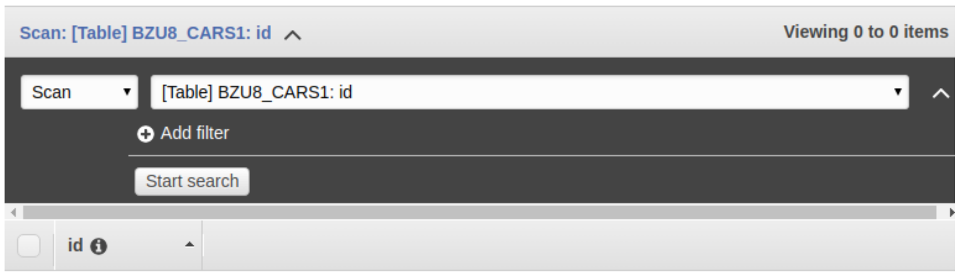
Request Body

```
{
    "id":1,
    "pricePerDay":"100$",
    "brand":"mrcedec",
    "model":"19977"
}
```



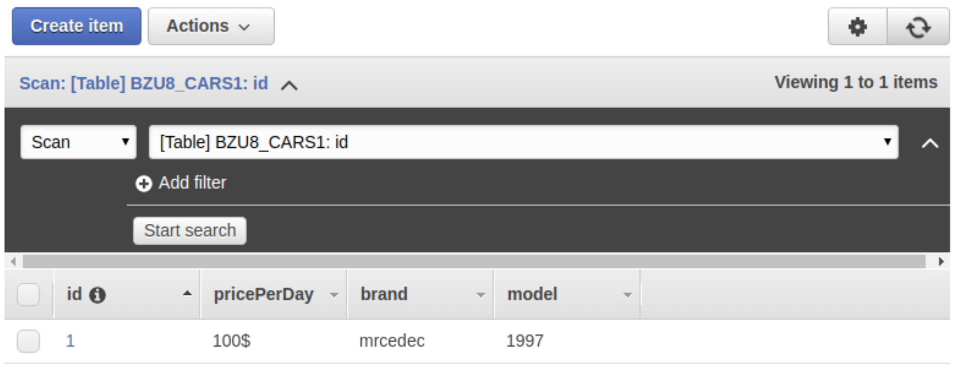
```
Fri May 15 21:53:26 UTC 2020 : Method request
query string: {}
Fri May 15 21:53:26 UTC 2020 : Method request
headers: {}
Fri May 15 21:53:26 UTC 2020 : Method request
body before transformations: {
    "id":1,
    "pricePerDay":"100$",
    "brand":"mrcedec",
    "model":"19977"
}
Fri May 15 21:53:26 UTC 2020 : Endpoint reque
```

Table before testing API



id

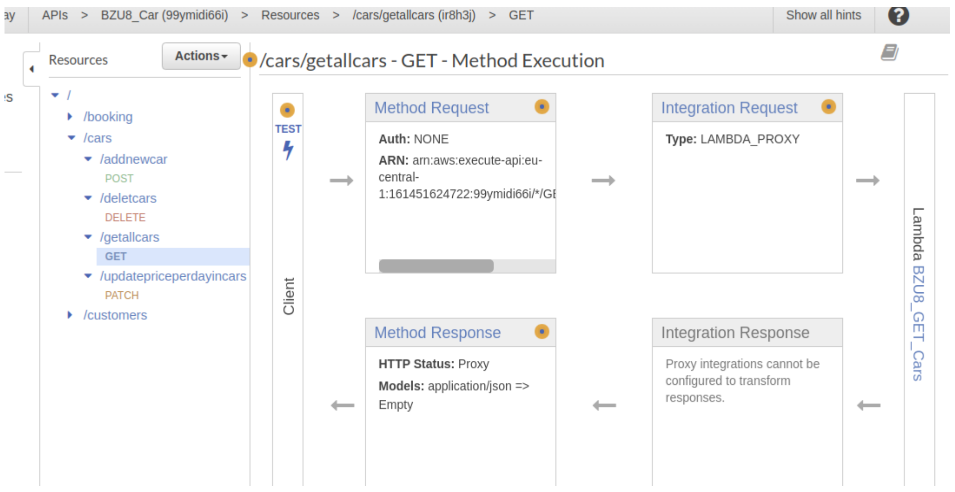
Table after run testing API



id	pricePerDay	brand	model
1	100$	mrcedec	1997

B. BZU8_GET_Cars:

1. /getallcars resource connect with BZU8_GET_Cars Lambda function


ay APIs > BZU8_Car (99ymidi66i) > Resources > /cars/getallcars (ir8h3j) > GET

Show all hints

Resources Actions

/cars/getallcars - GET - Method Execution

- /
  - /booking
  - /cars
    - /addnewcar
      - POST
    - /deletcars
      - DELETE
    - /getallcars
      - GET
    - /updatepriceperdayincars
      - PATCH
  - /customers

TEST

Client

Method Request

Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/GE

Integration Request

Type: LAMBDA_PROXY

Method Response

HTTP Status: Proxy

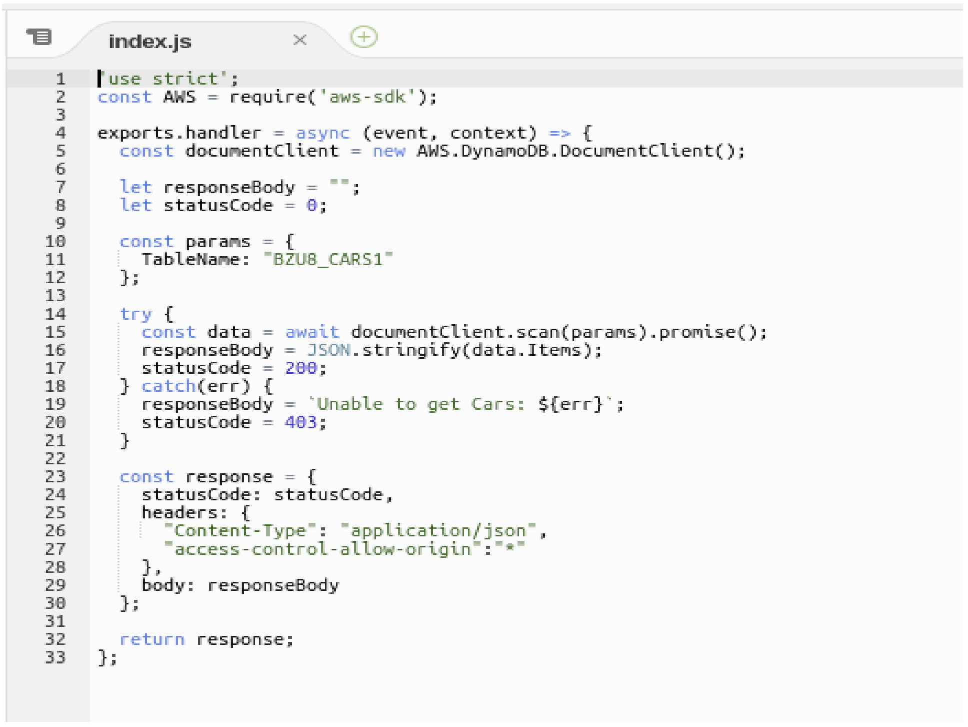
Models: application/json => Empty

Integration Response

Proxy integrations cannot be configured to transform responses.

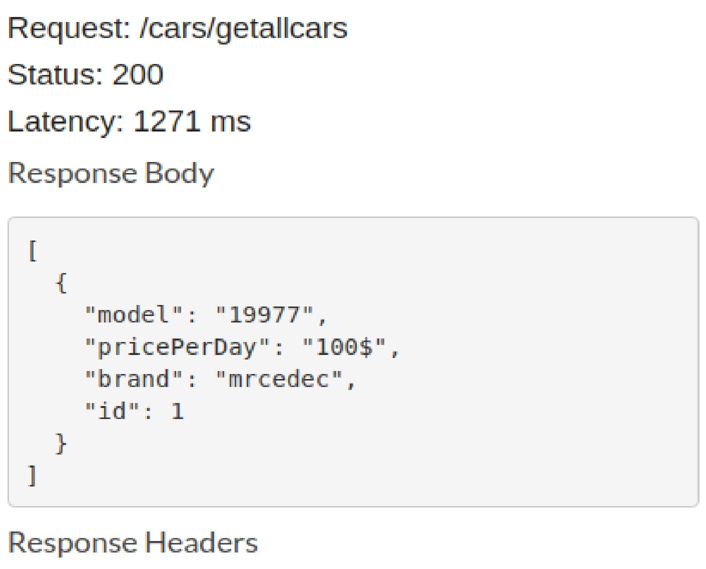
Lambda BZU8_GET_Cars

2. Lambda function file wrote by node.js


index.js

```
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const params = {
    TableName: "BZU8_CARS1"
  };

  try {
    const data = await documentClient.scan(params).promise();
    responseBody = JSON.stringify(data.Items);
    statusCode = 200;
  } catch(err) {
    responseBody = `Unable to get Cars: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

3. Test and result



Request: /cars/getallcars

Status: 200

Latency: 1271 ms

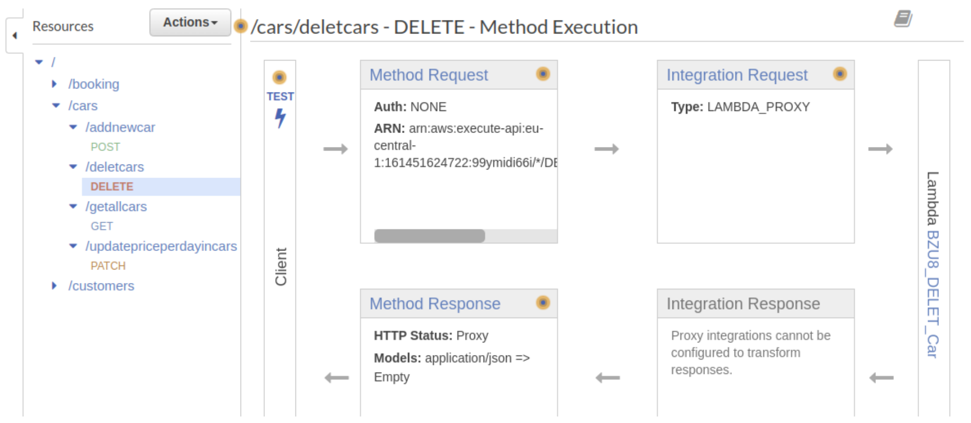
Response Body

```
[
  {
    "model": "19977",
    "pricePerDay": "100$",
    "brand": "mrcedec",
    "id": 1
  }
]
```

Response Headers

c. DELETE_Cars:

1. /deletcars resource connect with DELETE_Cars Lambda function



/cars/deletcars - DELETE - Method Execution

Method Request

Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/DE

Integration Request

Type: LAMBDA_PROXY

Method Response

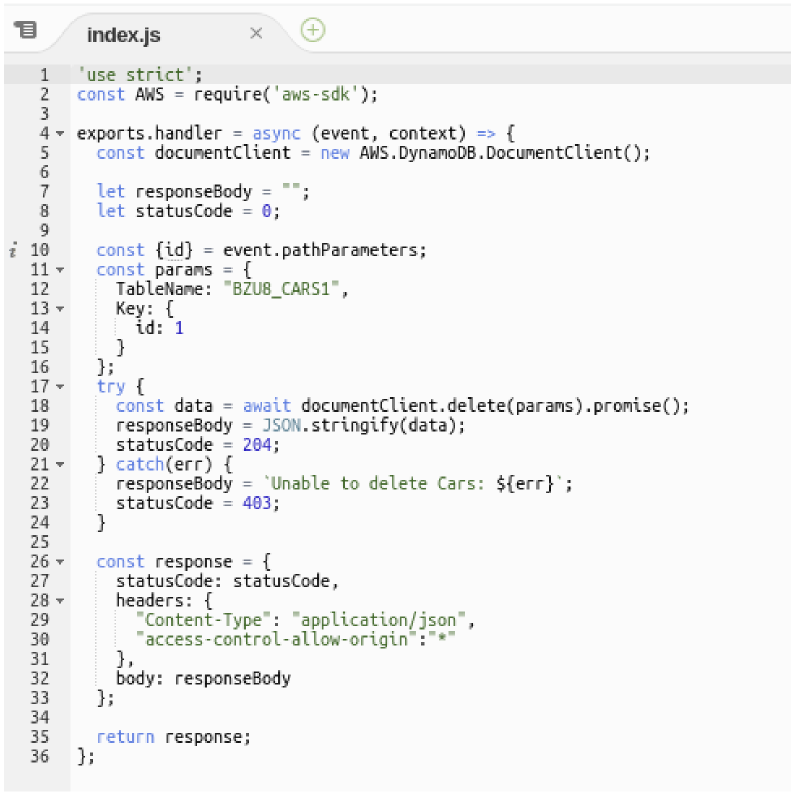
HTTP Status: Proxy

Models: application/json => Empty

Integration Response

Proxy integrations cannot be configured to transform responses.

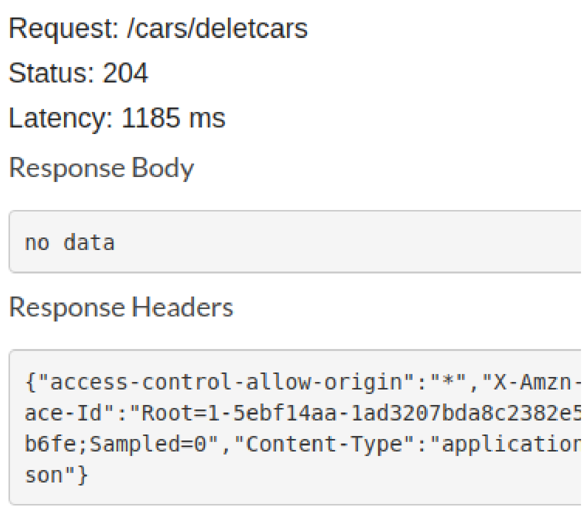
2. Lambda function file wrote by node.js



index.js

```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id} = event.pathParameters;
  const params = {
    TableName: "BZU8_CARS1",
    Key: {
      id: 1
    }
  };
  try {
    const data = await documentClient.delete(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to delete Cars: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

3. Result



Request: /cars/deletcars

Status: 204

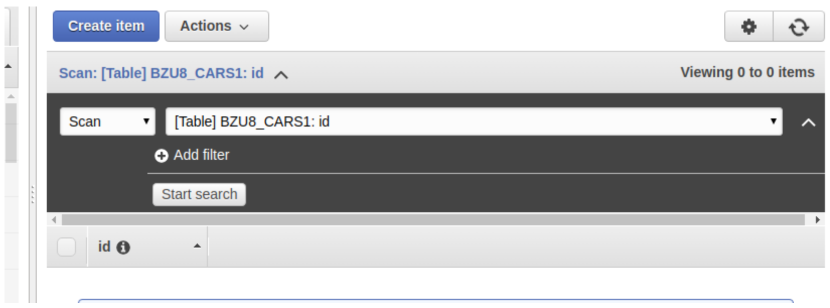
Latency: 1185 ms

Response Body

```
no data
```

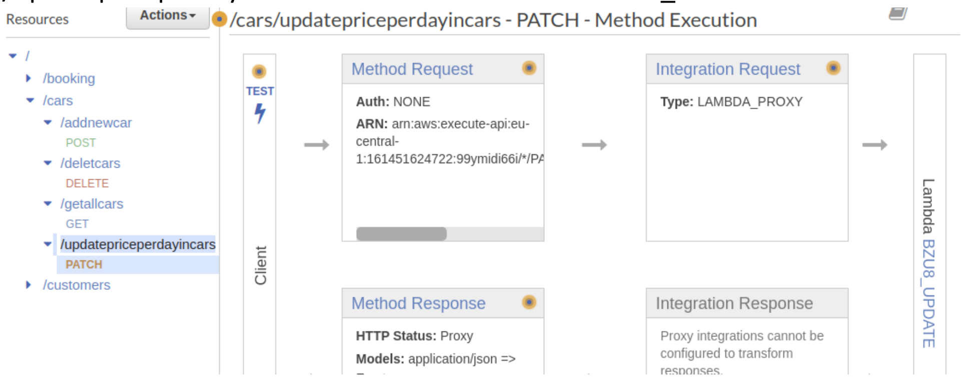
Response Headers

```
{"access-control-allow-origin":"*","X-Amzn-
ace-Id":"Root=1-5ebf14aa-1ad3207bda8c2382e5
b6fe;Sampled=0","Content-Type":"applicatior
son"}
```



D. BZU8_UPDATE:

1. /updatepriceperdayincars resource connect with BZU8_UPDATE Lambdafunction



Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/PA

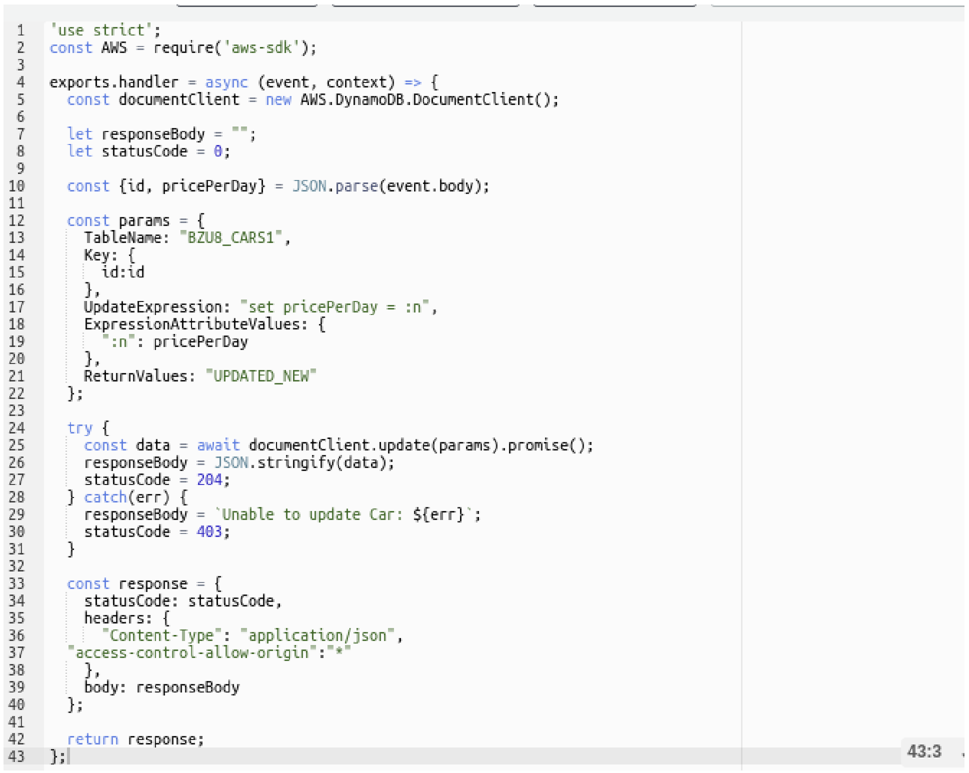
Type: LAMBDA_PROXY

HTTP Status: Proxy

Models: application/json =>

Proxy integrations cannot be configured to transform responses.

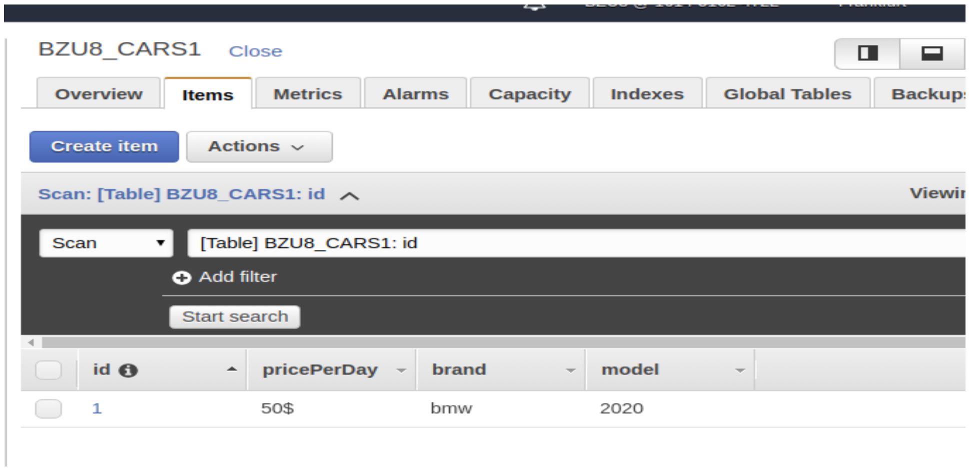
2. Lambda function file wrote by node.js



```
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id, pricePerDay} = JSON.parse(event.body);

  const params = {
    TableName: "BZU8_CARS1",
    Key: {
      id:id
    },
    UpdateExpression: "set pricePerDay = :n",
    ExpressionAttributeValues: {
      ":n": pricePerDay
    },
    ReturnValues: "UPDATED_NEW"
  };

  try {
    const data = await documentClient.update(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to update Car: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
  "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

43:3

3. Test and result



BZU8_CARS1 Close

Overview | Items | Metrics | Alarms | Capacity | Indexes | Global Tables | Backup

Create item | Actions

Scan: [Table] BZU8_CARS1: id

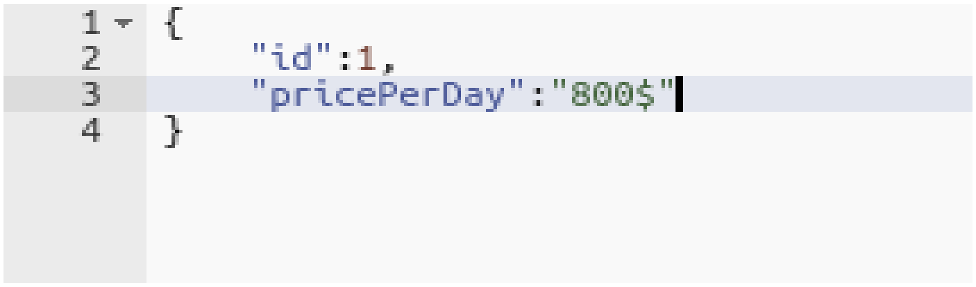
Scan [Table] BZU8_CARS1: id

Add filter

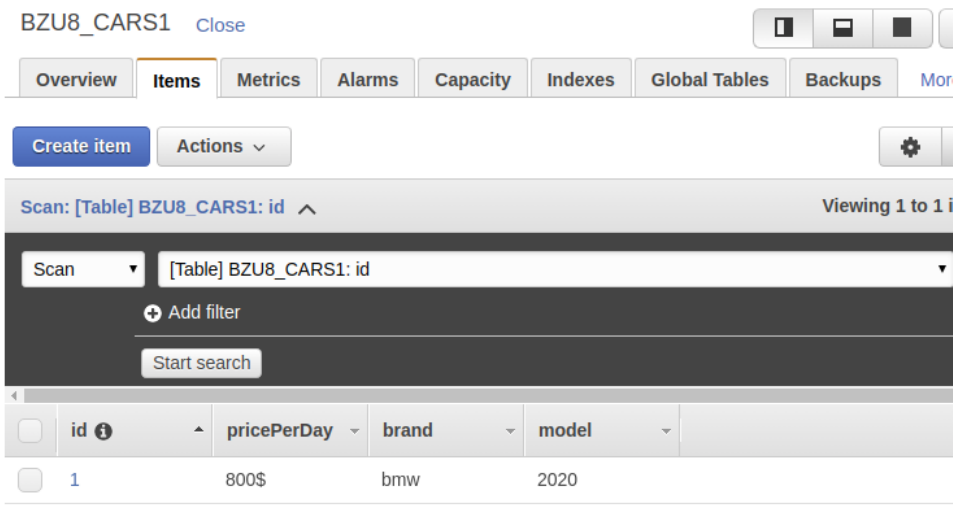
Start search

id	pricePerDay	brand	model
1	50$	bmw	2020

Request Body



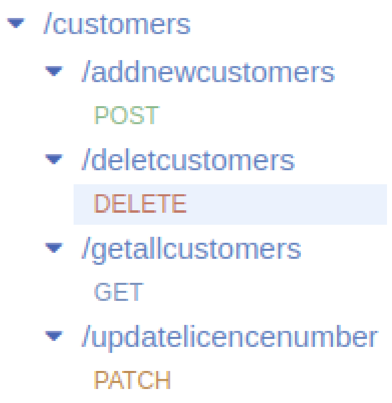
```
{
        "id":1,
        "pricePerDay":"800$"
}
```



id	pricePerDay	brand	model
1	800$	bmw	2020

2.Customers:

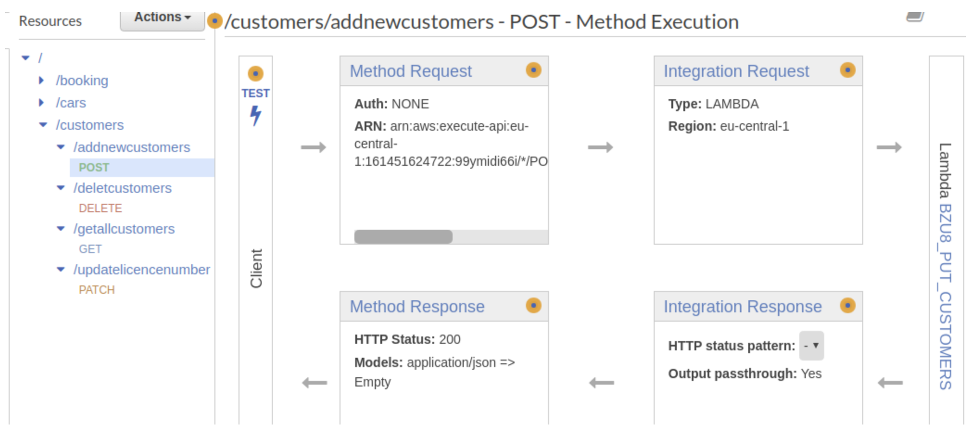
API gate way for customers entity



/customers
/addnewcustomers
POST
/deletcustomers
DELETE
/getallcustomers
GET
/updatelicencenumber
PATCH

A.BZU8_PUT_CUSTOMERS:

1. /addnewcustomers resource connect with BZU8_PUT_Customers Lambdafunction



Resources | Actions | /customers/addnewcustomers - POST - Method Execution

/
/booking
/cars
/customers
/addnewcustomers
POST
/deletcustomers
DELETE
/getallcustomers
GET
/updatelicencenumber
PATCH

TEST

Client

Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/PO

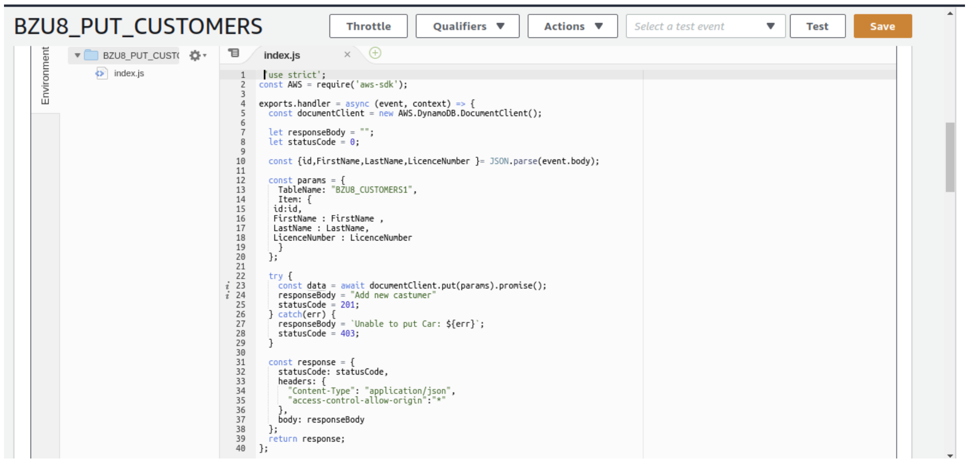
Integration Request
Type: LAMBDA
Region: eu-central-1

Method Response
HTTP Status: 200
Models: application/json => Empty

Integration Response
HTTP status pattern: -
Output passthrough: Yes

Lambda BZU8_PUT_CUSTOMERS

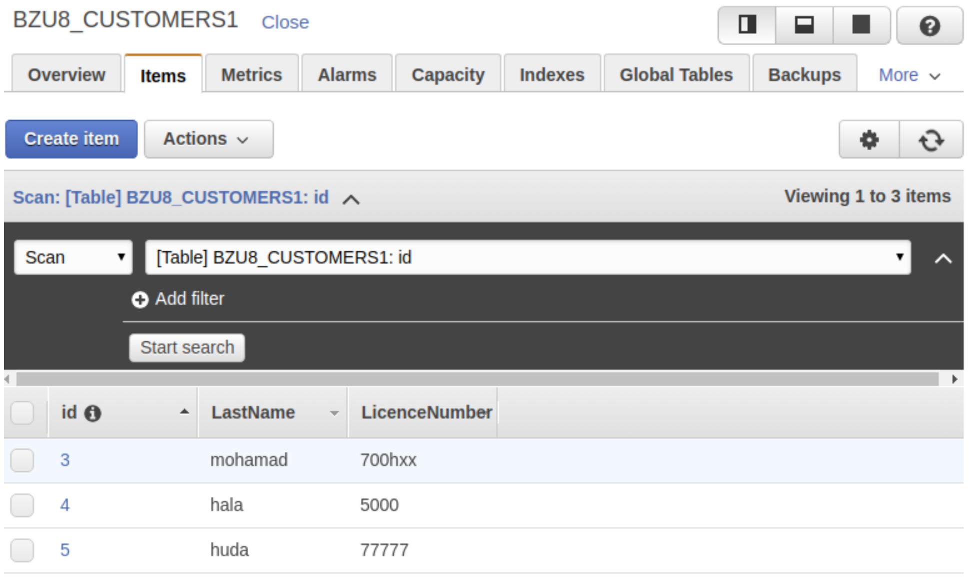
2. Lambda function file wrote by node.js



```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id,FirstName,LastName,LicenceNumber }= JSON.parse(event.body);

  const params = {
    TableName: "BZU8_CUSTOMERS1",
    Item: {
   id:id,
   FirstName : FirstName ,
   LastName : LastName,
   LicenceNumber : LicenceNumber
    }
  };

  try {
    const data = await documentClient.put(params).promise();
    responseBody = "Add new castumer"
    statusCode = 201;
  } catch(err) {
    responseBody = `Unable to put Car: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };
  return response;
};
```

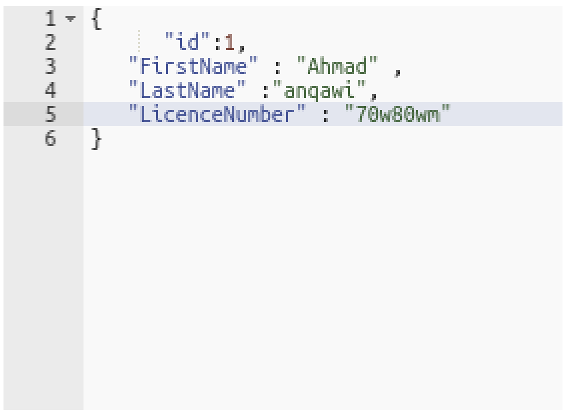
3. Test and result

Before API testing



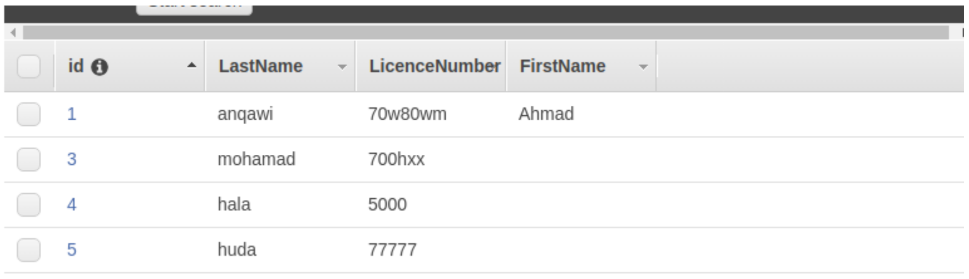
id	LastName	LicenceNumber
3	mohamad	700hxx
4	hala	5000
5	huda	77777

Test API



```
{
    "id":1,
  "FirstName" : "Ahmad" ,
  "LastName" :"anqawi",
  "LicenceNumber" : "70w80wm"
}
```

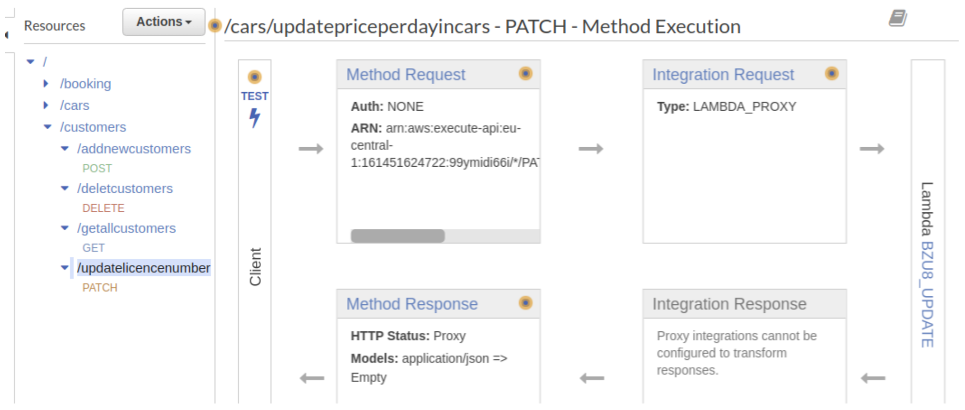
After testing API



id	LastName	LicenceNumber	FirstName
1	anqawi	70w80wm	Ahmad
3	mohamad	700hxx	
4	hala	5000	
5	huda	77777	

B. BZU8_UPDATE_CUSTOMERS:

1. /updatelicencenumber resource connect with BZU8_UPDATE_Customers Lambdafunction



/cars/updatepriceperdayincars - PATCH - Method Execution

Resources: / ; /booking ; /cars ; /customers ; /addnewcustomers POST ; /deletcustomers DELETE ; /getallcustomers GET ; /updatelicencenumber PATCH

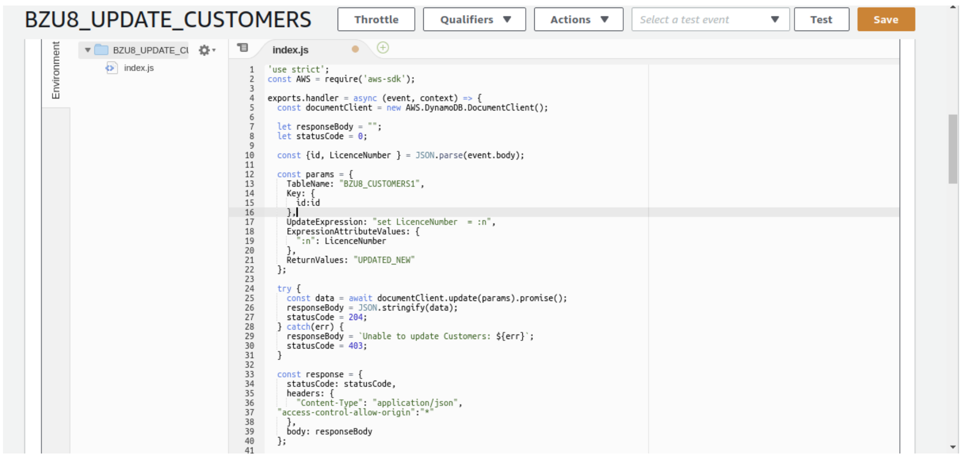
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/PAT

Integration Request
Type: LAMBDA_PROXY

Method Response
HTTP Status: Proxy
Models: application/json => Empty

Integration Response
Proxy integrations cannot be configured to transform responses.

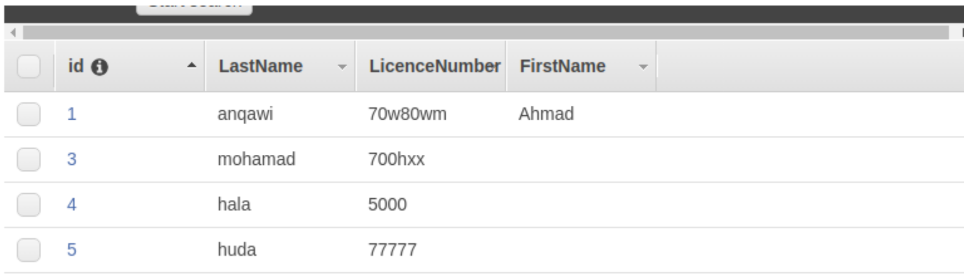
2. Lambda function file wrote by node.js



```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id, LicenceNumber } = JSON.parse(event.body);

  const params = {
    TableName: "BZU8_CUSTOMERS1",
    Key: {
      id:id
    },
    UpdateExpression: "set LicenceNumber  = :n",
    ExpressionAttributeValues: {
      ":n": LicenceNumber
    },
    ReturnValues: "UPDATED_NEW"
  };

  try {
    const data = await documentClient.update(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to update Customers: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
  "access-control-allow-origin":"*"
    },
    body: responseBody
  };

```

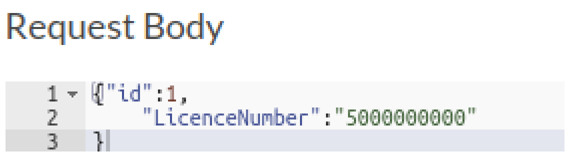
3. Test and result

Before testing



	id	LastName	LicenceNumber	FirstName
	1	anqawi	70w80wm	Ahmad
	3	mohamad	700hxx	
	4	hala	5000	
	5	huda	77777	

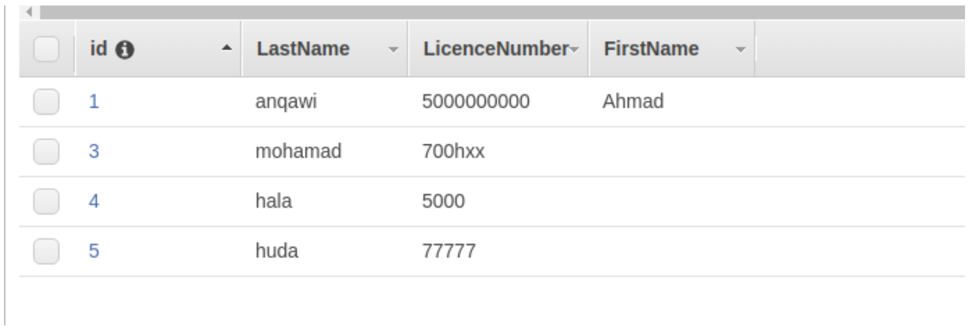
API Test



Request Body

```
{"id":1,
    "LicenceNumber":"5000000000"
}
```

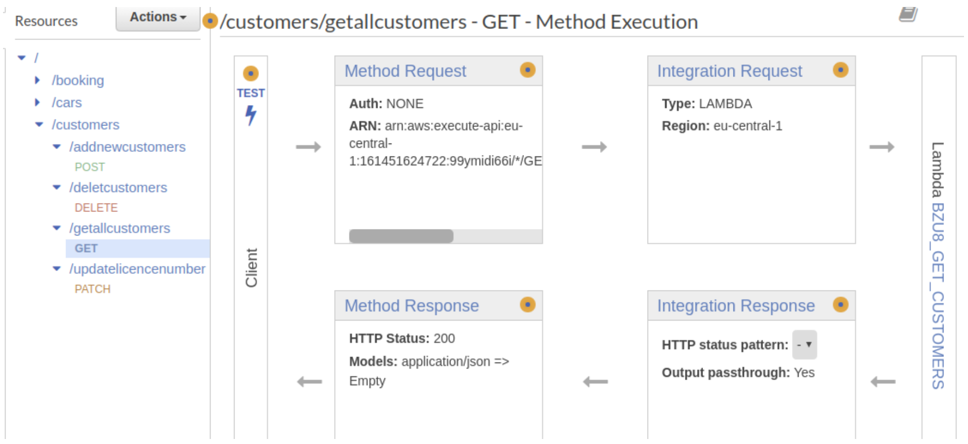
After testing



id	LastName	LicenceNumber	FirstName
1	anqawi	5000000000	Ahmad
3	mohamad	700hxx	
4	hala	5000	
5	huda	77777	

C. BZU8_GET_CUSTOMERS:

1. /getallcustomers resource connect with BZU8_GET_Customers Lambdafunction



/customers/getallcustomers - GET - Method Execution

Resources
- /
  - /booking
  - /cars
  - /customers
    - /addnewcustomers
      - POST
    - /deletcustomers
      - DELETE
    - /getallcustomers
      - GET
    - /updatelicencenumber
      - PATCH

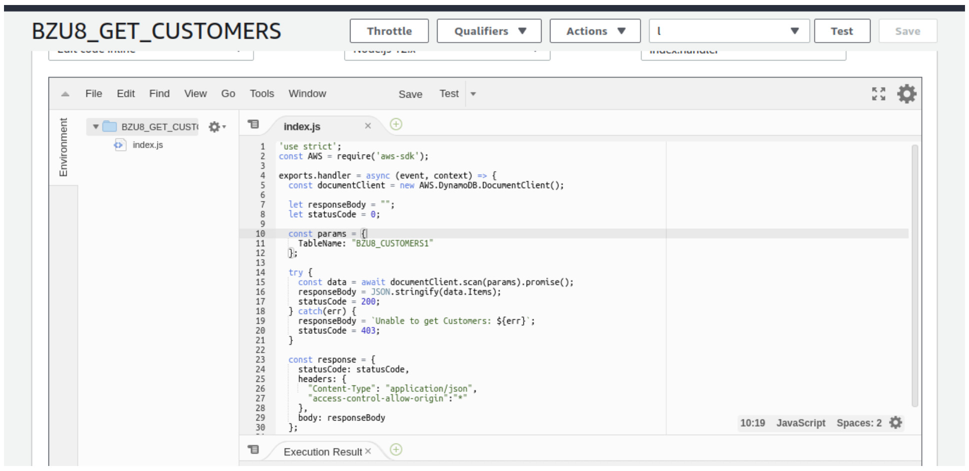
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/GE

Integration Request
Type: LAMBDA
Region: eu-central-1

Method Response
HTTP Status: 200
Models: application/json => Empty

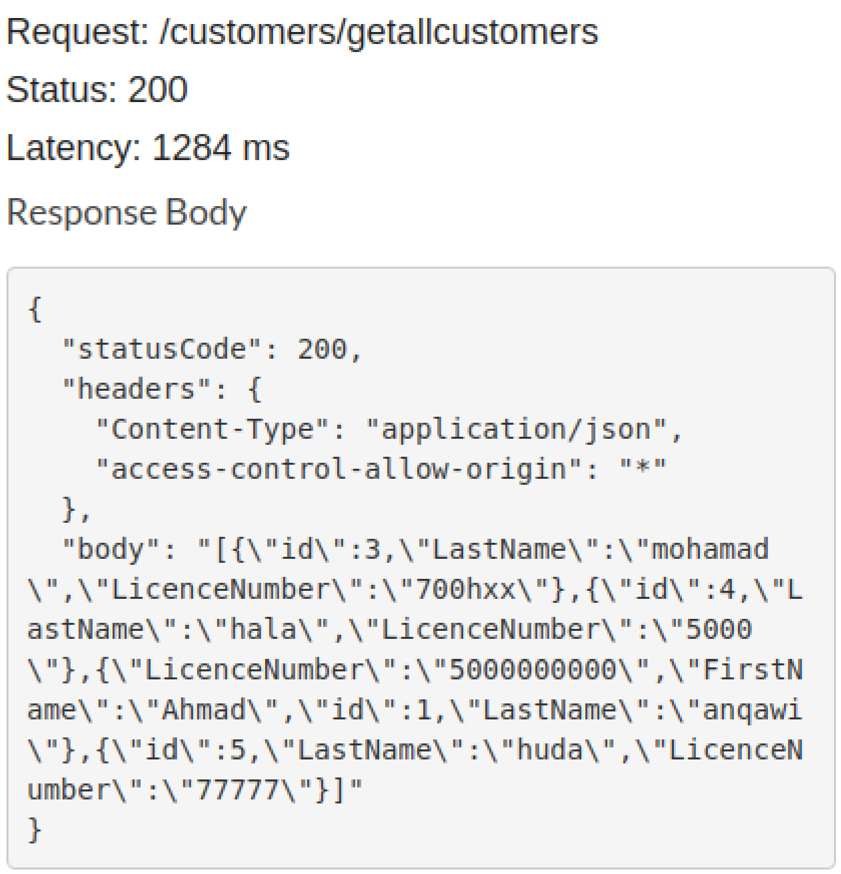
Integration Response
HTTP status pattern: -
Output passthrough: Yes

2. Lambda function file wrote by node.js



```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const params = {
    TableName: "BZU8_CUSTOMERS1"
  };

  try {
    const data = await documentClient.scan(params).promise();
    responseBody = JSON.stringify(data.Items);
    statusCode = 200;
  } catch(err) {
    responseBody = `Unable to get Customers: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };
```

3. Test and result


Request: /customers/getallcustomers

Status: 200

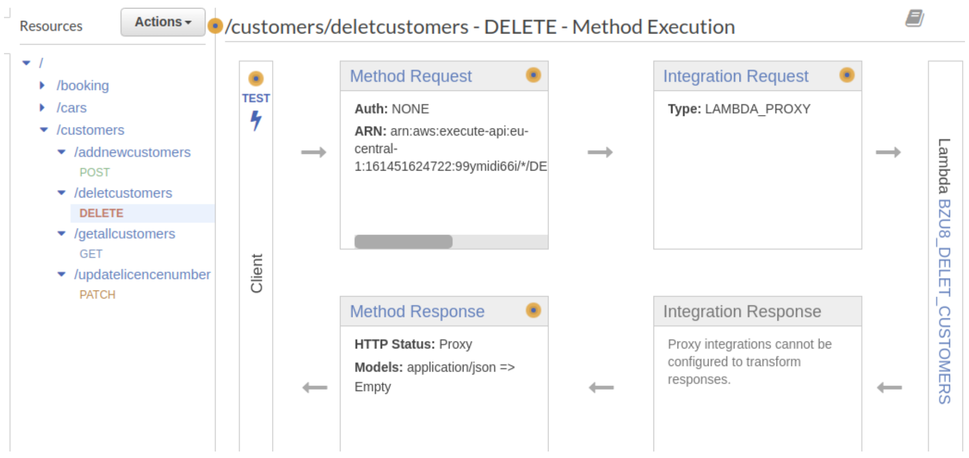
Latency: 1284 ms

Response Body

```
{
  "statusCode": 200,
  "headers": {
    "Content-Type": "application/json",
    "access-control-allow-origin": "*"
  },
  "body": "[{\"id\":3,\"LastName\":\"mohamad
\",\"LicenceNumber\":\"700hxx\"},{\"id\":4,\"L
astName\":\"hala\",\"LicenceNumber\":\"5000
\"},{\"LicenceNumber\":\"5000000000\",\"FirstN
ame\":\"Ahmad\",\"id\":1,\"LastName\":\"anqawi
\"},{\"id\":5,\"LastName\":\"huda\",\"LicenceN
umber\":\"77777\"}]"
}
```

D.BZU8_DELET_CUSTOMERS:

1. /deletcustomers resource connect with BZU8_DELETE_Customers Lambdafunction



/customers/deletcustomers - DELETE - Method Execution

Method Request

Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/DE

Integration Request

Type: LAMBDA_PROXY

Method Response

HTTP Status: Proxy

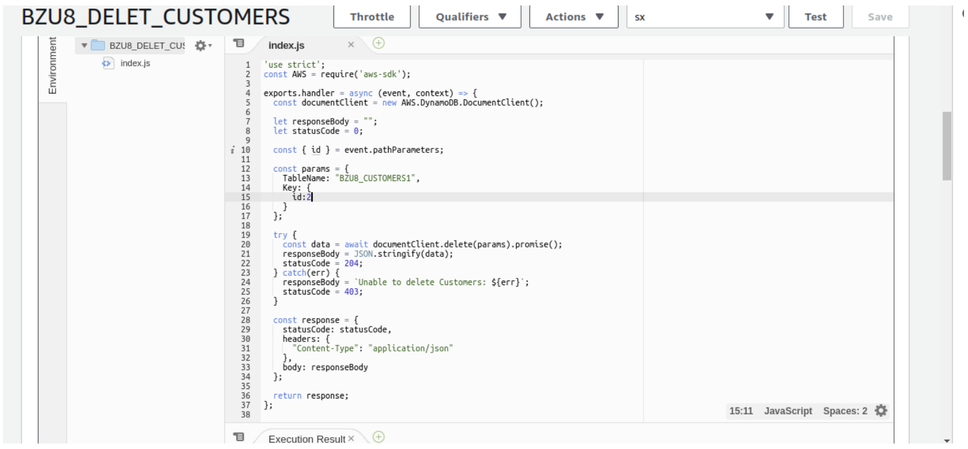
Models: application/json => Empty

Integration Response

Proxy integrations cannot be configured to transform responses.

Lambda BZU8_DELET_CUSTOMERS

2. Lambda function file wrote by node.js

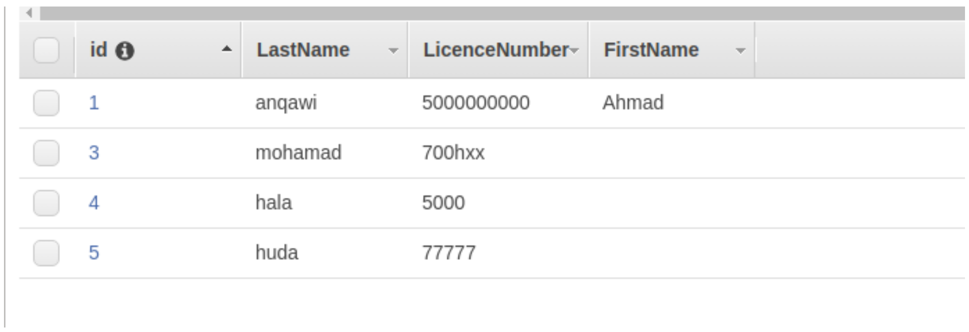


```
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const { id } = event.pathParameters;

  const params = {
    TableName: "BZU8_CUSTOMERS1",
    Key: {
      id:2
    }
  };

  try {
    const data = await documentClient.delete(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to delete Customers: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json"
    },
    body: responseBody
  };

  return response;
};
```

3. Test and result

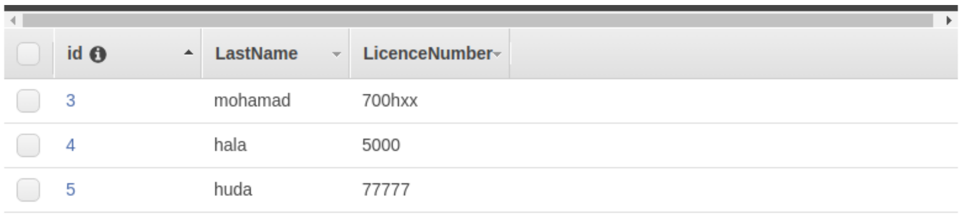
We delete the first row

Before



	id	LastName	LicenceNumber	FirstName
	1	anqawi	5000000000	Ahmad
	3	mohamad	700hxx	
	4	hala	5000	
	5	huda	77777	

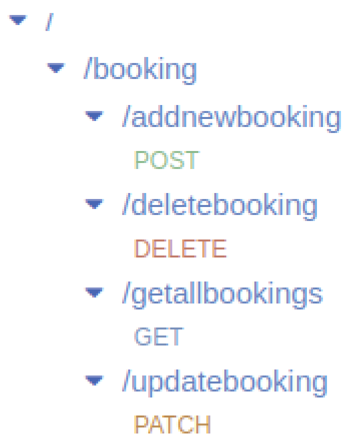
After



	id	LastName	LicenceNumber
	3	mohamad	700hxx
	4	hala	5000
	5	huda	77777

3.Booking:

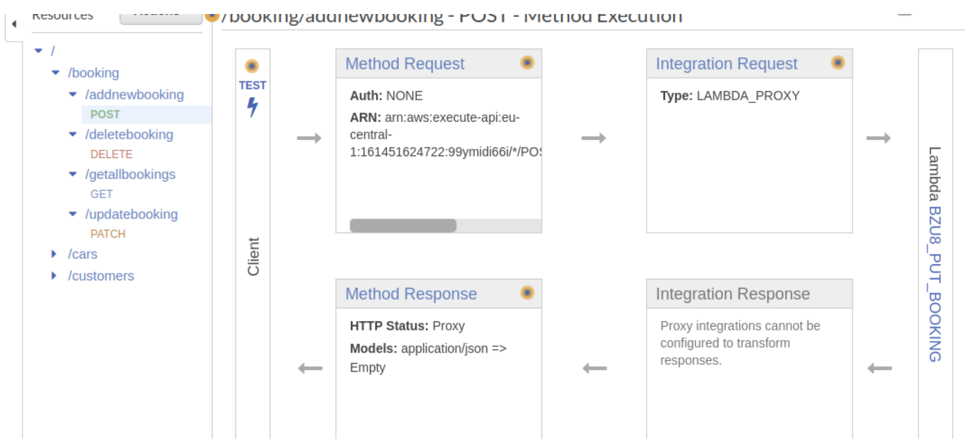
API gate way for Booking entity



- /
  - /booking
    - /addnewbooking
      - POST
    - /deletebooking
      - DELETE
    - /getallbookings
      - GET
    - /updatebooking
      - PATCH

A.BZU8_PUT_BOOKING :

1. /addnewbooking resource connect with BZU8_PUT_BOOKING Lambdafunction



Resources Actions /booking/addnewbooking - POST - Method Execution

- /
  - /booking
    - /addnewbooking
      - POST
    - /deletebooking
      - DELETE
    - /getallbookings
      - GET
    - /updatebooking
      - PATCH
  - /cars
  - /customers

Method Request

Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/POS

Integration Request

Type: LAMBDA_PROXY

Method Response

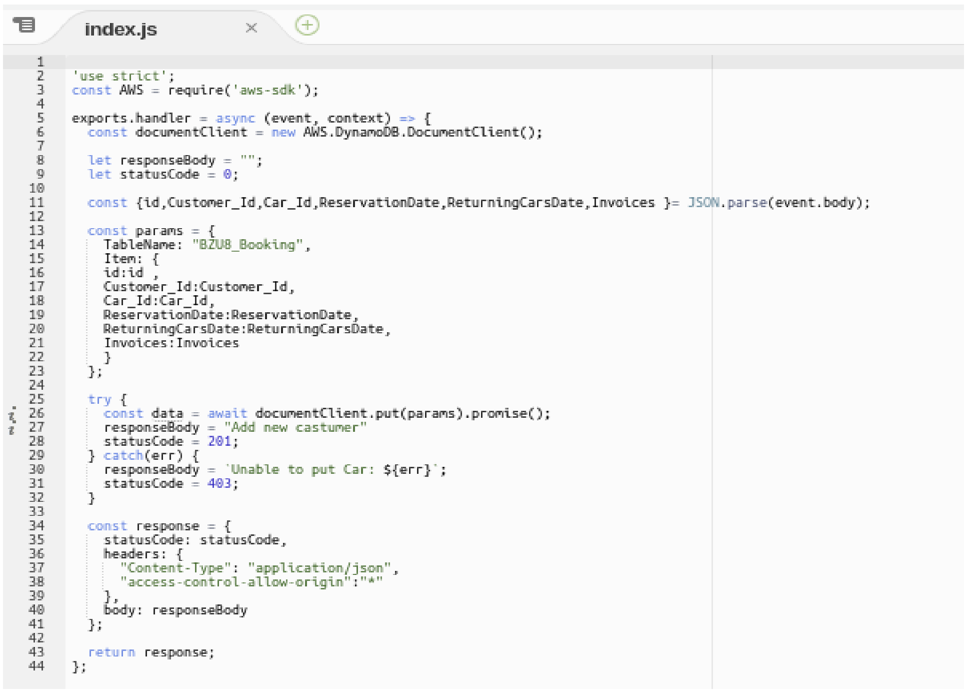
HTTP Status: Proxy

Models: application/json => Empty

Integration Response

Proxy integrations cannot be configured to transform responses.

2. Lambda function file wrote by node.js

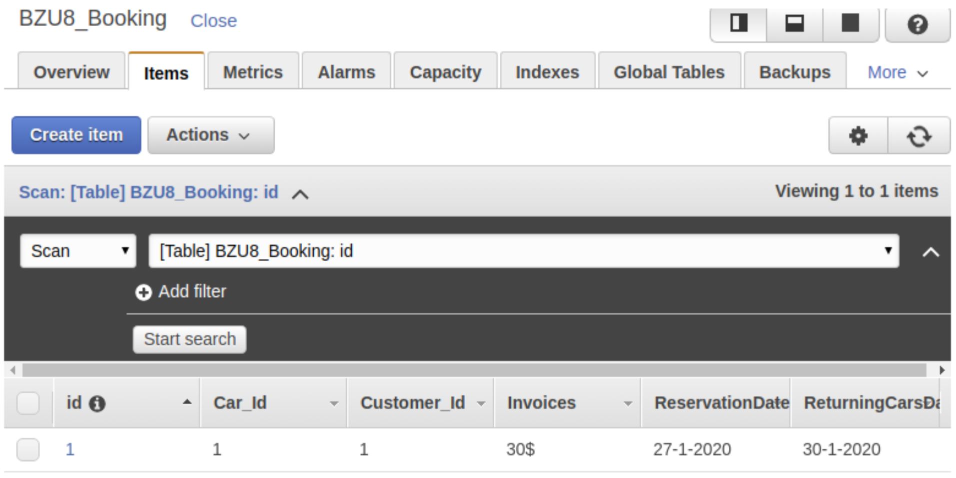


index.js

```

'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id,Customer_Id,Car_Id,ReservationDate,ReturningCarsDate,Invoices }= JSON.parse(event.body);

  const params = {
    TableName: "BZU8_Booking",
    Item: {
    id:id ,
    Customer_Id:Customer_Id,
    Car_Id:Car_Id,
    ReservationDate:ReservationDate,
    ReturningCarsDate:ReturningCarsDate,
    Invoices:Invoices
    }
  };

  try {
    const data = await documentClient.put(params).promise();
    responseBody = "Add new castumer"
    statusCode = 201;
  } catch(err) {
    responseBody = `Unable to put Car: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

3. Test and Result

Before testing API



BZU8_Booking Close

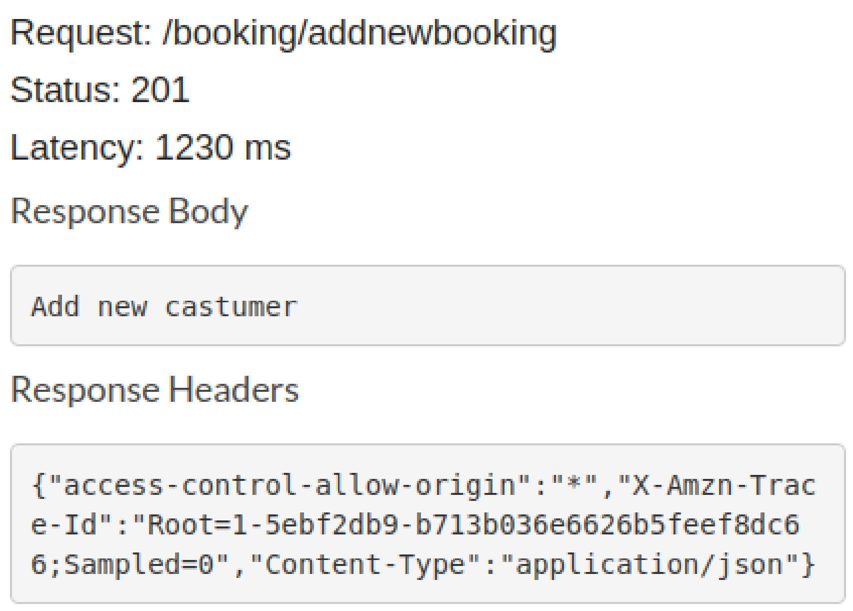
Scan: [Table] BZU8_Booking: id

Viewing 1 to 1 items

Scan [Table] BZU8_Booking: id

id	Car_Id	Customer_Id	Invoices	ReservationDate	ReturningCarsDa
1	1	1	30$	27-1-2020	30-1-2020

After testing API



Request: /booking/addnewbooking

Status: 201

Latency: 1230 ms

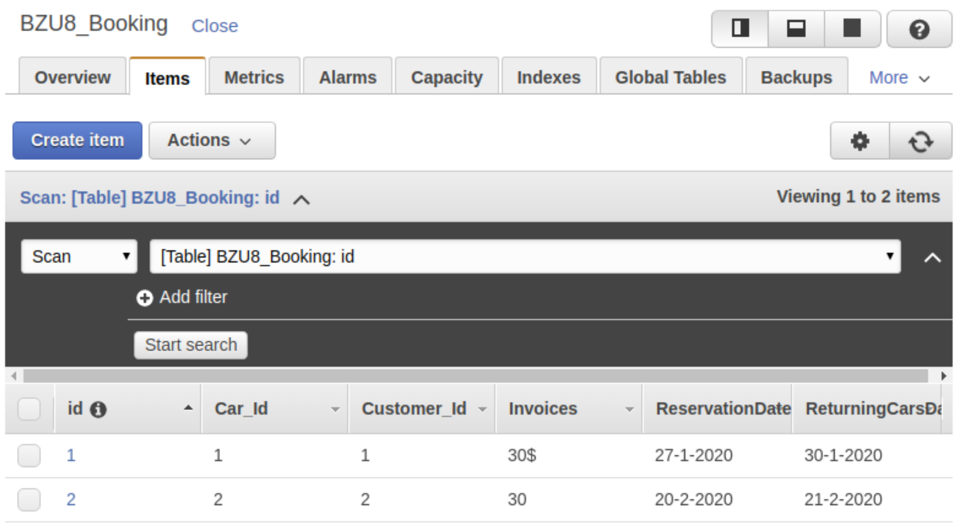
Response Body

```
Add new castumer
```

Response Headers

```
{"access-control-allow-origin":"*","X-Amzn-Trac
e-Id":"Root=1-5ebf2db9-b713b036e6626b5feef8dc6
6;Sampled=0","Content-Type":"application/json"}
```

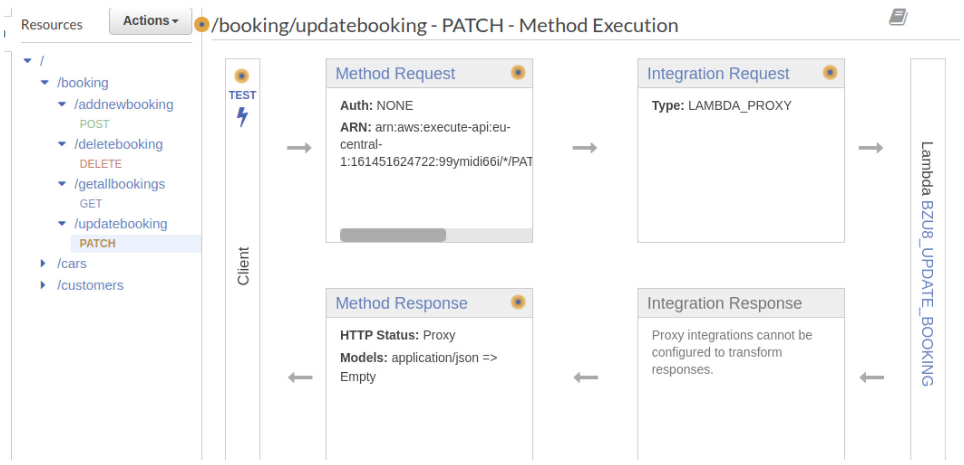
.



id	Car_Id	Customer_Id	Invoices	ReservationDate	ReturningCarsDa
1	1	1	30$	27-1-2020	30-1-2020
2	2	2	30	20-2-2020	21-2-2020

B. BZU8_UPDATE_BOOKING

1. /updatebooking resource connect with BZU8_UPDATE_BOOKING Lambdafunction



Auth: NONE

ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/PAT

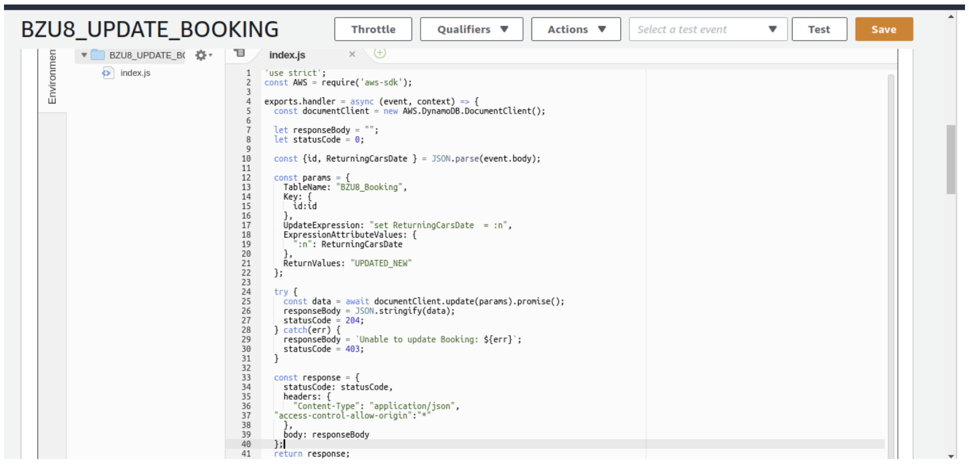
Type: LAMBDA_PROXY

HTTP Status: Proxy

Models: application/json => Empty

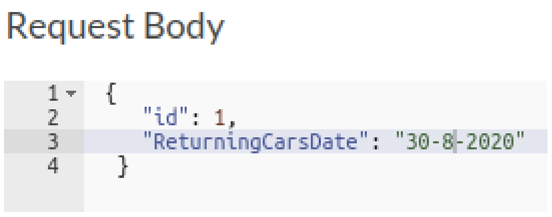
Proxy integrations cannot be configured to transform responses.

2. Lambda function file wrote by node.js



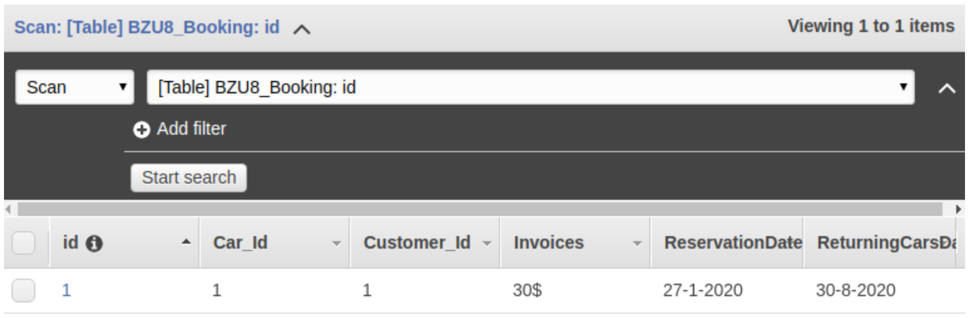
```
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const {id, ReturningCarsDate } = JSON.parse(event.body);

  const params = {
    TableName: "BZU8_Booking",
    Key: {
      id:id
    },
    UpdateExpression: "set ReturningCarsDate  = :n",
    ExpressionAttributeValues: {
      ":n": ReturningCarsDate
    },
    ReturnValues: "UPDATED_NEW"
  };

  try {
    const data = await documentClient.update(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to update Booking: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
  "access-control-allow-origin":"*"
    },
    body: responseBody
  };
  return response;
```

3. Test and Result



Request Body

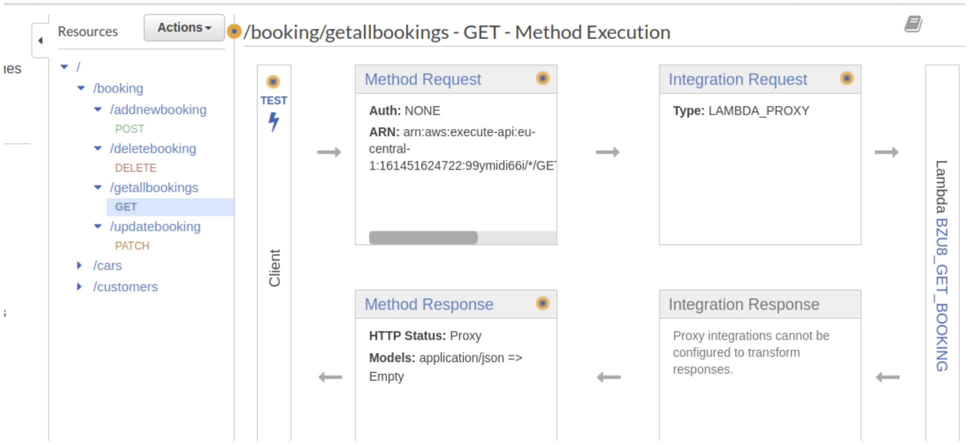
```
{
    "id": 1,
    "ReturningCarsDate": "30-8-2020"
  }
```



id	Car_Id	Customer_Id	Invoices	ReservationDate	ReturningCarsDa
1	1	1	30$	27-1-2020	30-8-2020

C. BZU8_GET_BOOKING

1. /get allbookings resource connect with BZU8_GET_BOOKING Lambdafunction



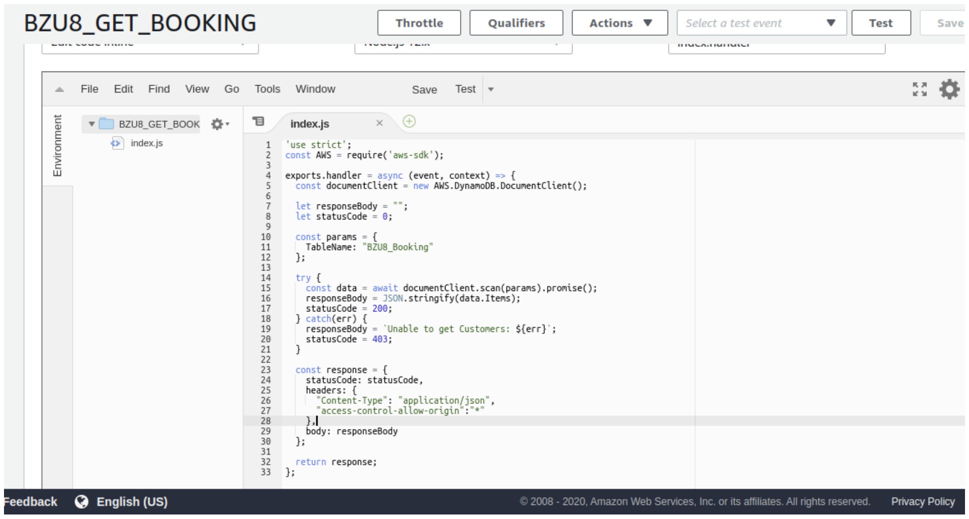
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/GET

Type: LAMBDA_PROXY

HTTP Status: Proxy
Models: application/json => Empty

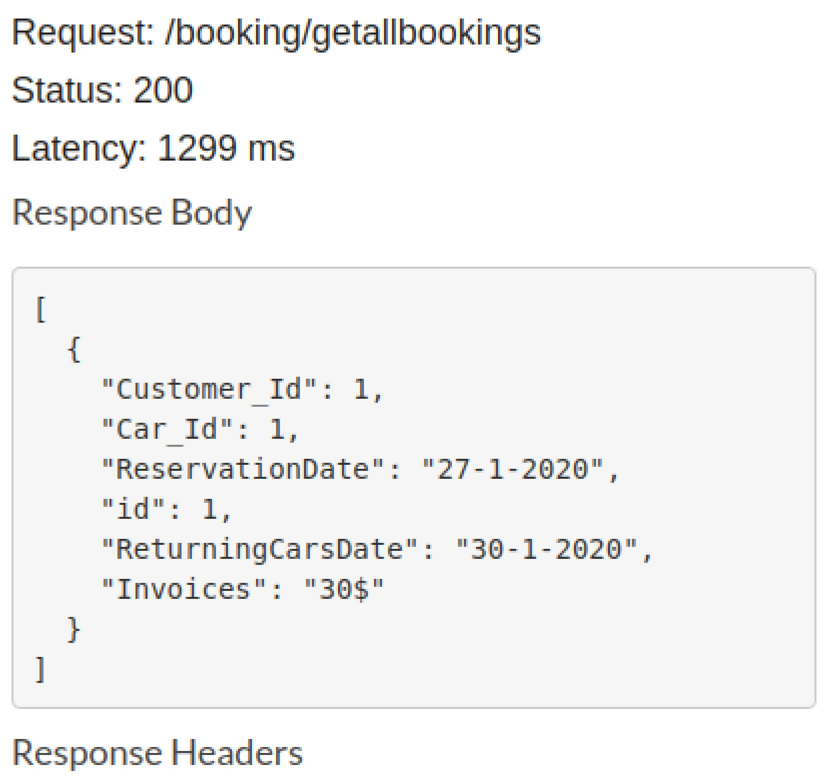
Proxy integrations cannot be configured to transform responses.

2. Lambda function file wrote by node.js



```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const params = {
    TableName: "BZU8_Booking"
  };

  try {
    const data = await documentClient.scan(params).promise();
    responseBody = JSON.stringify(data.Items);
    statusCode = 200;
  } catch(err) {
    responseBody = `Unable to get Customers: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json",
      "access-control-allow-origin":"*"
    },
    body: responseBody
  };

  return response;
};
```

3. Test and Result



Request: /booking/getallbookings

Status: 200

Latency: 1299 ms

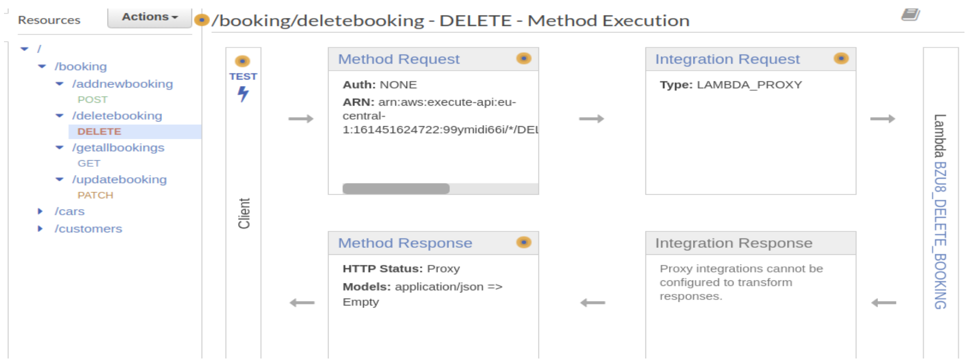
Response Body

```
[
  {
    "Customer_Id": 1,
    "Car_Id": 1,
    "ReservationDate": "27-1-2020",
    "id": 1,
    "ReturningCarsDate": "30-1-2020",
    "Invoices": "30$"
  }
]
```

Response Headers

D.BZU8_DELETE_BOOKING

1. /deletebooking resource connect with BZU8_DELETE_BOOKING Lambdafunction



/booking/deletebooking - DELETE - Method Execution

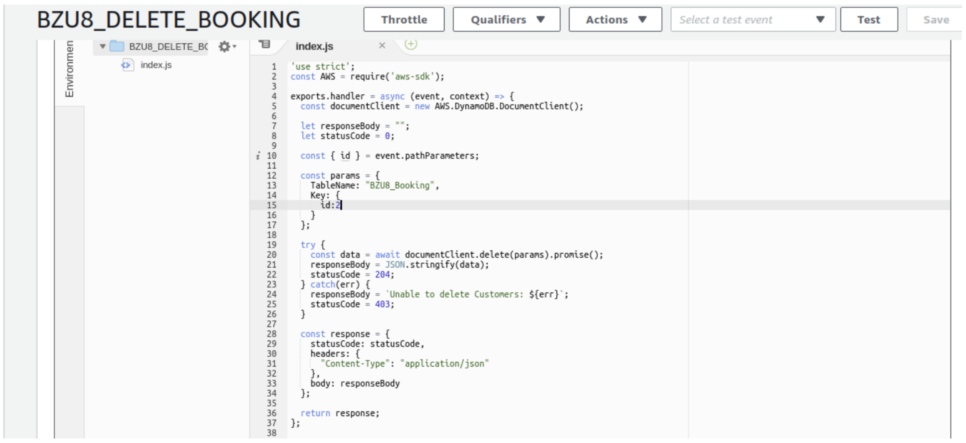
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:161451624722:99ymidi66i/*/DEL

Type: LAMBDA_PROXY

HTTP Status: Proxy
Models: application/json => Empty

Proxy integrations cannot be configured to transform responses.

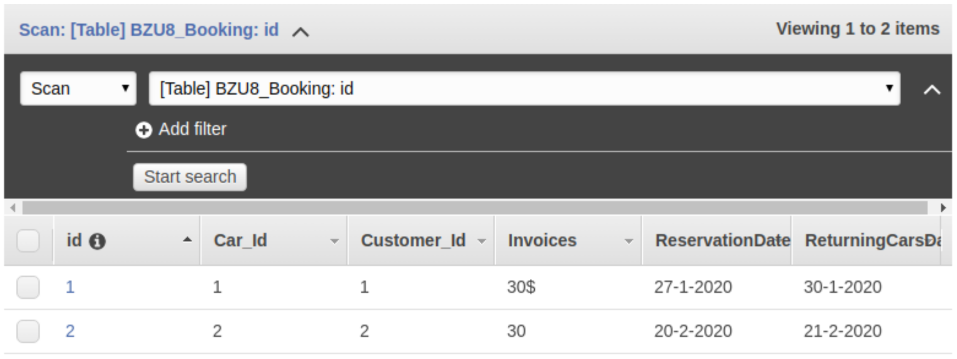
2. Lambda function file wrote by node.js


BZU8_DELETE_BOOKING

```javascript
'use strict';
const AWS = require('aws-sdk');

exports.handler = async (event, context) => {
  const documentClient = new AWS.DynamoDB.DocumentClient();

  let responseBody = "";
  let statusCode = 0;

  const { id } = event.pathParameters;

  const params = {
    TableName: "BZU8_Booking",
    Key: {
      id:2
    }
  };

  try {
    const data = await documentClient.delete(params).promise();
    responseBody = JSON.stringify(data);
    statusCode = 204;
  } catch(err) {
    responseBody = `Unable to delete Customers: ${err}`;
    statusCode = 403;
  }

  const response = {
    statusCode: statusCode,
    headers: {
      "Content-Type": "application/json"
    },
    body: responseBody
  };

  return response;
};
```

3. Test and Result

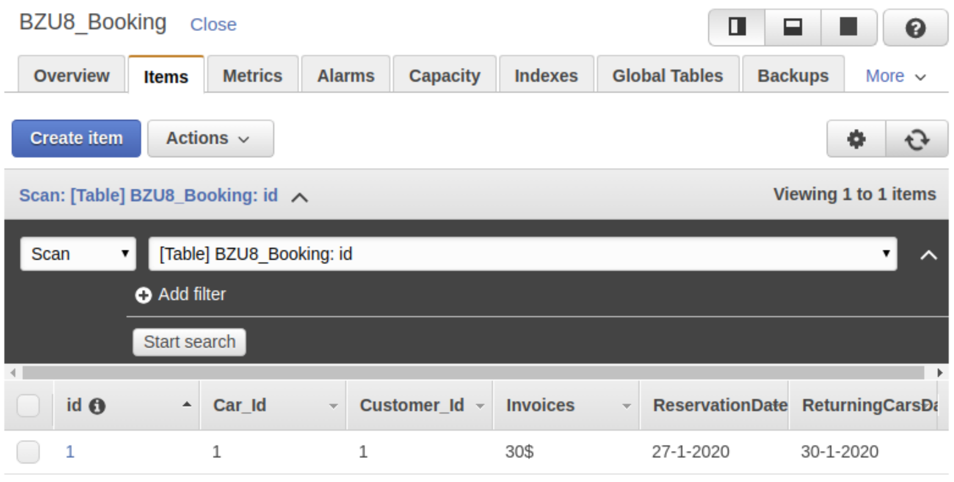
Before testing


Scan: [Table] BZU8_Booking: id

Viewing 1 to 2 items

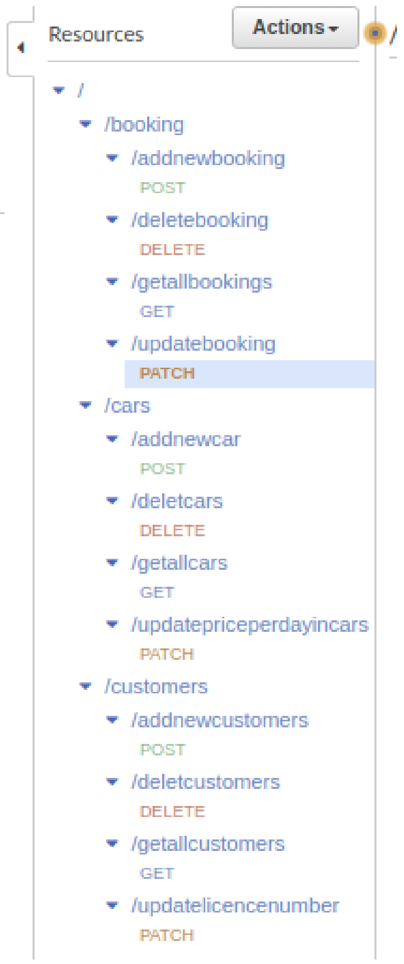
id	Car_Id	Customer_Id	Invoices	ReservationDate	ReturningCarsDa
1	1	1	30$	27-1-2020	30-1-2020
2	2	2	30	20-2-2020	21-2-2020

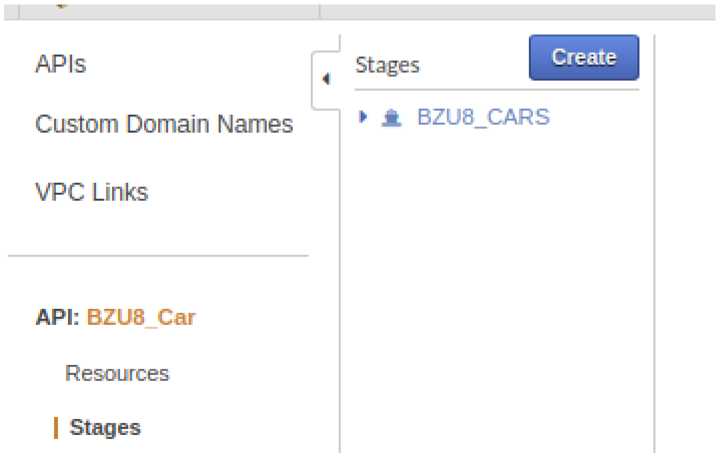
After testing



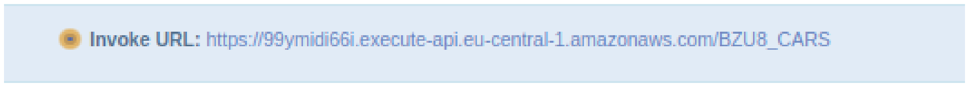
id	Car_Id	Customer_Id	Invoices	ReservationDate	ReturningCarsDa
1	1	1	30$	27-1-2020	30-1-2020

DEPLOY API:





API:


Invoke URL: https://99ymidi66i.execute-api.eu-central-1.amazonaws.com/BZU8_CARS

URL: https://99ymidi66i.execute-api.eu-central-1.amazonaws.com/BZU8_CARS