# Swaddle Swap Service Layer Design

## Overview

Swaddle Swap will have a recommendation engine and some key areas where data will be used to generate content dynamically on the screen. This makes it a perfect candidate for RESTful API Service Layers. For this application, since I will be using AWS to host and AWS DynamoDB for the database, I am going to use AWS Lambda for the functions and AWS API Gateway for my service layer. The Service Layer is divided into several different components:

- API Gateway Resources – This is a collection of programmable entities to use in our API. Each resource can have one or more method resources. These methods tell us what to do with the data. For example: GET, POST, DELETE, etc.
- Lambda Functions – Lambda will serve as the integration layer that hosts functions to call the data from the database. Within Lamba we can create filters and other code to pull back data that we specify.
- Endpoints – For this application we will have two endpoints. The front layer that will be hosted on an AWS instance and the database that will be hosted in Dynamo DB.

## Specification Overview

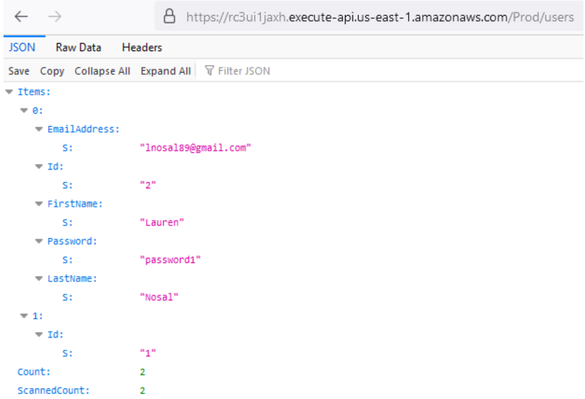
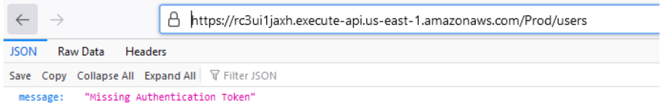
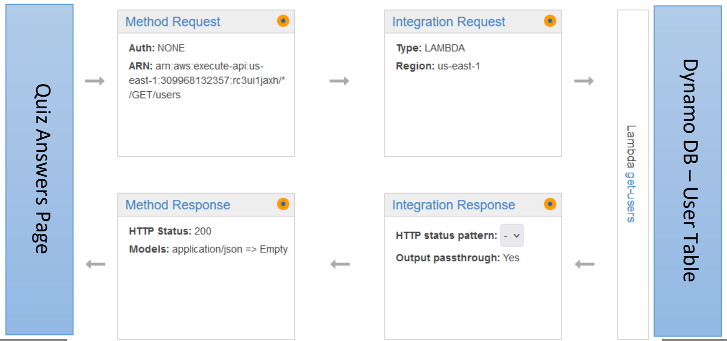
## Routes

This application will have a route for the following data types:

- User Route – the user route will facilitate passing data from the quiz screen to the database as well as can serve to call user specific data when personalization is important.
- Quiz Answers Route – the quiz answers route will help update and store the data for the quiz answers as well as allow us to get the quiz answers to power the swaddle recommendations. For the MVP, the user will not be able to update their answers.
- Swaddle Route – the swaddle route will allow us to get the swaddles based on the quiz answers. These will be displayed on the recommendations page.
- Recommendations Page Route – the recommendations page route will be used to power the recommendations page. The recommendations page will GET the quiz data and then GET the swaddle IDs based on attributes in the quiz answers.
- (Stretch) Rental Cart Route – the rental cart route will be used to power the rental cart screen and the payment screen. The rental cart screen will GET the quiz data and then GET the swaddle IDs based on their attributes and how closely they match the quiz data.
- (Stretch) Payment Information Route – The payment information route will only POST for the so that the site can store and charge the user for their rentals.

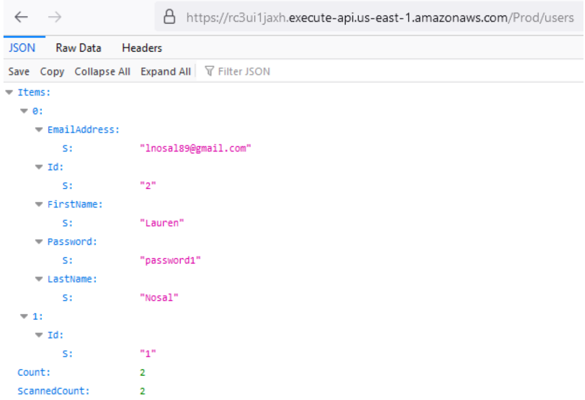
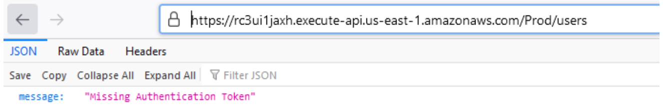
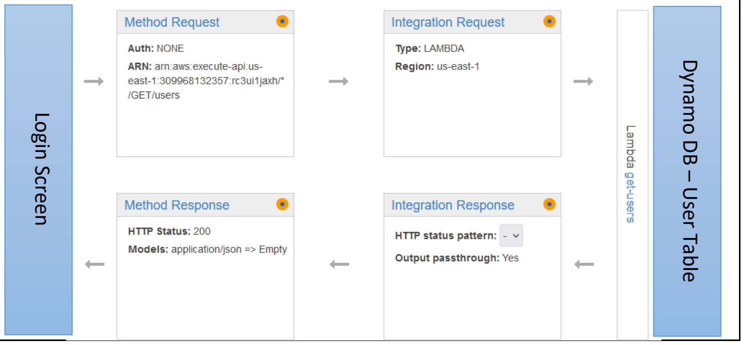
## User Route

1. Post User

| | |
|---|---|
| **Method** | POST |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users |
| **Purpose** | The purpose of this method is to create a new user when the user enters their information on the quiz screen |
| **Example Requests** | curl –request POST –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users |
| **Success Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>0:<br>EmailAddress:<br>S: "lnosal89@gmail.com"<br>Id:<br>S: "2"<br>FirstName:<br>S: "Lauren"<br>Password:<br>S: "password1"<br>LastName:<br>S: "Nosal"<br>1:<br>Id:<br>S: "1"<br>Count: 2<br>ScannedCount: 2 |
| **Error Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Quiz Answers Page<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-users<br>Dynamo DB – User Table<br>**Auth:** NONE<br>**ARN:** arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/users<br>**Type:** LAMBDA<br>**Region:** us-east-1<br>**HTTP Status:** 200<br>**Models:** application/json => Empty<br>**HTTP status pattern:** -<br>**Output passthrough:** Yes |

2. (Stretch) Get User by Email

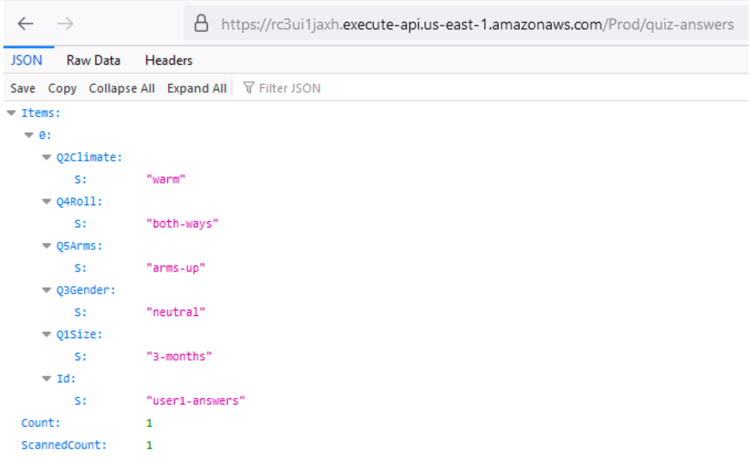
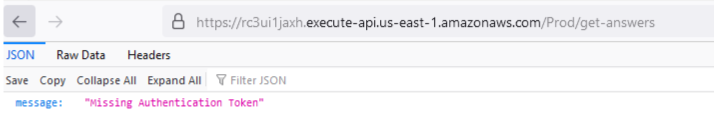
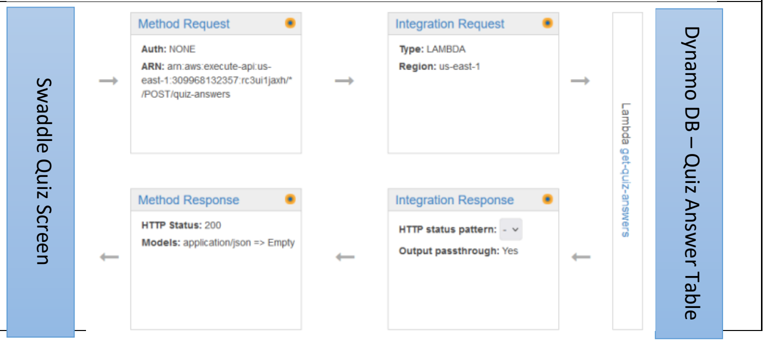
| | |
|---|---|
| **Method** | GET |

| | |
|---|---|
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users/id |
| **Purpose** | This will be used when a user logs in to the interface, it will get their UserID based on the email address that is input into the login screen. |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users/id |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>0:<br>EmailAddress:<br>S: "lnosal89@gmail.com"<br>Id:<br>S: "2"<br>FirstName:<br>S: "Lauren"<br>Password:<br>S: "password1"<br>LastName:<br>S: "Nosal"<br>1:<br>Id:<br>S: "1"<br>Count: 2<br>ScannedCount: 2 |
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/users<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Login Screen<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-users<br>Dynamo DB – User Table<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/users<br>Type: LAMBDA<br>Region: us-east-1<br>HTTP Status: 200<br>Models: application/json => Empty<br>HTTP status pattern: -<br>Output passthrough: Yes |

## Quiz Answer Route

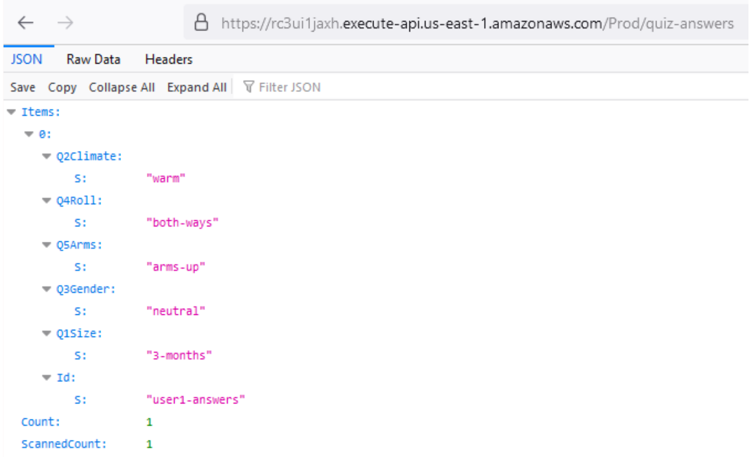
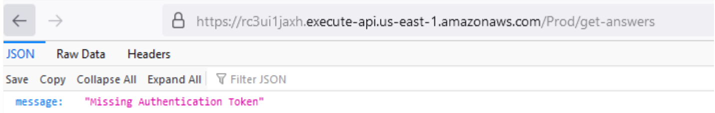
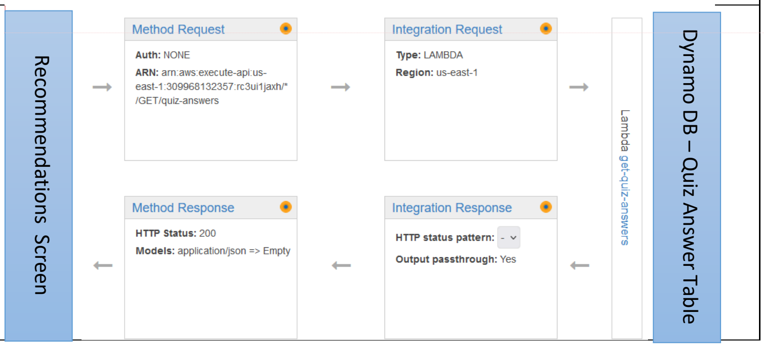
1. Post Quiz Answer

| | |
|---|---|
| **Method** | POST |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers |

| | |
|---|---|
| **Purpose** | This will be used to update the database with the quiz answer results from the quiz page. |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers |
| **Success Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers<br>JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>Items:<br>0:<br>Q2Climate:<br>S: "warm"<br>Q4Roll:<br>S: "both-ways"<br>Q5Arms:<br>S: "arms-up"<br>Q3Gender:<br>S: "neutral"<br>Q1Size:<br>S: "3-months"<br>Id:<br>S: "user1-answers"<br>Count: 1<br>ScannedCount: 1 |
| **Error Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/get-answers<br>JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Swaddle Quiz Screen<br>Method Request<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/POST/quiz-answers<br>Integration Request<br>Type: LAMBDA<br>Region: us-east-1<br>Lambda get-quiz-answers<br>Dynamo DB – Quiz Answer Table<br>Method Response<br>HTTP Status: 200<br>Models: application/json => Empty<br>Integration Response<br>HTTP status pattern: -<br>Output passthrough: Yes |

2. Get Quiz Answers

| | |
|---|---|
| **Method** | GET |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers |

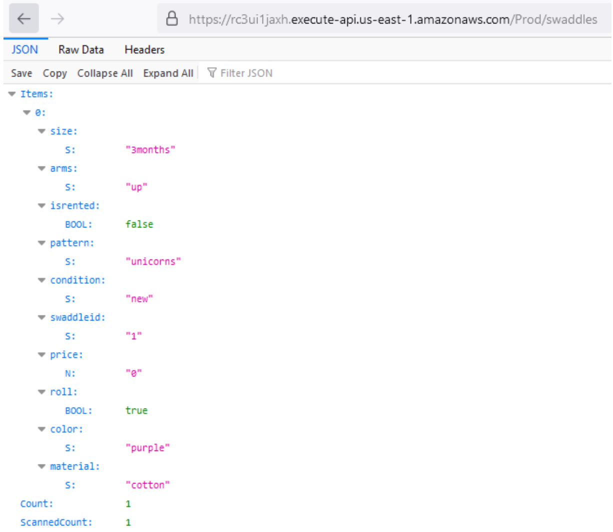
<table>
<tr><td><b>Purpose</b></td><td>This will be used to get the answers to the quiz and will power the top 5 swaddle recommendations on the reccomedations page for the MVP and the rental cart for the stretch.</td></tr>
<tr><td><b>Example Requests</b></td><td>curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers</td></tr>
<tr><td><b>Success Message</b></td><td><br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/quiz-answers<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>  0:<br>    Q2Climate:<br>      S: "warm"<br>    Q4Roll:<br>      S: "both-ways"<br>    Q5Arms:<br>      S: "arms-up"<br>    Q3Gender:<br>      S: "neutral"<br>    Q1Size:<br>      S: "3-months"<br>    Id:<br>      S: "user1-answers"<br>Count: 1<br>ScannedCount: 1</td></tr>
<tr><td><b>Error Message</b></td><td><br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/get-answers<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token"</td></tr>
<tr><td><b>Diagram</b></td><td><br> </td></tr>
</table>

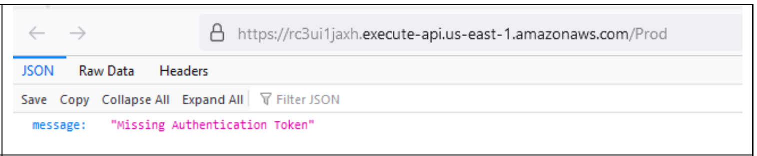
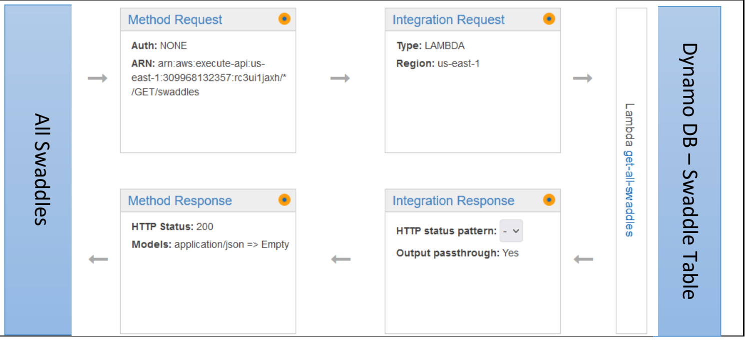
Commented [NL1]:

Swaddle Route

1. (Stretch) Get All Swaddles

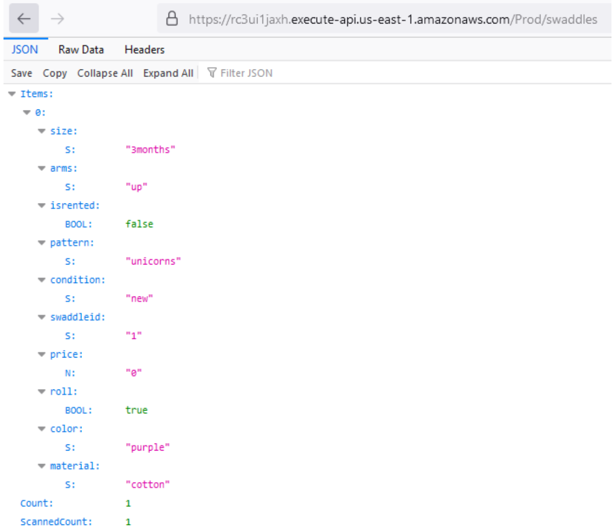
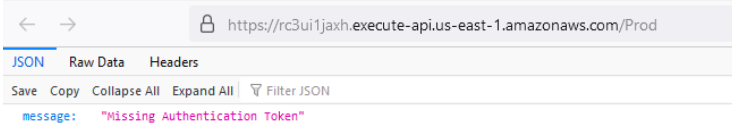
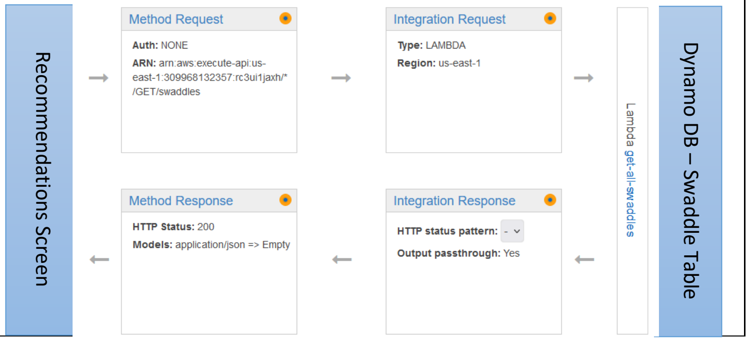
| | |
|---|---|
| **Method** | GET |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |

| | |
|---|---|
| **Purpose** | This will be used to get all the swaddles returned for a future page that has all the swaddles listed on it. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "Swaddles"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON   Raw Data   Headers<br>Save   Copy   Collapse All   Expand All   Filter JSON<br>Items:<br>  0:<br>    size:<br>      S:     "3months"<br>    arms:<br>      S:     "up"<br>    isrented:<br>      BOOL:     false<br>    pattern:<br>      S:     "unicorns"<br>    condition:<br>      S:     "new"<br>    swaddleid:<br>      S:     "1"<br>    price:<br>      N:     "0"<br>    roll:<br>      BOOL:     true<br>    color:<br>      S:     "purple"<br>    material:<br>      S:     "cotton"<br>Count:     1<br>ScannedCount:     1 |

| | |
|---|---|
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br>Method Request<br>**Auth:** NONE<br>**ARN:** arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>**Type:** LAMBDA<br>**Region:** us-east-1<br>Method Response<br>**HTTP Status:** 200<br>**Models:** application/json => Empty<br>Integration Response<br>**HTTP status pattern:** -<br>**Output passthrough:** Yes<br> All Swaddles<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddle Table |

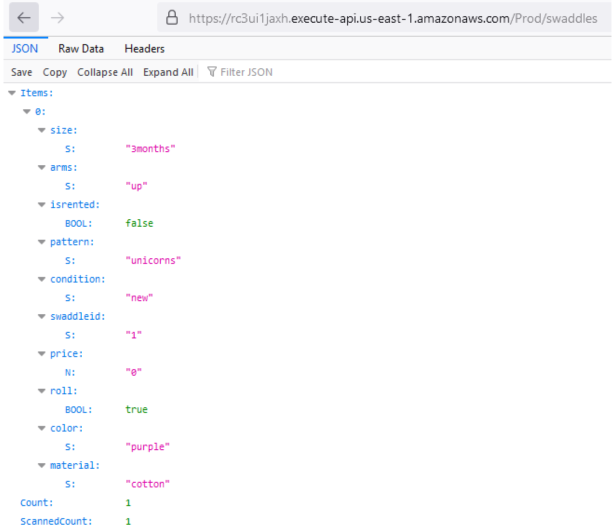
2. Get Swaddles by Attribute

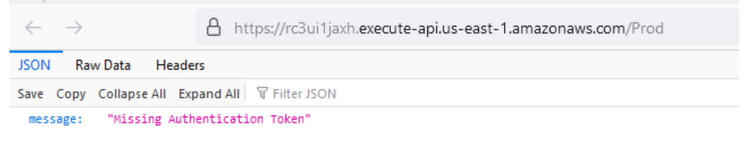
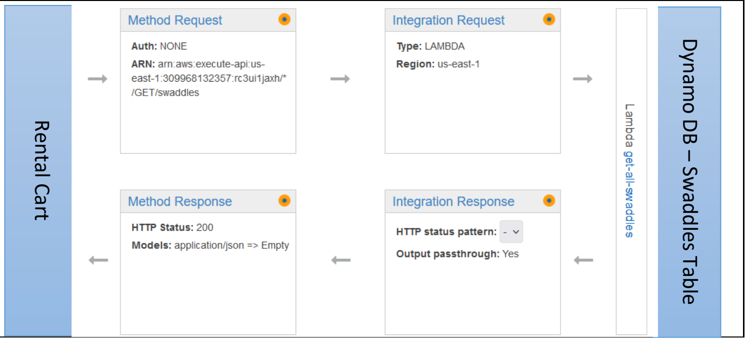
| | |
|---|---|
| **Method** | GET |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |
| **Purpose** | This will be used to return swaddles with attributes that match the quiz answers which will be exposed in the rental cart. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>            Key {<br>             “arms”: “up”<br>             “size”: “3months”<br>}<br>    TableName: "Swaddle"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  }); |

| | |
|---|---|
| | }; |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>0:<br>size:<br>S: "3months"<br>arms:<br>S: "up"<br>isrented:<br>BOOL: false<br>pattern:<br>S: "unicorns"<br>condition:<br>S: "new"<br>swaddleid:<br>S: "1"<br>price:<br>N: "0"<br>roll:<br>BOOL: true<br>color:<br>S: "purple"<br>material:<br>S: "cotton"<br>Count: 1<br>ScannedCount: 1 |
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br>Recommendations Screen<br>Method Request<br>**Auth:** NONE<br>**ARN:** arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>**Type:** LAMBDA<br>**Region:** us-east-1<br>Method Response<br>**HTTP Status:** 200<br>**Models:** application/json => Empty<br>Integration Response<br>**HTTP status pattern:** -<br>**Output passthrough:** Yes<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddle Table |

3. (Stretch) Update Swaddle Status

| | |
|---|---|
| **Method** | PUT |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |

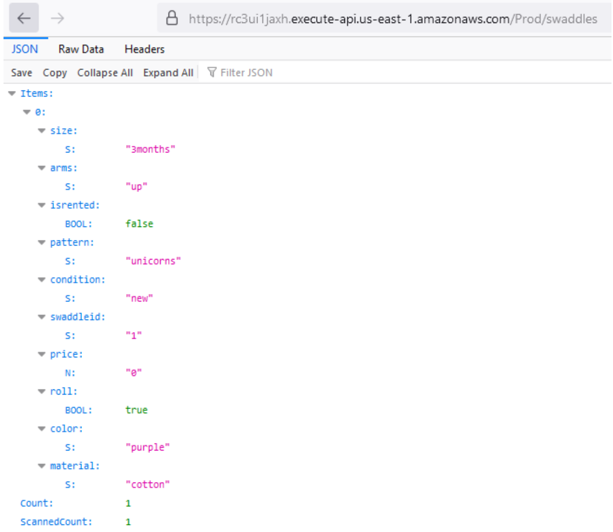
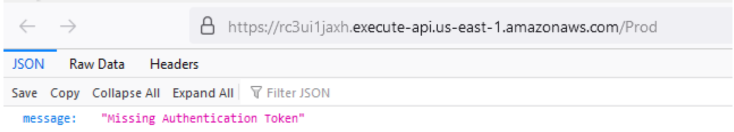
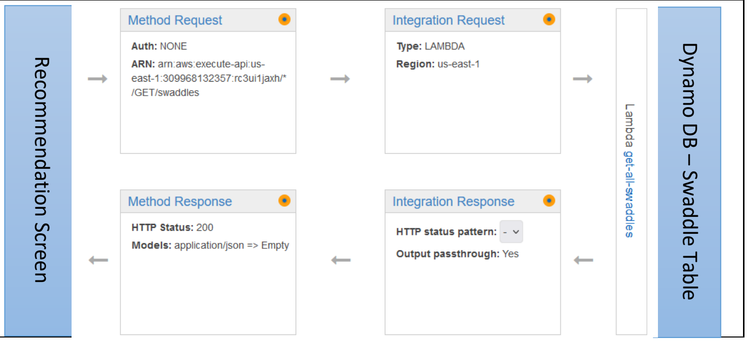
| | |
|---|---|
| **Purpose** | This will be used to update the swaddle status when the rental cart is updated. It will also be used for the stretch goal of updating the rental cart for return and buy. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "Swaddles"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |
| **Example Requests** | curl –request PUT –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>  0:<br>    size:<br>      S: "3months"<br>    arms:<br>      S: "up"<br>    isrented:<br>      BOOL: false<br>    pattern:<br>      S: "unicorns"<br>    condition:<br>      S: "new"<br>    swaddleid:<br>      S: "1"<br>    price:<br>      N: "0"<br>    roll:<br>      BOOL: true<br>    color:<br>      S: "purple"<br>    material:<br>      S: "cotton"<br>Count: 1<br>ScannedCount: 1 |

| | |
|---|---|
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** |  Rental Cart<br>Method Request<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>Type: LAMBDA<br>Region: us-east-1<br>Method Response<br>HTTP Status: 200<br>Models: application/json => Empty<br>Integration Response<br>HTTP status pattern: -<br>Output passthrough: Yes<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddles Table |

Recommendation Route

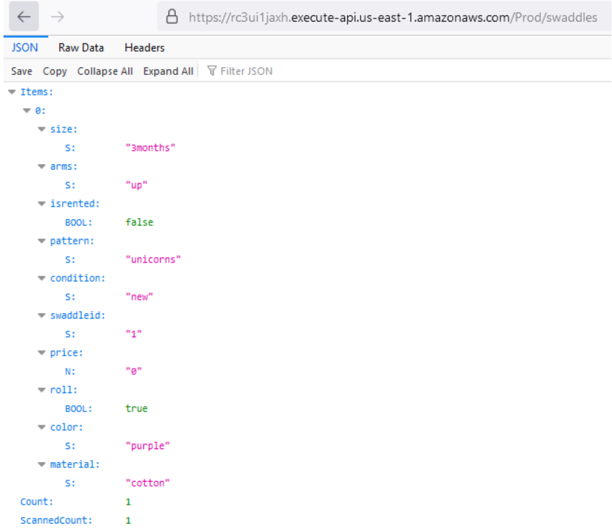
1. Get Recommendation

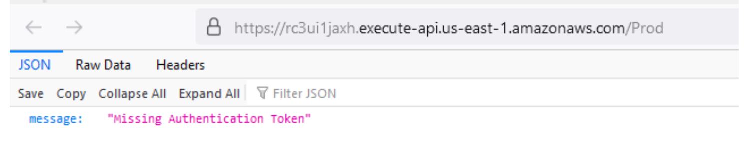
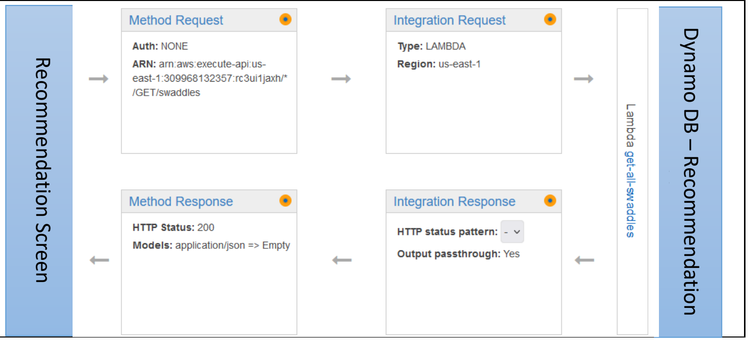
| | |
|---|---|
| **Method** | GET |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/recommendation |
| **Purpose** | This will be used to get the data in the recommendations table for the recommended swaddles screen. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "recommendation"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |

| | |
|---|---|
| | |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/recommendation |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>▾ Items:<br>▾ 0:<br>▾ size:<br>S: "3months"<br>▾ arms:<br>S: "up"<br>▾ isrented:<br>BOOL: false<br>▾ pattern:<br>S: "unicorns"<br>▾ condition:<br>S: "new"<br>▾ swaddleid:<br>S: "1"<br>▾ price:<br>N: "0"<br>▾ roll:<br>BOOL: true<br>▾ color:<br>S: "purple"<br>▾ material:<br>S: "cotton"<br>Count: 1<br>ScannedCount: 1 |
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Recommendation Screen<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddle Table<br>**Auth:** NONE<br>**ARN:** arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>**Type:** LAMBDA<br>**Region:** us-east-1<br>**HTTP Status:** 200<br>**Models:** application/json => Empty<br>**HTTP status pattern:** -<br>**Output passthrough:** Yes |

2. Post Recommendation

| | |
|---|---|
| **Method** | POST |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/recommendation |

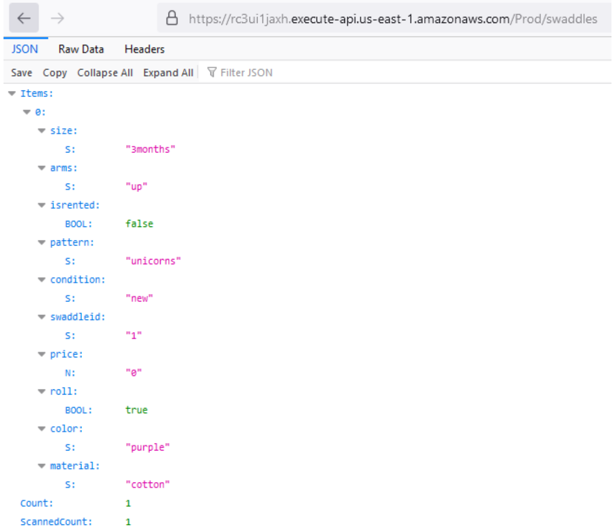
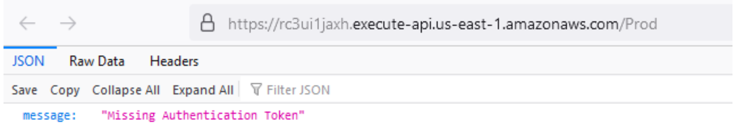
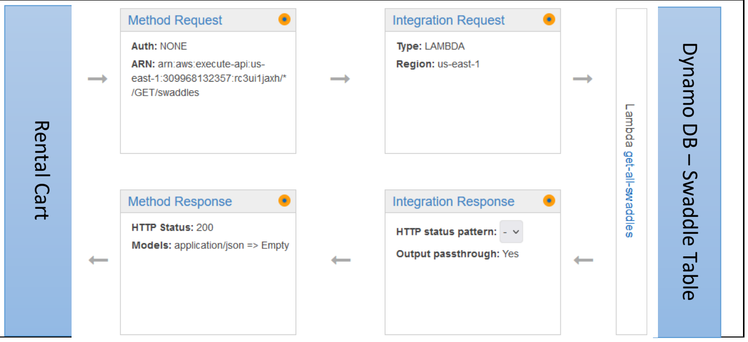
| | |
|---|---|
| **Purpose** | This will be used to create the recommendation table based on the quiz answers. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "recommendation"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |
| **Example Requests** | curl –request POST –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/recommendation |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>  0:<br>    size:<br>      S: "3months"<br>    arms:<br>      S: "up"<br>    isrented:<br>      BOOL: false<br>    pattern:<br>      S: "unicorns"<br>    condition:<br>      S: "new"<br>    swaddleid:<br>      S: "1"<br>    price:<br>      N: "0"<br>    roll:<br>      BOOL: true<br>    color:<br>      S: "purple"<br>    material:<br>      S: "cotton"<br>Count: 1<br>ScannedCount: 1 |

| | |
|---|---|
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br> JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Recommendation Screen<br>Method Request<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>Type: LAMBDA<br>Region: us-east-1<br>Lambda get-all-swaddles<br>Dynamo DB – Recommendation<br>Method Response<br>HTTP Status: 200<br>Models: application/json => Empty<br>Integration Response<br>HTTP status pattern: -<br>Output passthrough: Yes |

(Stretch) Rental Cart Route

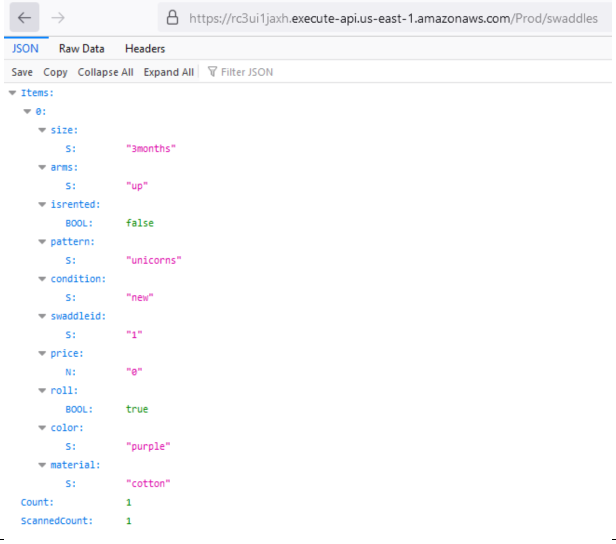
3. Get Rental Cart

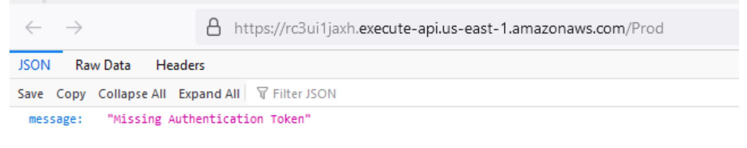
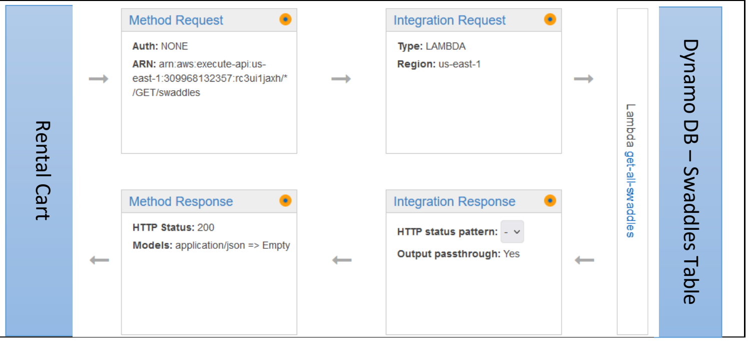
| | |
|---|---|
| **Method** | GET |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart |
| **Purpose** | This will be used to get the data in the rental cart for the cart page. For the stretch assignment, this will be used to get the cart data to power the return and buy screen. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "rentalcart"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |

<table>
<tr><td></td><td></td></tr>
<tr><td><b>Example Requests</b></td><td>curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart</td></tr>
<tr><td><b>Success Message</b></td><td>
https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles
JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON
<pre>
▾ Items:
  ▾ 0:
    ▾ size:
        S:        "3months"
    ▾ arms:
        S:        "up"
    ▾ isrented:
        BOOL:     false
    ▾ pattern:
        S:        "unicorns"
    ▾ condition:
        S:        "new"
    ▾ swaddleid:
        S:        "1"
    ▾ price:
        N:        "0"
    ▾ roll:
        BOOL:     true
    ▾ color:
        S:        "purple"
    ▾ material:
        S:        "cotton"
  Count:          1
  ScannedCount:   1
</pre></td></tr>
<tr><td><b>Error Message</b></td><td>
https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod
JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON
<pre>
message:   "Missing Authentication Token"
</pre></td></tr>
<tr><td><b>Diagram</b></td><td>
Rental Cart<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddle Table
Method Request<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>Type: LAMBDA<br>Region: us-east-1<br>Method Response<br>HTTP Status: 200<br>Models: application/json => Empty<br>Integration Response<br>HTTP status pattern: -<br>Output passthrough: Yes</td></tr>
</table>

4. Post Rental Cart

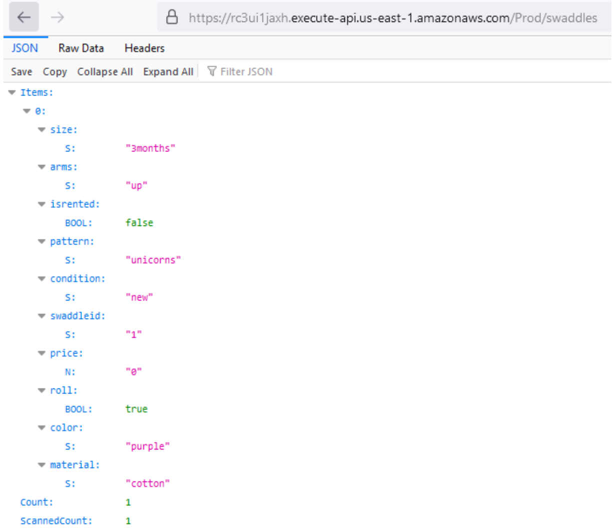
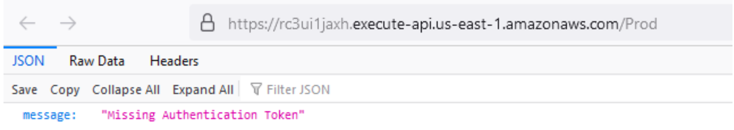
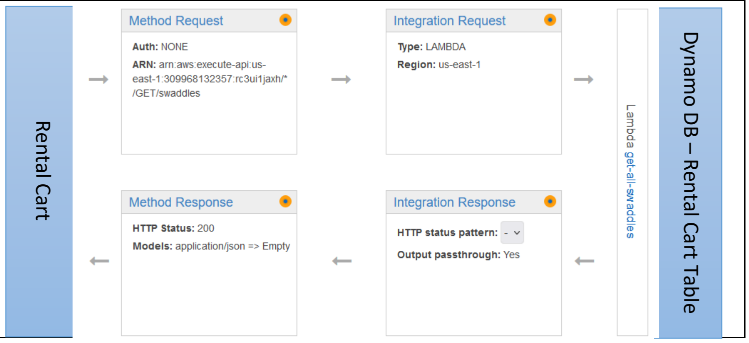
| | |
|---|---|
| **Method** | POST |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart |

| | |
|---|---|
| **Purpose** | This will be used to create the rental cart. The swaddles will be powered by the GET from the swaddle by attribute method. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "rentalcart"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |
| **Example Requests** | curl –request POST –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart |
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br> JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>Items:<br>  0:<br>    size:<br>      S:   "3months"<br>    arms:<br>      S:   "up"<br>    isrented:<br>      BOOL:   false<br>    pattern:<br>      S:   "unicorns"<br>    condition:<br>      S:   "new"<br>    swaddleid:<br>      S:   "1"<br>    price:<br>      N:   "0"<br>    roll:<br>      BOOL:   true<br>    color:<br>      S:   "purple"<br>    material:<br>      S:   "cotton"<br>Count:   1<br>ScannedCount:   1 |

| | |
|---|---|
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Rental Cart<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-all-swaddles<br>Dynamo DB – Swaddles Table<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Type: LAMBDA<br>Region: us-east-1<br>HTTP Status: 200<br>Models: application/json => Empty<br>HTTP status pattern: -<br>Output passthrough: Yes |

5. Update Rental Cart

| | |
|---|---|
| **Method** | PUT |
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart |
| **Purpose** | This will be used to update the rental cart based on the selections by the user on whether or not they want to rent the swaddles that are presented to them. For the stretch assignment, this will be used to update the rental cart table. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "rentalcart"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |

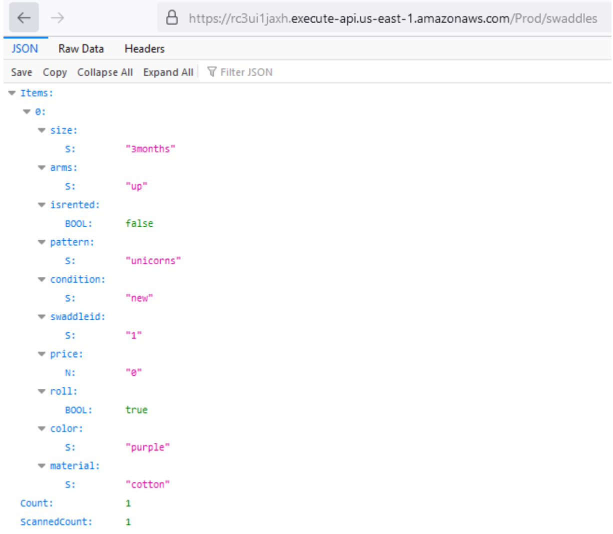
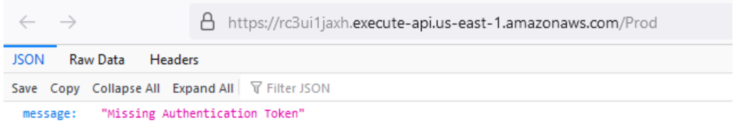
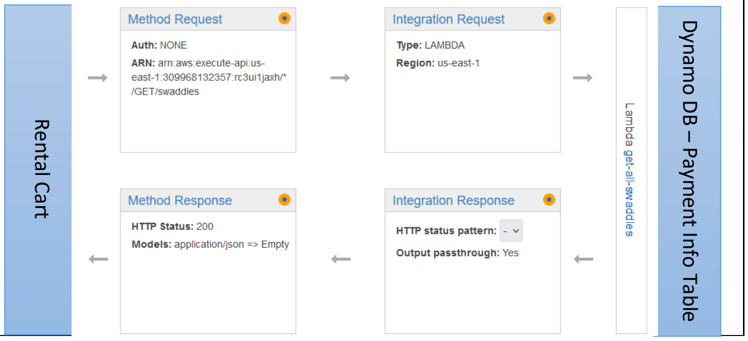
| | |
|---|---|
| **Example Requests** | curl –request PUT –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/rentalcart |
| **Success Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>Items:<br>0:<br>size:<br>S: "3months"<br>arms:<br>S: "up"<br>isrented:<br>BOOL: false<br>pattern:<br>S: "unicorns"<br>condition:<br>S: "new"<br>swaddleid:<br>S: "1"<br>price:<br>N: "0"<br>roll:<br>BOOL: true<br>color:<br>S: "purple"<br>material:<br>S: "cotton"<br>Count: 1<br>ScannedCount: 1 |
| **Error Message** | <br> https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON<br>Raw Data<br>Headers<br>Save<br>Copy<br>Collapse All<br>Expand All<br>Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Rental Cart<br>Method Request<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Integration Request<br>Type: LAMBDA<br>Region: us-east-1<br>Lambda get-all-swaddles<br>Dynamo DB – Rental Cart Table<br>Method Response<br>HTTP Status: 200<br>Models: application/json => Empty<br>Integration Response<br>HTTP status pattern: -<br>Output passthrough: Yes |

(Stretch) Payment Information Route

1. Post Payment Information

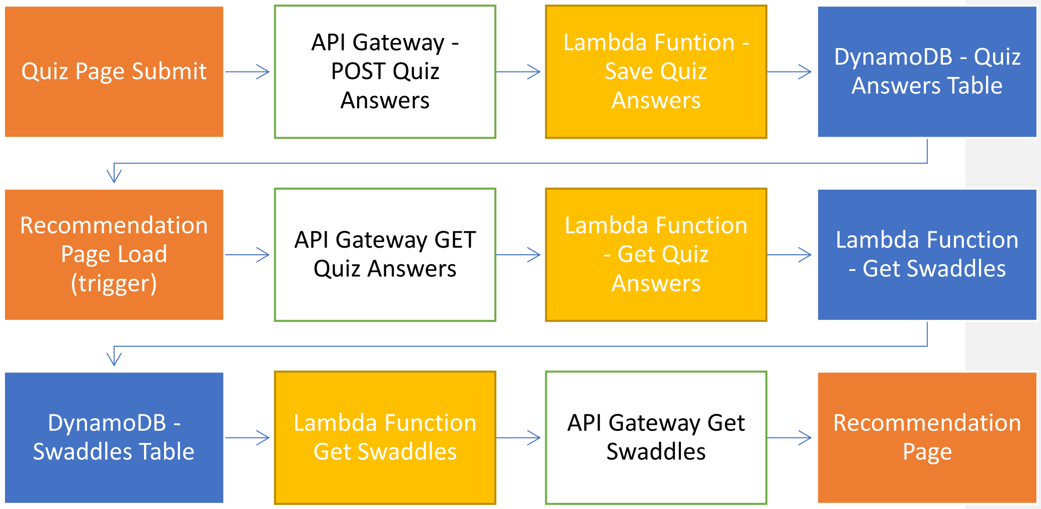
| | |
|---|---|
| **Method** | POST |

| | |
|---|---|
| **API URL** | https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/paymentinformation |
| **Purpose** | This will be used to post data to the payment information table so that the website can process the payment information for the MVP. For the stretch assignment, this will be used to get the payment information data for the return and buy screen. |
| **Lambda Code** | const AWS = require("aws-sdk");<br>const dynamodb = new AWS.DynamoDB({<br>  region: "us-east-1",<br>  apiVersion: "2022-03-16"<br>});<br><br>exports.handler = (event, context, callback) => {<br>  const params = {<br>    TableName: "paymentinformation"<br>  };<br>  dynamodb.scan(params, (err, data) => {<br>    if (err) {<br>      console.log(err);<br>      callback(err);<br>    } else {<br>      callback(null, data);<br>    }<br>  });<br>}; |
| **Example Requests** | curl –request GET –url https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/paymentinformation |

| | |
|---|---|
| **Success Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod/swaddles<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>Items:<br>0:<br>size:<br>S: "3months"<br>arms:<br>S: "up"<br>isrented:<br>BOOL: false<br>pattern:<br>S: "unicorns"<br>condition:<br>S: "new"<br>swaddleid:<br>S: "1"<br>price:<br>N: "0"<br>roll:<br>BOOL: true<br>color:<br>S: "purple"<br>material:<br>S: "cotton"<br>Count: 1<br>ScannedCount: 1 |
| **Error Message** | <br>https://rc3ui1jaxh.execute-api.us-east-1.amazonaws.com/Prod<br>JSON Raw Data Headers<br>Save Copy Collapse All Expand All Filter JSON<br>message: "Missing Authentication Token" |
| **Diagram** | <br> Rental Cart<br>Method Request<br>Integration Request<br>Method Response<br>Integration Response<br>Lambda get-all-swaddles<br>Dynamo DB – Payment Info Table<br>Auth: NONE<br>ARN: arn:aws:execute-api:us-east-1:309968132357:rc3ui1jaxh/*/GET/swaddles<br>Type: LAMBDA<br>Region: us-east-1<br>HTTP Status: 200<br>Models: application/json => Empty<br>HTTP status pattern: -<br>Output passthrough: Yes |

## Swaddle Recommendation Engine

One of the items that will differentiate this site from others is the swaddle recommendation engine. This recommendation engine is powered by the quiz answers. Based on the answers selected by the user, the site will recommend the swaddles that will work for the baby. Below is the overview of the architecture for this engine:


Quiz Page Submit
API Gateway - POST Quiz Answers
Lambda Funtion - Save Quiz Answers
DynamoDB - Quiz Answers Table
Recommendation Page Load (trigger)
API Gateway GET Quiz Answers
Lambda Function - Get Quiz Answers
Lambda Function - Get Swaddles
DynamoDB - Swaddles Table
Lambda Function Get Swaddles
API Gateway Get Swaddles
Recommendation Page