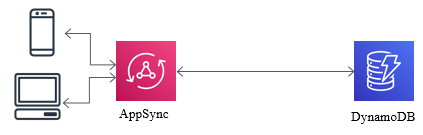
# Product catalog reference function

## 1.1 Functional image



## 1.2 Functional overview

### 1.2.1 Product catalog reference function

Product information is collected using the product ID selected by the user as the primary key.

Product Information：Product ID、Product name、Cost、Description、Picture

＜INPUT（JSON format）＞

　　・ProductID

＜OUTPUT（JSON format）＞

　　・StatusCode

　　・body

　　　・Item（The result of referencing the product table）

　　　　・Productname

　　 ・Picture

　　 ・Description

　　 ・Cost

## 1.3 API request path

|  |  |  |
| --- | --- | --- |
| API resource path | HTTP method | Supported Content Types |
| /jgs2020ia01/product | GET | application/json |

## 1.4 Configuration

### 1.4.1 DynamoDB Parameter Sheet

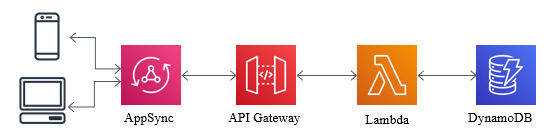
| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | ProductTable |  |
| Primary partition key |  | Productid（Int） |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDB Streams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | ProductTable-ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | ProductTable-WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

### AppSync Parameter Sheet

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| GraphQL API name |  |  | JGS2020IA01 |  |
| Schema  Schema | schema |  | query: Query | Query |
| Type | Query |  |  |
| Function | Get product information |  |
| Field name | getProduct |  |
| Request | productid: Int |  |
| Response | Product |  |
| Type | Type name | Product |  |
| Field | productid: Int! | Primary key |
| productname: String |  |
| picture: String |  |
| description: String |  |
| cost: Int |  |
| Resolvers | Query | getProduct |  |
| Data source name | ProductTable |  |
| Request mapping template |  |  |
| Operation | GetItem |  |
| Key | "productid": $util.dynamodb.toDynamoDBJson($ctx.args.productid) |  |
| Response mapping template | $util.toJson($ctx.result) | Returns a “stringified” JSON representation of that object. |
| Data sources | Name |  | ProductTable | Query: Data source for getProduct. |
| Type |  | AMAZON\_DYNAMODB |  |
| Region |  | AP-NORTHEAST-1 | Tokyo |
| Table name |  | ProductTable |  |
| Create or use an existing roll |  | N/A | Automatically granted at creation |

# Point reference function

## Functional image



## Functional overview

### Point reference function

Possession points is obtained using the user ID selected by the user as the primary key.

＜INPUT（JSON format）＞

　userid

＜OUTPUT（JSON format）＞

　userid,point

## API request path

|  |  |  |
| --- | --- | --- |
| API resource path | HTTP method | Supported Content Types |
| /jgs2020ia01/point | GET | application/json |

## Configuration

### DynamoDB Parameter Sheet

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | PointTable |  |
| Primary partition key |  | userid |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDBStreams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | PointTable-ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | PointTable-WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

### Lambda Parameter Sheet

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Basic settings | Function name |  | pointmaster |  |
| Runtime |  | Python 3.8 |  |
| Handler |  | lambda\_function.lambda\_handler |  |
| Memory |  | 128 MB | Default |
| Timeout |  | 0 min 3 sec | Default |
| Execution role | Role name | LambdaAccess2DynamoDB |  |
| Policy name | AmazonDynamoDBFullAccess |  |
| AWSLambdaDynamoDBExecutionRole |  |
| Function code |  |  | - | Manage on github |
| Environment variables |  |  | 【Not use】 |  |
| Tags |  |  | 【Not use】 |  |
| AWS X-Ray tracing |  |  | 【Not Active】 |  |
| Logging | Cloud Watch Logs |  | /aws/Lambda/pointmaster | Automatically set up |

### API Gateway Parameter Sheet

Endpoint： <https://NNNNNNNNNN.execute-api.ap-northeast-1.amazonaws.com/jgs2020ia01>

※NNNNNNNNNN will include the IAM user's account ID, so they are masked.

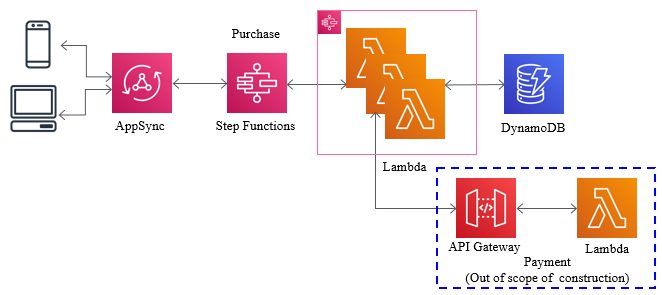
| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| API name |  |  | PointManageAPI |  |
| Protocol |  |  | REST |  |
| Endpoint type |  |  | Regional |  |
| Stages |  |  | jgs2020ia01 |  |
| API Configuration |  |  | /jgs2020ia01  　Resources：/point  　　　Method：GET |  |
| Resources | Name |  | point |  |
| Path |  | /point |  |
| Methods | Method |  | GET |  |
| Method Request | Configure the acceptance settings of APIGateway, such as whether to accept authentication and query parameters. | | |
| Authorization | **NONE**　or　AWS\_IAM |  |
| Request Validator | NONE |  |
| API Key Required | false | true or false |
| URL Query String Parametes | UserId |  |
| HTTP Request Headers | 【No headers】 |  |
| Integration Request | Configure upstream specifications and request body conversions. | | |
| Integration type | Lambda Function |  |
| Use Lambda Proxy integration | No check |  |
| Lambda Region | ap-northeast-1 | Tokyo |
| URL Path Parameters | No path parameters |  |
| URL Query String Parameters | No query strings |  |
| HTTP Headers | No Headers | Is there an additional HTTP request header? |
| Mapping Templates |  |  |
| Content-type | application/json |  |
| Code | "UserId":"$input.params('UserId')" | Mapping the accepted parameters |
| Method | Configure response settings to the client, such as response headers and response body settings for each status code. | | |
| HTTP Status | 200 |  |
| Integration Response | Configure integration settings such as status code mapping and response content conversion | | |
| Lambda Error Regex | - |  |
| Method response status | 200 |  |
| Content handing | Passthrough |  |
| Authorizers |  |  | 【Not Use】 | Using the Amazon Cognito user pool or Lambda functions to control access to the API |
| Documentation |  |  | 【Not use】 |  |

### AppSync Parameter Sheet

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| GraphQL API name |  |  | JGS2020IA01 |  |
| Schema | schema |  | query: Query |  |
| Type | Query |  |  |
| Function | Point reference |  |
| Field name | getPoint |  |
| Request | UserID: String |  |
| Response | Point |  |
| Type | Type name | Point | Query: Response to getPoint  Datasource: DynamoDB PointTable |
| Field | UserID: String! | Public key |
| Point: Int |  |
| Resolvers | Query | getPoint |  |
| Data source name | httppoint |  |
| Request mapping template |  |  |
| method | Get |  |
| params |  |  |
| headers | "Content-Type": "application/json" |  |
| ResourcePath | $util.toJson("/jgs2020ia01?UserId=${ctx.args.UserID}") |  |
| Response mapping template | $ctx.result.body |  |
| Data Source | Name |  | httppoint | Query: Data Sources for getPoint |
| Type |  | HTTP |  |
| Resource |  | https://NNNNNNNNN.execute-api.ap-northeast-1.amazonaws.com | Configuring APIGateway endpoints |

# Product purchase function

## 3.1 Functional image



## 3.2 Functional overview

### 3.2.1 Product purchase function

The purchase of the product selected by the user is processed. This process is carried out in the following flow.

1. Check inventory and reserve stock for the number of items purchased
2. Adding purchase information to the transaction table
3. Payment processing (external API)
4. Payment is complete
5. Secure the products
6. Record transaction data

＜INPUT（JSON format）＞

　　・TransactionID

　　・Amount

　　・PaymentID

　　・Price

　　・ProductID

　　・Status

　　・UserID

　　・date

＜OUTPUT（JSON format）＞

　　・StatusCode

　　・body

　　　・Item(Information in the transaction table after processing is complete)

　　　　・TransactionID

　　 ・Amount

　　 ・PaymentID

　　 ・Price

　　 ・ProductID

　　 ・Status

　　 ・UserID

　　 ・date

### 3.2.2 Endpoint

|  |  |
| --- | --- |
| Type | Path |
| HTTP Endpoint | https://states.ap-northeast-1.amazonaws.com/ |
| ARN | ARN:aws:iam::NNNNNNNNNN:role/service-role/StepFunctions-MyStateMachine-role-96754991 |

## 3.3 API Request Path

No settings.

## Configuration

### Dynamo DB Parameter

1. PointTable

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | PointTable |  |
| Primary partition key |  | UserID(String) |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDB Streams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | PointTable-ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | PointTable-WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

1. Product Table

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | ProductTable |  |
| Primary partition key |  | productid（Int） |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| TTL管理 | DynamoDB Streams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | ProductTable-ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | ProductTable-WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling  Read/Write capacity mode  ReadCapacity | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
|  | Provisioned |  |
|  | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

1. Stock Table

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | StockTable |  |
| Primary partition key |  | ProductID(String) |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDB Streams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | StockTable-ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | StockTable-WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

1. TransactionTable

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | TransactionTable |  |
| Primary partition key |  | TransactionID(String) |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDBStreams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | TransactionTable -ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | TransactionTable -WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

1. UserTable

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| Overview | Table name |  | UserTable |  |
| Primary partition key |  | UserID(String) |  |
| Primary sort key |  | 【Not use】 |  |
| Point-in-Time Recovery |  | ENABLE | Storage up to 35 days |
| Encryption Type |  | DEFAULT | Data encryption |
| Time to live attribute |  | DISABLED |  |
| DynamoDBStreams | No check |  |
| Region |  | AP Northeast(Tokyo) |  |
| Items |  |  | 【See the data definition document】 |  |
| Alarms | Edit Alarm |  | UserTable -ReadCapacityUnitsLimit-BasicAlarm | Automatically set up |
|  | UserTable -WriteCapacityUnitsLimit-BasicAlarm | Automatically set up |
| Capacity | Read/Write capacity mode |  | Provisioned |  |
| ReadCapacity |  | 10 |  |
| Strongly consistent reads |  | Use |  |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：Target Utilization | 80% |  |
| RCU：Minimum capacity | 5 |  |
| RCU：Max capacity | 20 |  |
| WCU：Target Utilization | 80% |  |
| WCU：Minimum capacity | 5 |  |
| WCU：Max capacity | 10 |  |
| Indexes | Global Secondary Indexes |  | 【Not use】 |  |
| Reserved capacity |  |  | 【Not use】 | Up to 76% off for 3 years  e.g. WCU 100, $180.00 in advance |
| Dynamo DB  Accelerator |  |  | 【Not use】 | Use the in-memory cache |
| Data storage location |  |  | 【Distributed across three AZ in ap-northeast-1】 |  |

### Lambda Parameter Sheet

1. Stock Confirm

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Function |  | StockConfirm |
| Runtime |  | Python 3.8 |
| Role | Execution role | Use an existing role |
|  | Role name  　 Policy name | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | - |
| Memory |  | 128 MB |
| Timeout |  | 0 min 3 sec |
| Concurrency for a Lambda function |  | 1000 |
| Function code |  | 【Manage on github】 |

1. SetTransaction Data

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Function |  | SetTransactionData |
| Runtime |  | Python 3.8 |
| Role | Execution role | Use an existing role |
|  | Role name  　 Policy name | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | - |
| Memory |  | 128 MB |
| Timeout |  | 0 min 3 sec |
| Concurrency for a Lambda function |  | 1000 |
| Function code |  | 【Manage on github】 |

1. SetPaymentID

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Function |  | SetPaymentID |
| Runtime |  | Python 3.8 |
| Role | Execution role | Use an existing role |
|  | Role name  　 Policy name | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | - |
| Memory |  | 128 MB |
| Timeout |  | 0 min 3 sec |
| Concurrency for a Lambda function |  | 1000 |
| Function code |  | 【Manage on github】 |

1. ReserveProduct

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Function |  | ReserveProduct |
| Runtime |  | Python 3.8 |
| Role | Execution role | Use an existing role |
|  | Role name  　 Policy name | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | - |
| Memory |  | 128 MB |
| Timeout |  | 0 min 3 sec |
| Concurrency for a Lambda function |  | 1000 |

1. UpdateTransaction

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Function |  | UpdateTransaction |
| Runtime |  | Node.js 12.x |
| Role | Execution role | Use an existing role |
|  | Role name  　 Policy name | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | - |
| Memory |  | 128 MB |
| Timeout |  | 0 min 3 sec |
| Concurrency for a Lambda function |  | 1000 |
| Runtime |  | Use an existing role |

### 3.4.3 Step Functions Parameter Sheet

|  |  |  |
| --- | --- | --- |
| Subject | Item | Setting value |
| Name |  | MyStateMachine |
| Definition |  | {  "Comment": "A simple AWS Step Functions state machine that automates a call center support session.",  "StartAt": "StockConfirm",  "States": {  "StockConfirm": {  "Type": "Task",  "Resource": "ARN:aws:lambda:ap-northeast-1:NNNNNNNNN:function:StockConfirm",  "Next":"SetTransaction"  },  "SetTransaction": {  "Type": "Task",  "Resource": "ARN:aws:lambda:ap-northeast-1:NNNNNNNNN:function:SetTransactionData",  "Next":"SetPaymentID"  },  "SetPaymentID": {  "Type": "Task",  "Resource": "ARN:aws:lambda:ap-northeast-1:NNNNNNNNN:function:SetPaymentID",  "Next":"ReserveProduct"  },  "ReserveProduct": {  "Type": "Task",  "Resource": "ARN:aws:lambda:ap-northeast-1:NNNNNNNNN:function:ReserveProduct",  "Next":"UpdateTransaction"  },  "UpdateTransaction": {  "Type": "Task",  "Resource": "ARN:aws:lambda:ap-northeast-1:NNNNNNNNN:function:UpdateTransaction",  "End":true  }  }  } |
| Role | Execution role | Use an existing role |
|  | Role name | StepFunctions-MyStateMachine-role-96754991  ・XRayAccessPolicy-b50e99ca-835f-43dc-8b34-db078a659e19  ・LambdaInvokeScopedAccessPolicy-d33bb3c2-8518-455b-8b8c-893135300d9a |
| Logging |  | ERROR |
| Taga |  | No tags |

※NNNNNNNNNN will include the IAM user's account ID, so they are masked.

### 3.4.4 AppSync Parameter Sheet

| Subject | Item | Sub item | Setting value | Note |
| --- | --- | --- | --- | --- |
| GraphQL API name |  |  | JGS2020IA01 |  |
| Schema  Schema | schema |  | mutation: Mutation |  |
| Type | Mutation |  |  |
| Function |  |  |
| Field name | addTransaction |  |
| Request | TransactionID: String! | Public key |
| Amount: Int |  |
| PaymentID: String | ID to be issued at the time of payment |
| Price: Int |  |
| ProductID: String |  |
| Status: String | Transaction Status |
| UserID: String |  |
| date: String |  |
| Count: Int |  |
| Response | Transaction |  |
| Type | Type name | Transaction |  |
| Field | TransactionID: String! | Public key |
| Amount: Int |  |
| PaymentID: String | ID to be issued at the time of payment |
| Price: Int |  |
| ProductID: String |  |
| Status: String | Transaction Status |
| UserID: String |  |
| date: String |  |
| Count: Int |  |
| Resolvers | Mutation | addTransaction |  |
| Data source name | StepFunctionHttpDataSource |  |
| Request mapping template |  |  |
| method | POST |  |
| resourcePath | / |  |
| params |  |  |
| headers | "content-type": "application/x-amz-json-1.0",  "x-amz-target":"AWSStepFunctions.StartExecution" |  |
| body  　　　stateMachineARN | "ARN:aws:states:ap-northeast-1:275739154947:stateMachine:MyStateMachine" |  |
| input | \"TransactionID\": \"$ctx.args.TransactionID\", \"Amount\":\"$ctx.args.Amount\",  \"PaymentID\":\"$ctx.args.PaymentID\",  \"Price\":\"$ctx.args.Price\",  \"ProductID\":\"$ctx.args.ProductID\",  \"Status\":\"$ctx.args.Status\",  \"UserID\":\"$ctx.args.UserID\",  \"date\":\"$ctx.args.date\",  \"Count\":\"$ctx.args.Count\" |  |
| Data Source | Name |  | StepFunctionHttpDataSource | Mutation:  Data Source for addTransaction |
| Type |  | HTTP endpoint |  |
| Resource |  | https://states.ap-northeast-1.amazonaws.com/ | Configuring Step Functions endpoint |