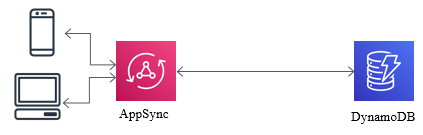
# 商品カタログ参照機能

## 機能イメージ



## 機能概要

### 商品カタログ参照機能

利用者が選択した商品IDをキーに商品情報を収集する。

商品情報：商品ID、商品名、金額、説明、画像

＜インプット（JSON形式）＞

　　・商品ID（ProductID）

＜アウトプット（JSON形式）＞

　　・ステータスコード（statusCode）

　　・body

　　　・Item（処理完了後の取引テーブルの情報）

　　　　・商品名（Productname）

　　 ・画像（Picture）

　　 ・説明（Description）

　　 ・金額（Cost）

## APIリクエストパス

|  |  |  |
| --- | --- | --- |
| APIリソースパス | HTTPメソッド | サポートされているコンテンツタイプ |
| /jgs2020ia01/product | GET | application/json |

## 設定

### DynamoDBパラメータシート

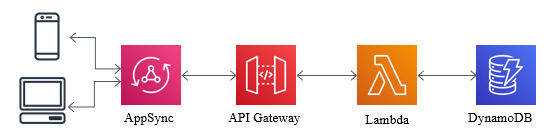
| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | ProductTable |  |
| プライマリ  パーティションキー |  | Productid（int） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | ProductTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | ProductTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

### AppSyncパラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| GraphQL API名 |  |  | JGS2020IA01 |  |
| Schema  Schema | schema |  | query: Query | Query |
| Type | Query |  |  |
| 機能 | 商品情報取得 |  |
| Field名 | getProduct |  |
| リクエスト | productid: Int | 商品ID |
| レスポンス | Product |  |
| Type | Type名 | Product |  |
| Field | productid: Int! | 商品ID　【PK】 |
| productname: String | 商品名 |
| picture: String | 商品画像 |
| description: String | 商品説明 |
| cost: Int | 金額 |
| Reslovers | Query | getProduct |  |
| データソース名 | ProductTable |  |
| リクエストマッピング |  |  |
| operation | GetItem |  |
| key | "productid": $util.dynamodb.toDynamoDBJson($ctx.args.productid) |  |
| レスポンスマッピング | $util.toJson($ctx.result) | オブジェクトの「文字列化された」JSON 表現を返す。 |
| Reslovers | Query | getPoint |  |
| データソース名 | httppoint |  |
| リクエストマッピング |  |  |
| method | Get |  |
| params |  |  |
| headers | "Content-Type": "application/json" |  |
| resourcePath | $util.toJson("/jgs2020ia01?UserId=${ctx.args.UserID}") |  |
| レスポンスマッピング | $ctx.result.body |  |
| Datasource | データソース名 |  | ProdauctTable | Query：getProduct用のデータソース |
| データソースタイプ |  | Amazon DynamoDBテーブル |  |
| リージョン |  | AP-NORTHEAST-1 | 東京リージョン |
| テーブル名 |  | ProductTable |  |
| 既存のロールを作成または適用 |  | N/A | 作成時に自動付与 |

# ポイント参照機能

## 機能イメージ



## 機能概要

### ポイント参照機能

利用者のユーザーIDをキーに保有ポイント数を取得する。

＜インプット＞

　userid

＜アウトプット＞

　userid,point

## APIリクエストパス

|  |  |  |
| --- | --- | --- |
| APIリソースパス | HTTPメソッド | サポートされているコンテンツタイプ |
| /jgs2020ia01/point | GET | application/json |

## 設定

### DynamoDBパラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | PointTable |  |
| プライマリ  パーティションキー |  | userid |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | PointTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | PointTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | グローバルセカンダリインデックス |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

### Lambdaパラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 基本設定 | 関数名 |  | pointmaster |  |
| ランタイム |  | Python 3.8 |  |
| ハンドラ |  | lambda\_function.lambda\_handler |  |
| メモリ |  | 128MB | 初期値 |
| タイムアウト値 |  | 3秒 | 初期値 |
| 実行ロール | ロール | LambdaAccess2DynamoDB |  |
| ポリシー | AmazonDynamoDBFullAccess |  |
| AWSLambdaDynamoDBExecutionRole |  |
| 関数 | コード |  | - | Githubで管理 |
| 環境変数 |  |  | なし |  |
| タグ |  |  | なし |  |
| X-RAY |  |  | 未設定 | 監視設定 |
| ログ出力 | Cloud Watch Logs |  | /aws/Lambda/pointmaster | 自動 |

### API Gateway パラメータシート

エンドポイント： <https://NNNNNNNNNN.execute-api.ap-northeast-1.amazonaws.com/jgs2020ia01>

※NNNNNNNNNNにはIAMユーザーのアカウントIDが記載されるため、伏字としている。

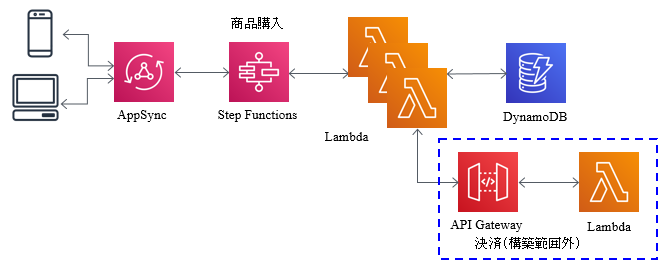
| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| API名 |  |  | PointManageAPI |  |
| 利用プロトコル |  |  | REST |  |
| エンドポイント |  |  | リージョン |  |
| ステージ名 | 公開用 |  | jgs2020ia01 |  |
| 構成要素 |  |  | /jgs2020ia01  　リソース：/point  　　　メソッド：GET |  |
| リソース | リソース名 |  | point |  |
| リソースパス |  | /point |  |
| メソッド |  |  | GET |  |
| 統合タイプ |  | Lambda関数 |  |
| Lambdaリージョン |  | ap-northeast-1 |  |
| Lambda関数 |  | pointmaster |  |
| メソッドのリクエスト | 認証やクエリパラメータ受付けるか等、APIGatewayの受付設定を行う。 | | |
| 認可 | **なし**　or　AWS\_IAM |  |
| リクエストの検証 | なし |  |
| APIキーの必要性 | false | true or false |
| URLクエリパラメータ文字列 | UserId |  |
| HTTPリクエストヘッダ | 追加なし |  |
| 統合リクエスト | アップストリーム指定、リクエストボディの変換設定などを行う。 | | |
| 統合タイプ | Lambda関数 |  |
| Lambdaプロキシ統合 | なし |  |
| Lambdaリージョン | ap-northeast-1 | 東京リージョン |
| URLパスパラメータ | なし | URIでドメインの後？の前まで |
| URLクエリ文字列パラメータ | なし | URIでドメインの後？後ろ |
| HTTPヘッダー | なし | HTTPリクエストヘッダ追加あるか |
| マッピングテンプレート |  |  |
| Content-type | application/json |  |
| 設定 | "UserId":"$input.params('UserId')" | 受付したパラメータマッピング |
| メソッドレスポンス | ステータスコードごとのレスポンスヘッダーやレスポンスボディ設定など、クライアントへのレスポンス設定を行う | | |
| HTTPステータス | 200 |  |
| 統合レスポンス | ステータスコードのマッピング、レスポンス内容の変換などの統合設定を行う | | |
| Lambdaエラーステータスの  正規表現 | default |  |
| メソッドレスポンスのステータス |  |  |
| コンテンツの処理 |  |  |
| オーソライザー |  |  | なし | Amazon Cognito ユーザープールまたは Lambda 関数を使用して API へのアクセスを制御 |
| ゲートウェイの  レスポンス |  |  |  |  |
| リソースポリシー |  |  | ＃設定　IPベース | プライベート API へのアクセスコントロールを設定 |
| ドキュメント |  |  | 利用なし |  |

### AppSyncパラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| GraphQL API名 |  |  | JGS2020IA01 |  |
| Schema | schema |  | query: Query |  |
| Type | Query |  |  |
| 機能 | ポイント参照 |  |
| Field名 | getPoint |  |
| リクエスト | UserID: String | ユーザーID |
| レスポンス | Point |  |
| 機能 | 商品情報取得 |  |
| Field名 | getProduct |  |
| リクエスト | productid: Int | 商品ID |
| レスポンス | Product |  |
| Type | Type名 | Point | Query：getPointのレスポンス  データソース：DynamoDB PointTable |
| Field | UserID: String! | ユーザーID　【PK】 |
| Point: Int | ポイント数 |
| Reslovers | Query | getPoint |  |
| データソース名 | httppoint |  |
| リクエストマッピング |  |  |
| method | Get |  |
| params |  |  |
| headers | "Content-Type": "application/json" |  |
| resourcePath | $util.toJson("/jgs2020ia01?UserId=${ctx.args.UserID}") |  |
| レスポンスマッピング | $ctx.result.body |  |
| Datasource | データソース名 |  | httppoint | Query：getPoint用のデータソース |
| データソースタイプ |  | HTTPエンドポイント |  |
| HTTPエンドポイント |  | https://NNNNNNNNN.execute-api.ap-northeast-1.amazonaws.com | APIGatewayエンドポイントを設定する |

# 商品購入機能

## 機能イメージ



## 機能概要

### 商品購入機能

利用者が選択した商品の購入処理を行う。本処理は以下の処理フローで処理を行う。

1. 在庫確認と購入数分の在庫予約
2. 取引テーブルへの購入情報追加
3. 決済処理（外部API）
4. 決済完了
5. 商品確保
6. 取引情報記録

＜インプット（JSON形式）＞

　　・取引ID（TransactionID）

　　・購入数（Amount）

　　・支払いID（PaymentID）

　　・価格（Price）

　　・商品ID（ProductID）

　　・ステータス（Status）

　　・ユーザーID（UserID）

　　・日付（date）

　　・購入数（Count）

＜アウトプット（JSON形式）＞

　　・ステータスコード（statusCode）

　　・body

　　　・Item（処理完了後の取引テーブルの情報）

　　　　・取引ID（TransactionID）

　　 ・購入数（Amount）

　　 ・支払いID（PaymentID）

　　 ・価格（Price）

　　 ・商品ID（ProductID）

　　 ・ステータス（Status）

　　 ・ユーザーID（UserID）

　　 ・日付（date）

### エンドポイント

|  |  |
| --- | --- |
| タイプ | パス |
| HTTPエンドポイント | https://states.ap-northeast-1.amazonaws.com/ |
| arn | arn:aws:iam::NNNNNNNNNN:role/service-role/StepFunctions-MyStateMachine-role-96754991 |

## APIリクエストパス

なし

## 設定

### Dynamo DBパラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | PointTable |  |
| プライマリ  パーティションキー |  | UserID（文字列） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | PointTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | PointTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | ProductTable |  |
| プライマリ  パーティションキー |  | productid（int） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | PointTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | PointTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | ProductTable |  |
| プライマリ  パーティションキー |  | productid（int） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | ProductTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | ProductTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | StockTable |  |
| プライマリ  パーティションキー |  | ProductID（文字列） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | StockTable-ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | StockTable-WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | TransactionTable |  |
| プライマリ  パーティションキー |  | TransactionID（文字列） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | TransactionTable -ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | TransactionTable -WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| 概要 | テーブル名 |  | UserTable |  |
| プライマリ  パーティションキー |  | UserID（文字列） |  |
| プライマリソートキー |  | 【None】 |  |
| ポイントインタイムリカバリー |  | 有効 | 最大35日間保管 |
| 暗号化タイプ |  | AWS Key Management Service | データ暗号化 |
| TTL管理 | TTL有効化 | なし |  |
| TTL管理 | DynamoDBストリーム | なし |  |
| リージョン |  | AP Northeast(東京) |  |
| 項目 |  |  | #　テーブル定義書 参照 |  |
| ログ管理 | アラーム作成 |  | UserTable -ReadCapacityUnitsLimit-BasicAlarm | 自動生成 |
|  | UserTable -WriteCapacityUnitsLimit-BasicAlarm | 自動生成 |
| スループット  キャパシティ | キャパシティモード |  | プロビジョンド |  |
| ReadCapacityUnit |  | 10 | 強力な結果整合性読み込み前提 |
| 強力な結果整合性読み込み |  | あり | クエリ発行側で指定 |
| WriteCapacityUnit |  | 5 |  |
| Auto Scaling | RCU：ターゲット使用率 | 80% |  |
| RCU：最小キャパシティ | 5 |  |
| RCU：最大キャパシティ | 20 |  |
| WCU：ターゲット使用率 | 80% |  |
| WCU：最小キャパシティ | 5 |  |
| WCU：最大キャパシティ | 10 |  |
| インデックス | ｸﾞﾛｰﾊﾞﾙｾｶﾝﾀﾞﾘｲﾝﾃﾞｯｸｽ |  | なし |  |
| リザーブド  キャパシティ |  |  | なし | 3年間利用時　最大76%OFF  例：WCU 100、前払いで$180.00 |
| Dynamo DB  Acceletaor |  |  | なし | インメモリキャッシュ利用 |
| データ保管先 |  |  | ap-northeast-1内の3つのAZに分散して格納 |  |

### Lambdaパラメータシート

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| Function |  | StockConfirm |
| Runtime |  | Python 3.8 |
| Role | アクセス許可 | 既存のロールを使用する |
|  | ロール名 | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | 空欄 |
| メモリー（MB） |  | 128 |
| タイムアウト |  | 3秒 |
| 同時実行数 |  | 1000 |
| 関数コード |  | Githubで管理 |

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| Function |  | SetTransactionData |
| Runtime |  | Python 3.8 |
| Role | アクセス許可 | 既存のロールを使用する |
|  | ロール名  　ポリシー名 | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | 空欄 |
| メモリー（MB） |  | 128 |
| タイムアウト |  | 3秒 |
| 同時実行数 |  | 1000 |
| 関数コード |  | Githubで管理 |

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| Function |  | SetPaymentID |
| Runtime |  | Python 3.8 |
| Role | アクセス許可 | 既存のロールを使用する |
|  | ロール名 | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | 空欄 |
| メモリー（MB） |  | 128 |
| タイムアウト |  | 3秒 |
| 同時実行数 |  | 1000 |
| 関数コード |  | Githubで管理 |

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| Function |  | ReserveProduct |
| Runtime |  | Python 3.8 |
| Role | アクセス許可 | 既存のロールを使用する |
|  | ロール名 | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
| メモリー（MB） |  | 128 |
| タイムアウト |  | 3秒 |
| 同時実行数 |  | 1000 |
| 関数コード |  | Githubで管理 |

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| Function |  | UpdateTransaction |
| Runtime |  | Node.js 12.x |
| Role | アクセス許可 | 既存のロールを使用する |
|  | ロール名 | LambdaAccess2DynamoDB  ・AmazonDyanamoDBFullAccess  ・AWSLambdaDynamoDBExecutionRole  ・AWSStepFunctionFullAccess |
|  | Policy template | 空欄 |
| メモリー（MB） |  | 128 |
| タイムアウト |  | 3秒 |
| 同時実行数 |  | 1000 |
| 関数コード |  | Githubで管理 |

### StepFunction パラメータシート

|  |  |  |
| --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 |
| 名前 |  | MyStateMachine |
| 定義 |  | {  "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  }  }  } |
| ロール | アクセス許可 | 既存のロールを選択する |
|  | ロール名 | StepFunctions-MyStateMachine-role-96754991  ・XRayAccessPolicy-b50e99ca-835f-43dc-8b34-db078a659e19  ・LambdaInvokeScopedAccessPolicy-d33bb3c2-8518-455b-8b8c-893135300d9a |
| ログ記録 |  | ERROR |
| タグ |  | 設定なし |

### AppSync パラメータシート

| 設定　大項目 | 設定　中項目 | 設定　小項目 | 設定　内容 | 備考 |
| --- | --- | --- | --- | --- |
| GraphQL API名 |  |  | JGS2020IA01 |  |
| Schema  Schema | schema |  | mutation: Mutation |  |
| Type | Mutation |  |  |
| 機能 |  |  |
| Field名 | addTransaction |  |
| リクエスト | TransactionID: String! | 取引ID　【PK】 |
| Amount: Int | 数量 |
| PaymentID: String | 電子決済時に発番されるID |
| Price: Int | 金額 |
| ProductID: String | 商品ID |
| Status: String | 取引ステータス |
| UserID: String | ユーザーID |
| date: String | 日付 |
| Count: Int | 回数 |
| レスポンス | Transaction |  |
| Type | Type名 | Transaction |  |
| Field | TransactionID: String! | 取引ID　【PK】 |
| Amount: Int | 数量 |
| PaymentID: String | 電子決済時に発番されるID |
| Price: Int | 金額 |
| ProductID: String | 商品ID |
| Status: String | 取引ステータス |
| UserID: String | ユーザーID |
| date: String | 日付 |
| Count: Int | 回数 |
| Reslovers | Mutation | addTransaction |  |
| データソース名 | StepFunctionHttpDataSource |  |
| リクエストマッピング |  |  |
| 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\" |  |
| Datasource | データソース名 |  | StepFunctionHttpDataSource | Query：getPoint用のデータソース |
| データソースタイプ |  | HTTPエンドポイント |  |
| HTTPエンドポイント |  | https://states.ap-northeast-1.amazonaws.com/ |  |