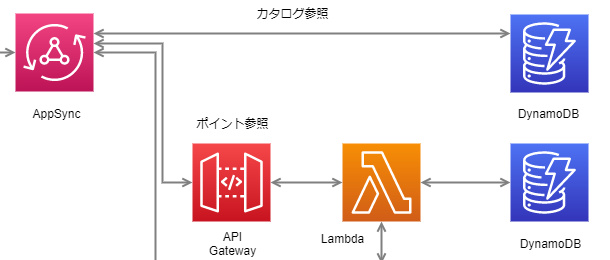
# ポイント参照機能、ポイント付与機能

## 機能イメージ



　※Lambdaはポイント付与にも対応

## 機能概要

### ポイント参照機能

＃機能概要を記載する

＜インプット＞

　userid

＜アウトプット＞

　userid,username,point # Userマスタへもアクセスし、UserNameを取得するでも良いかも

### ポイント付与機能

＃機能概要を記載する

＜インプット＞

　userid,point（購入金額の10%）　＃切り上げ／切り捨てはお任せ

＜アウトプット＞

userid,point（加算後の総ポイント数）

## アクセスURL

### ポイント参照機能

　https://#作成時にセット.execute-api.ap-northeast-1.amazonaws.com/jgs2020ia01/getpoint

## 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 | POST | application/xml |
| /jgs2020ia01/getpoint | GET | application/json |

## 設定

### Lambdaパラメータシート

|  |  |  |  |
| --- | --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 | 対応する設計書 |
| Function |  | PointMaster |  |
| Runtime |  | Node.js 12.x |  |
| Role | アクセス許可 | AWSポジティブテンプレートから新しいロールを作成 |  |
|  | ロール名 | PointMasterLambdaBasicExecutionRole |  |
|  | Policy template | 空欄 |  |
|  | → Create function |  |  |
| index.js |  | Githubで管理 |  |
|  |  |  |  |
|  |  |  |  |

### API Gateway パラメータシート

|  |  |  |  |
| --- | --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 | 対応する設計書 |
| API名 |  | PointManageAPI |  |
| リソース作成 | リソース名 | point |  |
|  | プロキシリソースとして設定 | off |  |
|  | リソースパス | /point |  |
| メソッドの作成 | メソッド | GET |  |
|  | 統合タイプ | Lambda関数 |  |
|  | Lambdaプロキシ統合の使用 | 選択 |  |
|  | Lambdaリージョン | Lambda関数を作成したリージョンを選択する |  |
|  | Lambda関数 | PointMaster |  |
|  | →Save |  |  |
|  | Lambda関数に権限を追加 | OK |  |
| デプロイ | アクション | APIのデプロイを選択 |  |
|  | デプロイされるステージ | 新しいステージ |  |
|  | ステージ名 | jgs2020ia01 |  |
|  | →デプロイ |  |  |
|  |  |  |  |
| ステージ |  | jgs2020ia01　選択 |  |
|  | URL | https://{xxxx}.execute-api.{Region}.amazon.com/jgs2020ia01 |  |
|  | メソッドスロットリング |  |  |
|  | WAF |  |  |
|  | クライアント証明書 |  |  |
| リソース |  | /point　選択 |  |
|  | URLクエリ文字列パラメータ |  |  |
|  | HTTPリクエストヘッダ |  |  |
|  | リクエスト本文 |  |  |
|  | SDK設定 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### AppSyncパラメータシート

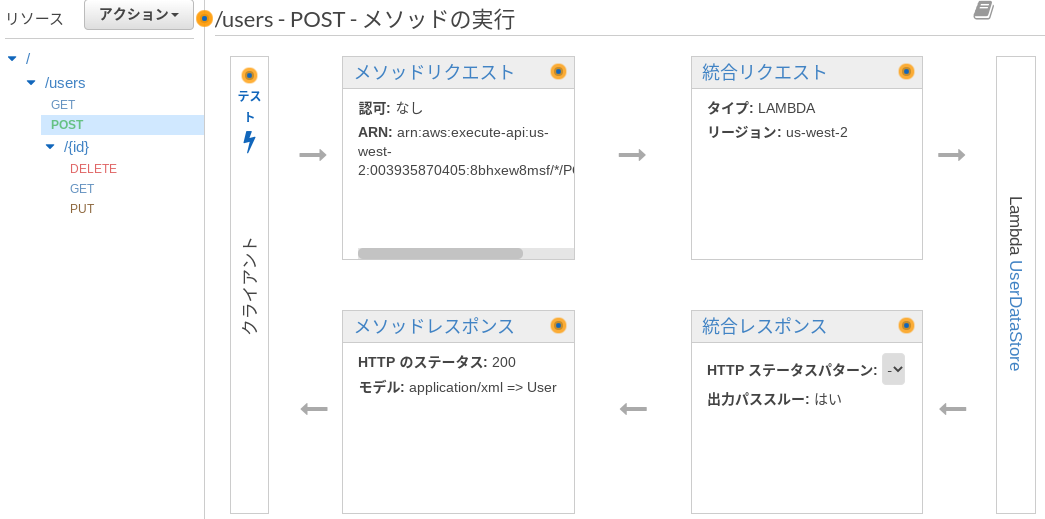
|  |  |  |  |
| --- | --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 | ＃備考 |
| GraphQL API名 |  | Jgs2020ia01-PointManageAPI |  |
| Schema | 定義  Post 型と、Post オブジェクトを追加および取得する  オペレーションを定義 | schema {  query: Query  　　　 mutaion: Mutation  }  type Query {  getPoint(userid: ID): Point  }  type Mutation {  }  type Point {  userid: ID!  username: String  point: int  } |  |
| Schema | Query：getPoint  　データソース名 | HTTP |  |
|  | リクエストマッピング | {  "version": "2018-05-29",  "method": "GET",  "params": {  "headers": {  "Content-Type": "application/json"  }  },  "resourcePath": "/jgs2020ia01/point/${ctx.args.userid}"  } | resourcePath：Lambdaコーディングに依存 |
|  | レスポンスマッピング | ## return the body  #if($ctx.result.statusCode == 200 )  ##if response is 200, check the body for response  #if($util.isNullOrEmpty($ctx.result.body))  null  #else  $ctx.result.body  #end  #else  ##if response is not 200, append the response to error block.  $utils.appendError($ctx.result.body, $ctx.result.statusCode)  #end |  |
| DataSource | データソース名 | HTTP |  |
|  | データソースのタイプ | HTTPエンドポイント |  |
|  | HTTPエンドポイントのURL | https://{xxxx}.execute-api.{Region}.amazon.com | ステージ名は含めない |
|  | リージョン | us-west2（オレゴン） |  |

### Dynamo DB

|  |  |  |  |
| --- | --- | --- | --- |
| 設定　大項目 | 設定　小項目 | 設定　内容 | 対応する設計書 |
| コアコンポーネント | テーブル、項目、属性 | テーブル名、項目名、属性名、データ型、レングス | テーブル一覧、テーブル定義書 |
| プライマリーキー | ・パーティションキー  ・パーティションキーとソートキー |
| セカンダリインデックス | インデックス設定 |
| DynamoDBストリーム | 利用有無 |  |
| 読み込み整合性 | - | 結果整合性 | ・性能要件  　-DB処理要件  ・（仮称）DynamoDB論理設計  　・スループットキャパシティ設定 |
| 読み込み／書き込みキャパシティーモード | オンデマンドモード | - |
| プロビジョニングモード | ・Read Capacity Unit  ・Write Capacity Unit |
| DynamoDBのセットアップ | AWSアクセスキー | ・アクセスキー  ・シークレットキー  ・IAM権限設定（ユーザー、グループ、ロール、ポリシー） | ・AWSセキュリティ設計  　-認証・認可（IAM） |
| DynamoDBへのアクセス | アクセス方法 | ・WEBアクセス（HTTP、API）  ・AWS CLI  ・No SQLWorkbench  ・IPアドレスの範囲 | ・外部インターフェース一覧  ・保守環境設計  ・開発ツール  ・ネットワーク設計 |
| DynamoDB使用開始 |  |  |  |
| AWS SDKの開始方法 | ※アプリケーション側で考慮すべき要素 | | |
|  |  |  |  |

NoSQL Workbench は、DynamoDB テーブルの設計、作成、クエリ、管理に役立つデータモデリング、データ可視化、クエリ開発機能を提供する統合可視化ツールです。





### 