Specifying Item Attributes When Using Expressions

PDF (dynamodb-dg.pdf#Expressions.Attributes)
Kindle (https://www.amazon.com/dp/B0763ZV7JG)
RSS (dynamodbupdates.rss)

This section describes how to refer to item attributes in an expression in Amazon DynamoDB. You can work with any attribute, even if it is deeply nested within multiple lists and maps.

Topics

A Sample Item: ProductCatalog

The following is a representation of an item in the `ProductCatalog` table. (This table is described in Example Tables and Data (./AppendixSampleTables.html) .)

```
{
    "Id": 123,
    "Title": "Bicycle 123",
    "Description": "123 description",
    "BicycleType": "Hybrid",
    "Brand": "Brand-Company C",
    "Price": 500,
    "Color": ["Red", "Black"],
    "ProductCategory": "Bicycle",
    "InStock": true,
    "QuantityOnHand": null,
    "RelatedItems": [
        341,
        472,
        649
    ],
    "Pictures": {
        "FrontView": "http://example.com/products/123_front.jpg",
        "RearView": "http://example.com/products/123_rear.jpg",
        "SideView": "http://example.com/products/123_left_side.jpg"
    },
    "ProductReviews": {
            "FiveStar": [
                        "Excellent! Can't recommend it highly enough! Buy it!",
                        "Do yourself a favor and buy this."
            ],
            "OneStar": [
                        "Terrible product! Do not buy this."
            ]
    },
    "Comment": "This product sells out quickly during the summer",
    "Safety.Warning": "Always wear a helmet"
 }
```

Note the following:

- The partition key value (`Id`) is `123`. There is no sort key.
- Most of the attributes have scalar data types, such as `String`, `Number`, `Boolean`, and `Null`.
- One attribute (`Color`) is a `String Set`.
- The following attributes are document data types:
  - A list of `RelatedItems`. Each element is an `Id` for a related product.
  - A map of `Pictures`. Each element is a short description of a picture, along with a URL for the corresponding image file.
  - A map of `ProductReviews`. Each element represents a rating and a list of reviews corresponding to that rating. Initially, this map is populated with five-star and one-star reviews.

Top-Level Attributes

An attribute is said to be *top level* if it is not embedded within another attribute. For the `ProductCatalog` item, the top-level attributes are as follows:

- `Id`
- `Title`
- `Description`
- `BicycleType`
- `Brand`
- `Price`
- `Color`
- `ProductCategory`
- `InStock`
- `QuantityOnHand`
- `RelatedItems`
- `Pictures`
- `ProductReviews`
- `Comment`
- `Safety.Warning`

All of these top-level attributes are scalars, except for `Color` (list), `RelatedItems` (list), `Pictures` (map), and `ProductReviews` (map).

Nested Attributes

An attribute is said to be *nested* if it is embedded within another attribute. To access a nested attribute, you use *dereference operators*:

- `[n]` — for list elements
- `.` (dot) — for map elements

Accessing List Elements

The dereference operator for a list element is [*n*], where *n* is the element number. List elements are zero-based, so [0] represents the first element in the list, [1] represents the second, and so on. Here are some examples:

- `MyList[0]`
- `AnotherList[12]`
- `ThisList[5][11]`

The element `ThisList[5]` is itself a nested list. Therefore, `ThisList[5][11]` refers to the 12th element in that list.

The number within the square brackets must be a non-negative integer. Therefore, the following expressions are not valid:

- `MyList[-1]`
- `MyList[0.4]`

Accessing Map Elements

The dereference operator for a map element is . (a dot). Use a dot as a separator between elements in a map:

- `MyMap.nestedField`
- `MyMap.nestedField.deeplyNestedField`

Document Paths

In an expression, you use a *document path* to tell DynamoDB where to find an attribute. For a top-level attribute, the document path is simply the attribute name. For a nested attribute, you construct the document path using dereference operators.

The following are some examples of document paths. (Refer to the item shown in Specifying Item Attributes When Using Expressions (./Expressions.Attributes.html) .)

- A top-level scalar attribute.

  ```
  Description
  ```

- A top-level list attribute. (This returns the entire list, not just some of the elements.)

  ```
  RelatedItems
  ```

- The third element from the `RelatedItems` list. (Remember that list elements are zero-based.)

  ```
  RelatedItems[2]
  ```

- The front-view picture of the product.

  ```
  Pictures.FrontView
  ```

- All of the five-star reviews.

  ```
  ProductReviews.FiveStar
  ```

- The first of the five-star reviews.

  ```
  ProductReviews.FiveStar[0]
  ```

Note

The maximum depth for a document path is 32. Therefore, the number of dereferences operators in a path cannot exceed this limit.

You can use any attribute name in a document path, provided that the first character is `a-z` or `A-Z` and the second character (if present) is `a-z`, `A-Z`, or `0-9`. If an attribute name does not meet this requirement, you must define an expression attribute name as a placeholder. For more information, see Expression Attribute Names in DynamoDB (./Expressions.ExpressionAttributeNames.html) .

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.