

Demo of Amazon DynamoDB Operations

Source:	Amazon DynamoDB Developer guide, API guide.
Description:	Demo of DynamoDB provided all Low Level API actions and how to do operations.
By:	Ahsanul Hadi
Email:	adil.gt@gmail.com
Published on:	05-Dec-2014
Status:	This version is working fine. Need to add 3 more API action method.
Language:	Java - jdk-1.8.0_25 - jre-1.8.0_25 - AWS SDK for Java 1.9.7 (comes with AWS Toolkit for Eclipse)
Database:	Amazon Dynamo DB API Version: 2012-08-10
O/S:	Windows 8 (64bit)

(1) Description:

This is a general application to demonstrate and test all the Amazon DynamoDB provided low-level API actions. The following are the low-level API actions, organized by function.

Managing Tables:

- **CreateTable** - Creates a table with user-specified provisioned throughput settings.
- **DescribeTable** - Returns metadata for a table, such as table size, status, and index information.
- **UpdateTable** - Modifies the provisioned throughput settings for a table. Optionally, you can modify the provisioned throughput settings for global secondary indexes on the table.
- **ListTables** - Returns a list of all tables associated with the current AWS account and endpoint.
- **DeleteTable** - Deletes a table and all of its indexes.

Reading Data:

- **GetItem** - Returns a set of attributes for the item that has a given primary key.
- **BatchGetItem** - Performs multiple GetItem requests for data items using their primary keys, from one table or multiple tables.
- **Query** - Returns one or more items from a table or a secondary index.
- **Scan** - Reads every item in a table; the result set is eventually consistent.

Modifying Data:

- **PutItem** - Creates a new item, or replaces an existing item with a new item (including all the attributes).
- **UpdateItem** - Modifies the attributes of an existing item.
- **DeleteItem** - Deletes an item in a table by primary key.
- **BatchWriteItem** - Performs multiple PutItem and DeleteItem requests across multiple tables in a single request.

[Source: <http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>]

(2) Install and using AWS Toolkit for Eclipse:

Prerequisites:

- I. An Amazon Web Services account,
- II. A supported operating system
- III. Java 1.6 or later
- IV. Eclipse IDE for Java Developers 3.6 or later

(Details:

http://docs.aws.amazon.com/AWSToolkitEclipse/latest/GettingStartedGuide/tke_setup_prereqs.html)

How to install:

Install AWS Toolkit for Eclipse - <http://aws.amazon.com/eclipse/> .

Follow the below steps:

First go to <http://download.eclipse.org/releases/helios> (or your Eclipse version i.e. Juno) and install Database development. Then go to <http://aws.amazon.com/eclipse> and install Amazon EC2 Management. And finally install the AWS Toolkit for Eclipse.

Source: <http://stackoverflow.com/questions/21847788/cannot-install-aws-toolkit-for-eclipse-how-to-fix-these-errors>

Getting Started with AWS Toolkit:

<http://docs.aws.amazon.com/AWSToolkitEclipse/latest/GettingStartedGuide/Welcome.html>

(3) AWS Credential setting:

You have to create the credentials in IAM. Detail:

http://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html

For this demo, the default credential file is used that is located in:

C:\Users\\.aws\credentials [For Windows]

Sample entry:

```
[alias user name]
aws_access_key_id=<IAM access key>
aws_secret_access_key=<IAM secret key>
```

Note:

- This is only a demo for learning purpose.
- Updated code will be added periodically.
- Code not optimized.