COURSEWARE

AWS Java SDK

Contents

Overview

AWS SDK for Java support will enable developers to to easily work within the AWS environment and build scalable solutions with Amazon S3, DynamoDB and so on.

This tutorial will go over how we can add AWS Java SDK to our spring-boot application and retrieve a list of tables created in DynamoDB.

Prerequisites

- Linux Machine
- AWS CLI
- Maven Installed
- Spring-boot

Linux Machine

We will be using Ubuntu 18.04 LTS in this section, but any Linux Machine will work (as long as you can change the package manager and install the correct packages accordingly).

AWS CLI

You will need to download the AWS CLI:

```
sudo apt install awscli -y
```

Run:

```
aws configure
```

Ensure you are using the Access and Secret Key of an account that has administrative permissions. This is only an example and it is not best practice to use an account that has administrative access. For this example, you can use **DynamoDB full access**.

Maven Installed

You will require a machine that has both Java and Maven installed. The following example is for Ubuntu 18.04LTS:

```
sudo apt install openjdk-8-jdk maven -y
```

Spring-boot

Ideally, you will require an existing spring-boot application for this. If you do not have any existing spring-boot code, you will need at least the basic understanding of the springboot framework.

Tutorial

1. Open a Maven project that has a pom.xml file.
2. Add the following codes to pom.xml:

```
<dependencyManagement>
    <dependencies>
        <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk-bom</artifactId>
        <version>1.11.327</version>
        <type>pom</type>
        <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

The following dependency will allow you access to all the AWS resources through your Java application. This is dependent on whether the user logged in through AWS configure and also has the same permission. For instance, if the user logged in and only has DynamoDB permissions, then your Java application cannot access AWS Lambda, and other services.

```
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk</artifactId>
    <version>1.11.327</version>
</dependency>
```

3. Add the following code to your Java application:

```
import org.springframework.beans.BeanUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*

import com.amazonaws.AmazonServiceException;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.services.dynamodbv2.model.ListTablesRequest;
import com.amazonaws.services.dynamodbv2.model.ListTablesResult;

@RestController
@RequestMapping(value = "/tables")
@CrossOrigin(origins = "http://localhost:8080")
public class AccountDBController {

    final AmazonDynamoDB ddb = AmazonDynamoDBClientBuilder.defaultClient();
    ListTablesRequest request;
    boolean more_tables = true;
    String last_name = null;

    @RequestMapping(value = "/get-ddb-table", method = RequestMethod.GET)
    public void getTables() {
        while(more_tables) {
            if (last_name == null) {
                request = new ListTablesRequest().withLimit(10);
            }
            else {
                request = new ListTablesRequest()
                        .withLimit(10)
                        .withExclusiveStartTableName(last_name);
            }

            ListTablesResult table_list = ddb.listTables(request);
            List<String> table_names = table_list.getTableNames();

            if (table_names.size() > 0) {
                for (String cur_name : table_names) {
                    System.out.format("* %s\n", cur_name);
                }
            } else {
                System.out.println("No tables found!");
            }

            last_name = table_list.getLastEvaluatedTableName();
            if (last_name == null) {
                more_tables = false;
            }
        }
    }
}
```

4. Run the following commands:

```
mvn package

mvn clean install

mvn spring-boot:run
```

5. When sending a request to the url path, you will not receive anything in return. You will have to navigate to your application and, in the terminal, it will print out the list of tables you have in DynamoDB.

6. Create some tables in DynamoDB. This way when you run your application you can see your tables in the terminal.

Exercises

There are no exercises for this module.