**AWS SNS + Direct SMS + Java Example**

This guide explains how to **send SMS directly using AWS SNS in Java** without using a topic.

**1. Prerequisites**

✔ **AWS Account** – You need an active AWS account.  
✔ **IAM Permissions** – Ensure your AWS user has **SNS publish permissions**.  
✔ **AWS SDK for Java** – Install AWS SDK dependencies in pom.xml.

**2. Add Dependencies (pom.xml)**

Include these dependencies in your Spring Boot project:

<dependencies>

<!-- AWS SDK for SNS -->

<dependency>

<groupId>software.amazon.awssdk</groupId>

<artifactId>sns</artifactId>

<version>2.20.0</version>

</dependency>

<!-- SLF4J Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

</dependencies>

**3. Java Code to Send Direct SMS Using AWS SNS**

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

import software.amazon.awssdk.regions.Region;

import software.amazon.awssdk.services.sns.SnsClient;

import software.amazon.awssdk.services.sns.model.PublishRequest;

import software.amazon.awssdk.services.sns.model.PublishResponse;

public class AwsSnsSmsSender {

public static void main(String[] args) {

String phoneNumber = "+1234567890"; // Replace with the recipient's phone number

String message = "Hello! This is a test message from AWS SNS.";

// Create SNS Client

SnsClient snsClient = SnsClient.builder()

.region(Region.US\_EAST\_1) // Change to your AWS region

.credentialsProvider(DefaultCredentialsProvider.create())

.build();

// Send SMS

PublishRequest request = PublishRequest.builder()

.message(message)

.phoneNumber(phoneNumber)

.build();

PublishResponse response = snsClient.publish(request);

System.out.println("Message sent! Message ID: " + response.messageId());

// Close the SNS Client

snsClient.close();

}

}

**4. Explanation**

✔ **SnsClient.builder()** – Creates an SNS client with your AWS credentials.  
✔ **phoneNumber** – Direct phone number (must include + and country code).  
✔ **PublishRequest.builder()** – Sends an SMS without needing a topic.  
✔ **response.messageId()** – Confirms the message was sent.

**5. How to Run the Code?**

1. **Configure AWS Credentials**
   * Set up your AWS credentials in ~/.aws/credentials (Linux/Mac) or C:\Users\YourUser\.aws\credentials (Windows).
2. [default]
3. aws\_access\_key\_id=YOUR\_ACCESS\_KEY
4. aws\_secret\_access\_key=YOUR\_SECRET\_KEY
5. **Compile and Run**
6. mvn clean install
7. java -jar target/your-app.jar

**6. Expected Output**

Message sent! Message ID: abc123-xyz456

**7. Common Issues & Fixes**

| **Issue** | **Solution** |
| --- | --- |
| **Invalid phone number** | Ensure it starts with + and country code (e.g., +91 for India, +1 for the USA). |
| **Access Denied (403 error)** | Ensure your IAM user has SNS:Publish permission. |
| **Region Error** | Use a valid AWS region where SMS is supported. |