

## Using AWS Comprehend in Java

In this task, you will learn how to use **AWS Comprehend**, a Natural Language Processing (NLP) service, to analyse text in Java. You will build a simple Java program that:

1. **Detect Languages** - Detects the **dominant language** of the input text.
2. **Performs Sentiment Analysis** – Determines whether a given text is **positive, negative, neutral, or mixed**.
3. **Extracts Named Entities** – Identifies important words like **people, places, dates, and organizations** in the text.
4. **Extracts Key Phrases** - Extracts important words or phrases from the input text (useful for text summarisation).
5. **Detects Personally Identifiable Information** - helps automatically identify and redact sensitive information to enhance security, ensure compliance, and prevent data leaks.

By completing this task, you will gain hands-on experience with AWS Comprehend, the AWS SDK for Java, and text analytics concepts. This will help you understand how **machine learning can be used to analyse and understand human language**.

---

### Step 1: Setting Up Your Java Project in Eclipse

1. **Create a new Maven project in Eclipse:**
  - Open Eclipse → File → New → Maven Project.
  - Select a workspace and choose the **skip archetype** for a simple Java project (e.g., maven-archetype-quickstart).
  - Click **Finish**.
2. **Add AWS SDK for Comprehend to your pom.xml:** Open your pom.xml file and add the following dependency inside the <dependencies> tag:

```
<dependencies>
  <dependency>
    <groupId>software.amazon.awssdk</groupId>
    <artifactId>comprehend</artifactId>
    <version>2.20.26</version>
  </dependency>
</dependencies>
```

---

### Step 2: AWS Credentials and Utils Class

AWS SDK for Java requires access to your **AWS credentials**. The easiest way to manage credentials locally is by setting them up with the **AWS CLI** or using environment variables. **But for this lab we will add the credentials to our code base.**

**2.1** Create a new java class called “**AWSComprehendUtils**” and set it up with your AWS access credentials (**lecture slide 27**).

---

### Step 3: Implementing AWS Rekognition methods.

1. In the **AWSComprehendUtils**, implement a **Detect Language** method (**lecture slide 28**).
2. Create a new class with a main method called “**AWSComprehendController**” and test the detect language method (**lecture slide 28**). You can try testing phrases in different languages to test if Comprehend is able to accurately detect the language. Remember this API call will just return the language code e.g. en=English, ur=urdu, ja=Japanese, ar=Arabic etc.
3. In the **AWSComprehendUtils** implement a **Detect Sentiment** method and test it in the **AWSComprehendController** (**lecture slide 29**). You can use some text of your choice or use the example text from the lecture.
4. In the **AWSComprehendUtils** implement a **Detect Entities** method and test it in the **AWSComprehendController** (**lecture slide 30**). You can use some text of your choice or use the example text from the lecture slide.
- 5.
6. In the **AWSComprehendUtils** implement a **Detect Key Phrases** method and test it in the **AWSComprehendController** (**lecture slide 31**). You can use some text of your choice or use the example text from the lecture slide.
7. In the **AWSComprehendUtils** implement a **Detect Pii** (Personally Identifiable Information) method and test it in the **AWSComprehendController** (**lecture slide 32**). You can use some text of your choice or use the example text from the lecture slide.

### Extension Task

#### Modify the text input

Try different paragraphs, tweets, or news articles and compare how AWS Comprehend responds with each of the methods you’ve implemented.