Contents

[Versions 2](#_Toc105154947)

[Set Up 2](#_Toc105154948)

[Using the SDK with Apache Maven 3](#_Toc105154949)

[**Importing all SDK modules** 3](#_Toc105154950)

[**Specifying individual SDK modules** 3](#_Toc105154951)

Versions

🡪AWS SDK for Java, version 1.x (J2SE Development Kit 6.0 or later)

Java version 1.6 (J2SE 6.0) did not have built-in support for SHA256-signed SSL certificates, which are required for all HTTPS connections with AWS after September 30, 2015.

Java versions 1.7 or newer are packaged with updated certificates and are unaffected by this issue.

🡪AWS SDK for Java, version 2.x (Java 8 and later recommended)

Reference: <https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html>

The AWS SDK for Java provides a Java API for AWS services. Using the SDK, you can easily build Java applications that work with Amazon S3, Amazon EC2, DynamoDB, and more.

The AWS SDK for Java 2.x is a major rewrite of the version 1.x code base. It’s built on top of Java 8+.

These include support for non-blocking I/O and the ability to plug in a different HTTP implementation at run time.

Before, you need an active AWS account, an AWS Identity and Access Management (IAM) user with a programmatic access key and permissions to Amazon S3, and a Java development environment configured to use that access key as credentials for AWS.

**AWS SDK 1.11 to 2.0 changelog reference:**

[https://github.com/aws/aws-sdk-java-v2/blob/master/docs/LaunchChangelog.md#7-high-level-libraries](https://github.com/aws/aws-sdk-java-v2/blob/master/docs/LaunchChangelog.md%237-high-level-libraries)

**SDK - 1.x group id**

<groupId>com.amazonaws</groupId>

**SDK - 2.x group id**

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

Set Up

Follow these steps to set up for this tutorial:

* [Create an AWS account](https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html#get-started-setup-account)
* [Create an IAM user](https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html#get-started-setup-user)
* [Install Java and Apache Maven](https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html#get-started-setup-javamaven)

Your development environment needs to have Java 8 or later and Apache Maven installed.

* [Configure credentials](https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html#get-started-setup-credentials)

Configure your development environment with your Access Key ID and the Secret Access Key. The AWS SDK for Java uses this access key as credentials when your application makes requests to Amazon Web Services.

* In a text editor, create a new file with the following code:

aws\_access\_key\_id = YOUR\_AWS\_ACCESS\_KEY\_ID

aws\_secret\_access\_key = YOUR\_AWS\_SECRET\_ACCESS\_KEY

* In the text file you just created, replace YOUR\_AWS\_ACCESS\_KEY with your unique AWS access key ID, and replace YOUR\_AWS\_SECRET\_ACCESS\_KEY with your unique AWS secret access key.
* Save the file without a file extension. Refer to the following table for the correct location and file name based on your operating system.

Windows 🡪 C:\Users\<yourUserName>\.aws\credentials

Linux, macOS, Unix 🡪 ~/.aws/credentials

Using the SDK with Apache Maven

To use the AWS SDK for Java in your project, you’ll need to declare it as a dependency in your project’s pom.xml file.

You can import [**individual components**](https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-project-maven.html#configuring-maven-individual-components)**or the**[**entire SDK**](https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-project-maven.html#configuring-maven-entire-sdk).

### **Importing all SDK modules**

If you would like to pull in the entire SDK as a dependency, don’t use the BOM method, but simply declare it in your pom.xml like this:

<dependencies>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk</artifactId>

<version>1.11.1000</version>

</dependency>

</dependencies>

### **Specifying individual SDK modules**

To select individual SDK modules, use the AWS SDK for Java bill of materials (BOM) for Maven, which will ensure that the modules you specify use the same version of the SDK and that they’re compatible with each other.

To use the BOM, add a <dependencyManagement> section to your application’s pom.xml file, adding aws-java-sdk-bom as a dependency and specifying the version of the SDK you want to use.

<dependencyManagement>

<dependencies>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-bom</artifactId>

<version>1.11.1000</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

You can now select individual modules from the SDK that you use in your application. Because you already declared the SDK version in the BOM, you don’t need to specify the version number for each component.

<dependencies>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-s3</artifactId>

</dependency>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-dynamodb</artifactId>

</dependency>

</dependencies>

sdsds