# Spring Boot with Amazon S3 : File Upload & Download Example | S3 Bucket

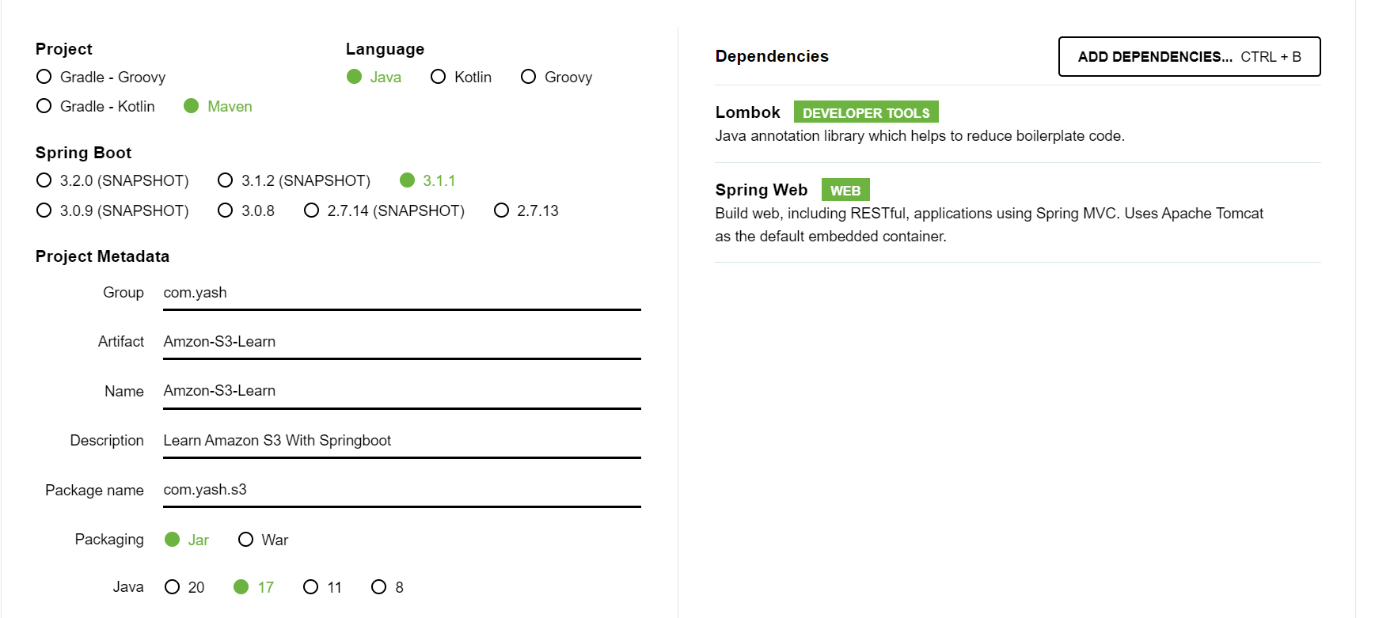
**Approaches for storing files.**

* Directly on the server where application is deployed.
* On database as blob files
* Using some cloud storage platforms

**Amazon S3 (Simple Storage Service)**

Advantages of using amazon s3

1. It’s easy to upload any files programmatically using API.
2. Amazon supports different coding languages.
3. It provides GUI so that we can see graphically what files are uploaded.
4. If we want, we can manually upload and delete the files that we want to.



Start with Spring starter IO and add dependency of Lombok to avoid boiler plate code and add spring web to have web things.

Amazon SDK is not present in the spring starter dependency section so get the dependency from central maven repository.

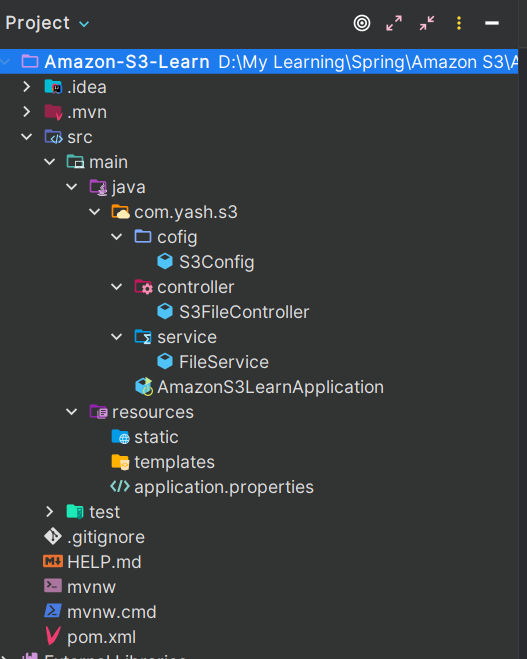
A screen shot of a computer code

Description automatically generated

* Create bucket from your amazon cloud and generate the s3 bucket access key and secret id.
* Add the following into your application properties directly if your making stand alone application or if you want to post in GitHub or some any other production work then you must use your environmental variables.



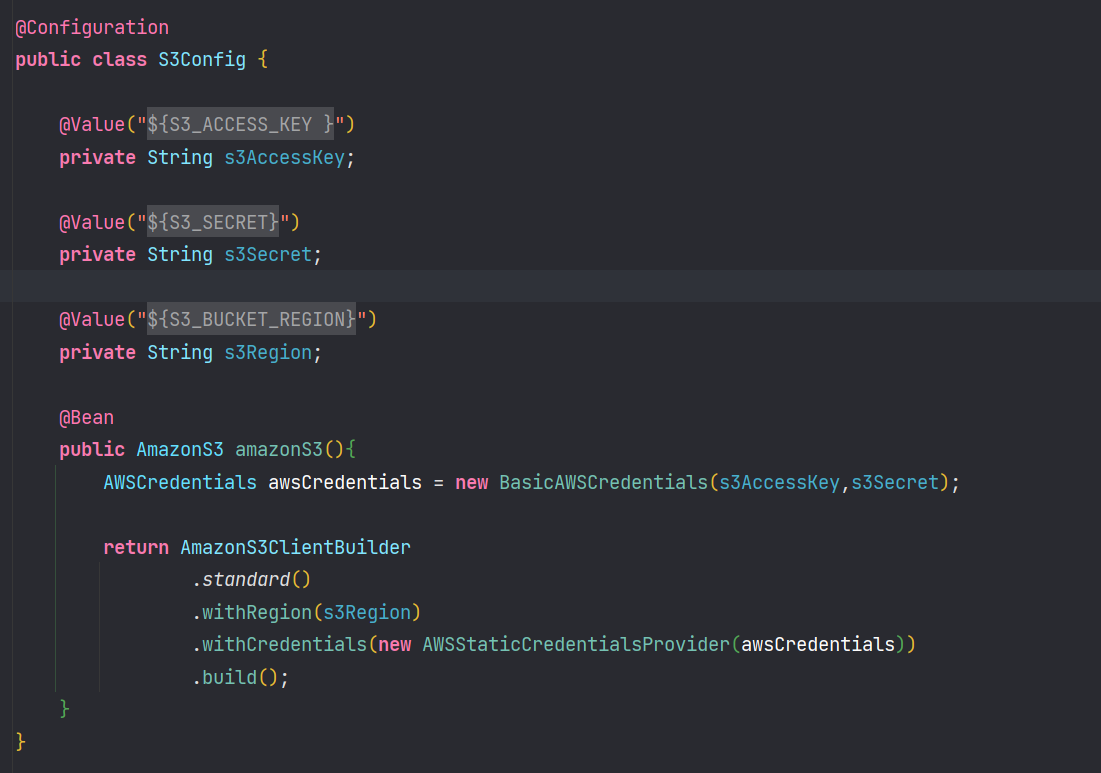
Project structure



**About Packages**

* **Config** : It is a configuration class, in which we will write the code about configuring the bean of amazon s3
* **Controller** : It is a rest controller class where we will define all REST methods
* **Service** : It is class which contains all the required service methods for controllers.

**package com.yash.s3.cofig;**



* Import all values using @Value annotation into variables.
* Create an object of AWSCredentails interface object and instantiate the object using implementing class of BasicAWSCredentials;
* Now using amazon s3 client builder which returns the object of amazons3 object, and it will take the region and AWS credentials.

**package com.yash.s3.service.FileService;**

Now let’s go to the service class by name FileService where we written the methods that contains business logic of below methods.

* Method to save file in s3 bucket.
* Method to delete file by filename from our s3 bucket.
* Method to fetch the file by using file name.
* Method to fetch all file name that contains in our respective s3 bucket.

**Method logic that we used to save file in s3 bucket:**

**A computer screen with text

Description automatically generated**

* So, we will get data of file from user in multipart format where we must convert multipart to file format.
* We must create a method that converts multipart to file so that we can upload our file to s3 bucket. (S3 bucket does not accept multipart format)

**Simple method that returns file by consuming multipart**

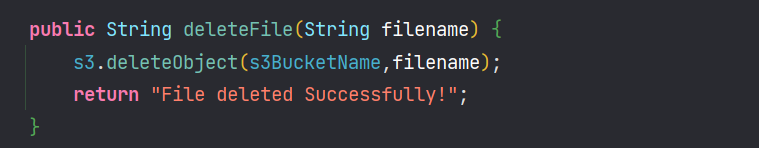
**A computer screen shot of a program code

Description automatically generated**

* First create a file object with original file name or you can use some random UUID for the file name.
* Next create an output stream which allows us to write data into file.
* Pass the multipart data to output stream and return the file.

Now by using put object method from s3 which takes the input of **bucket name, filename, and file** and return the instance of PutObjectResult in which we can get metadata of file that was stored in s3 bucket. I am returning the md5 hash of file to controller.

**Simple method to delete file form bucket using filename.**

****

**Method to get all file names that contains in the s3 bucket.**

**A computer code on a black background

Description automatically generated**

* S3 list objects method will return the objects listings which returns the list of s3 object summary instances which each object resembles a file in our S3 bucket.
* So, we can loop all the list of object summaries and get all file names.

**Method that allows users to download file using filename.**

**A computer screen shot of code

Description automatically generated**

* When we want to allow people to download our files through a API we have to transfer by converting to byte otherwise if we transfer directly S3 object directly then we are exposing lot of details to user because S3 object contains lot of data about bucket name and many more

**Controller Mappings (/api/v1/s3-learn)**

* File upload to S3 Bucket : /upload
* Delete file by file name : /{filename}
* To Download : /download/{filename}
* To fetch all file names : /files

**A computer code on a black background

Description automatically generated­­­­**

**A screenshot of a computer code

Description automatically generated**