**SpringBoot整合阿里云OSS存储服务**

**摘要**：OSS可用于图片、音视频、日志等海量文件的存储。各种终端设备、Web网站程序、移动应用可以直接向OSS写入或读取数据，非常方便。本篇文章介绍如何在java的SpringBoot项目中，整个使用OSS服务，spring 项目也可以参考。

主要步骤如下：

前提是开通了阿里云OSS服务，然后；

1.引入依赖

2.获取关键参数，如endpoint,accessKeyId等，这个进入阿里云OSS控制台即可获取

3.建一个文件上传工具类

4.调用工具类，上传文件

**1.pom.xml**

1. <!--aliyunOSS-->
2. <dependency>
3. <groupId>com.aliyun.oss</groupId>
4. <artifactId>aliyun-sdk-oss</artifactId>
5. <version>2.4.0</version>
6. </dependency>
7. <dependency>
8. <groupId>commons-fileupload</groupId>
9. <artifactId>commons-fileupload</artifactId>
10. <version>1.3.1</version>
11. </dependency>

**2.配置文件**

注意：为了以后的拓展性，代码和配置解耦，我这里会专门建一个配置文件，然后在使用时直接从配置类中获取，而不是写死在java代码中。所以2,3两步不是必须的，作用只是规范的使用相关配置参数，若觉得麻烦，学习时，可直接在java代码中写死。

application.properties  这里是我定义的一些必要常量

1. #OSS
2. java4all.file.endpoint=oss-cn-shanghai.aliyuncs.com 不同的服务器，地址不同
3. java4all.file.keyid=去OSS控制台获取
4. java4all.file.keysecret=去OSS控制台获取
5. java4all.file.bucketname1=java4all-file 这个自己创建bucket时的命名，控制台创建也行，代码创建也行
6. java4all.file.filehost=blog 文件路径，我这里是blog

**3.ConstantProperties.java**

这是我定义的常量类，读取配置文件application.properties中的配置，定义为常量，方便使用

1. package com.java4all.config;
3. import org.springframework.beans.factory.InitializingBean;
4. import org.springframework.beans.factory.annotation.Value;
5. import org.springframework.stereotype.Component;
7. */\*\**
8. *\* Created by lightClouds917*
9. *\* Date 2018/1/16*
10. *\* Description:配置文件配置项*
11. *\*/*
12. @Component
13. public class ConstantProperties implements InitializingBean{
15. @Value("${java4all.file.endpoint}")
16. private String java4all\_file\_endpoint;
18. @Value("${java4all.file.keyid}")
19. private String java4all\_file\_keyid;
21. @Value("${java4all.file.keysecret}")
22. private String java4all\_file\_keysecret;
24. @Value("${java4all.file.filehost}")
25. private String java4all\_file\_filehost;
27. @Value("${java4all.file.bucketname1}")
28. private String java4all\_file\_bucketname1;

31. public static String JAVA4ALL\_END\_POINT ;
32. public static String JAVA4ALL\_ACCESS\_KEY\_ID ;
33. public static String JAVA4ALL\_ACCESS\_KEY\_SECRET ;
34. public static String JAVA4ALL\_BUCKET\_NAME1 ;
35. public static String JAVA4ALL\_FILE\_HOST ;
37. @Override
38. public void afterPropertiesSet() throws Exception {
39. JAVA4ALL\_END\_POINT = java4all\_file\_endpoint;
40. JAVA4ALL\_ACCESS\_KEY\_ID = java4all\_file\_keyid;
41. JAVA4ALL\_ACCESS\_KEY\_SECRET = java4all\_file\_keysecret;
42. JAVA4ALL\_FILE\_HOST = java4all\_file\_filehost;
43. JAVA4ALL\_BUCKET\_NAME1 = java4all\_file\_bucketname1;
44. }
45. }

**4.工具类  AliyunOSSUtil.java**

1. package com.java4all.utils;
3. import com.aliyun.oss.ClientException;
4. import com.aliyun.oss.OSSClient;
5. import com.aliyun.oss.OSSException;
6. import com.aliyun.oss.model.CannedAccessControlList;
7. import com.aliyun.oss.model.CreateBucketRequest;
8. import com.aliyun.oss.model.PutObjectRequest;
9. import com.aliyun.oss.model.PutObjectResult;
10. import com.java4all.config.ConstantProperties;
11. import org.slf4j.LoggerFactory;
13. import java.io.File;
14. import java.text.SimpleDateFormat;
15. import java.util.Date;
16. import java.util.UUID;
18. */\*\**
19. *\* Created by lightClouds917*
20. *\* Date 2018/2/7*
21. *\* Description:aliyunOSSUtil*
22. *\*/*
23. public class AliyunOSSUtil {
25. private static final org.slf4j.Logger logger = LoggerFactory.getLogger(AliyunOSSUtil.class);

28. public static String upload(File file){
29. logger.info("=========>OSS文件上传开始："+file.getName());
30. String endpoint=ConstantProperties.JAVA4ALL\_END\_POINT;
31. String accessKeyId=ConstantProperties.JAVA4ALL\_ACCESS\_KEY\_ID;
32. String accessKeySecret=ConstantProperties.JAVA4ALL\_ACCESS\_KEY\_SECRET;
33. String bucketName=ConstantProperties.JAVA4ALL\_BUCKET\_NAME1;
34. String fileHost=ConstantProperties.JAVA4ALL\_FILE\_HOST;
36. SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
37. String dateStr = format.format(new Date());
39. if(null == file){
40. return null;
41. }
43. OSSClient ossClient = new OSSClient(endpoint,accessKeyId,accessKeySecret);
44. try {
45. *//容器不存在，就创建*
46. if(! ossClient.doesBucketExist(bucketName)){
47. ossClient.createBucket(bucketName);
48. CreateBucketRequest createBucketRequest = new CreateBucketRequest(bucketName);
49. createBucketRequest.setCannedACL(CannedAccessControlList.PublicRead);
50. ossClient.createBucket(createBucketRequest);
51. }
52. *//创建文件路径*
53. String fileUrl = fileHost+"/"+(dateStr + "/" + UUID.randomUUID().toString().replace("-","")+"-"+file.getName());
54. *//上传文件*
55. PutObjectResult result = ossClient.putObject(new PutObjectRequest(bucketName, fileUrl, file));
56. *//设置权限 这里是公开读*
57. ossClient.setBucketAcl(bucketName,CannedAccessControlList.PublicRead);
58. if(null != result){
59. logger.info("==========>OSS文件上传成功,OSS地址："+fileUrl);
60. return fileUrl;
61. }
62. }catch (OSSException oe){
63. logger.error(oe.getMessage());
64. }catch (ClientException ce){
65. logger.error(ce.getMessage());
66. }finally {
67. *//关闭*
68. ossClient.shutdown();
69. }
70. return null;
71. }
72. }

**5.页面  upload.html**

Springboot如何写页面，参考：[Spring Boot---(16)Spring Boot使用Thymeleaf开发web页面](http://blog.csdn.net/weixin_39800144/article/details/79281686)

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8"/>
5. <title>文件上传</title>
6. </head>
7. <body>
8. <form method="POST" enctype="multipart/form-data" action="/upload/uploadBlog">
9. <p>文件：<input type="file" name="file"/></p>
10. <p><input type="submit" value="上传" /></p>
11. </form>
12. </body>
13. </html>

**6.使用   UploadController.java**

1. package com.java4all.controller;
3. import com.java4all.utils.AliyunOSSUtil;
4. import org.slf4j.LoggerFactory;
5. import org.springframework.stereotype.Controller;
6. import org.springframework.web.bind.annotation.RequestMapping;
7. import org.springframework.web.bind.annotation.RequestMethod;
8. import org.springframework.web.multipart.MultipartFile;
10. import java.io.File;
11. import java.io.FileOutputStream;
13. */\*\**
14. *\* Created by lightClouds917*
15. *\* Date 2018/2/7*
16. *\* Description:文件上传*
17. *\*/*
18. @Controller
19. @RequestMapping("upload")
20. public class UploadController {
21. private final org.slf4j.Logger logger = LoggerFactory.getLogger(getClass());
22. */\*\**
23. *\* 文件上传*
24. *\* @param file*
25. *\*/*
26. @RequestMapping(value = "uploadBlog",method = RequestMethod.POST)
27. public String uploadBlog(MultipartFile file){
29. logger.info("============>文件上传");
30. try {
32. if(null != file){
33. String filename = file.getOriginalFilename();
34. if(!"".equals(filename.trim())){
35. File newFile = new File(filename);
36. FileOutputStream os = new FileOutputStream(newFile);
37. os.write(file.getBytes());
38. os.close();
39. file.transferTo(newFile);
40. *//上传到OSS*
41. String uploadUrl = AliyunOSSUtil.upload(newFile);
43. }
45. }
46. }catch (Exception ex){
47. ex.printStackTrace();
48. }
49. return "index";
50. }
52. @RequestMapping(value = "toUploadBlog",method = RequestMethod.GET)
53. public String toUploadBlog(){
54. return "upload";
55. }
57. }

**7.上传**

访问：http://localhost:8088/upload/toUploadBlog

页面：

点击上传，跳转页面到：

上传成功

**8.查看控制台**

打开OSS控制台，可以查看我们上传的文件

下一篇，写文件的下载和删除......