阿里云之OSS对象存储

课程目标

1. 介绍阿里云的OSS对象存储服务
2. 开通OSS存储对象服务并使用
3. 使用Java SDK 操作oss
4. 介绍阿里云的OSS对象存储服务

## 1.1阿里云对象存储服务

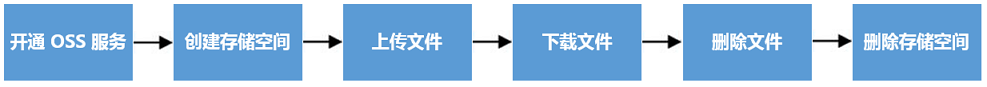
阿里云对象存储服务（Object Storage Service，简称 OSS）为您提供基于网络的数据存取服务。使用 OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

阿里云 OSS 将数据文件以对象（Object）的形式上传到存储空间（Bucket）中。您可以进行以下操作：

* 创建一个或者多个存储空间，向每个存储空间中添加一个或多个文件。
* 通过获取已上传文件的地址进行文件的分享和下载。
* 通过修改存储空间或文件的读写权限（ACL）来设置访问权限。
* 通过阿里云管理控制台、各种便捷工具以及丰富的 SDK 包执行基本和高级 OSS 操作。

## 1.2 使用 OSS 管理控制台

使用阿里云管理控制台来完成 OSS 基本操作的流程如下：



1. [开通 OSS 服务](https://help.aliyun.com/document_detail/31884.html#task-njz-hf4-tdb)
2. [创建存储空间](https://help.aliyun.com/document_detail/31885.html#task-u3p-3n4-tdb)
3. [上传文件](https://help.aliyun.com/document_detail/31886.html#task-zx1-4p4-tdb)
4. [下载文件](https://help.aliyun.com/document_detail/31887.html#task-wlr-lr4-tdb)
5. [删除文件](https://help.aliyun.com/document_detail/31888.html#task-pgg-qt4-tdb)
6. [删除存储空间](https://help.aliyun.com/document_detail/31889.html#task-pkl-qdp-tdb)
7. 开通oss服务

1. 开通“对象存储OSS”服务

（1）申请阿里云账号

（2）实名认证

（3）开通“对象存储OSS”服务

（4）进入管理控制台

2、创建Bucket

选择：标准存储、公共读、不开通





3、管理存储空间中的文件

注意:存储空间是可以创建文件夹的。



上传文件、查询文件详细信息、删除文件…..

1. 使用Java SDK 操作oss

1. 操作前准备:找到编码时需要用到的常量值

（1）endpoint 服务器对应的节点



（2）bucketName 存储空间名称

（3）accessKeyId 阿里云主账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用RAM账号进行API访问或日常运维，请登录RAM控制台创建RAM账号

（4）accessKeySecret

2. 使用Java SDK操作OSS的步骤

2.1 导入Java SDK依赖

|  |
| --- |
| *<!--aliyunOSS-->* <**dependency**>  <**groupId**>com.aliyun.oss</**groupId**>  <**artifactId**>aliyun-sdk-oss</**artifactId**>  <**version**>3.10.2</**version**> </**dependency**> |

2.2 实现对象管理的操作

### 实现文件上传

|  |
| --- |
| **public static void** main(String[] args) **throws** FileNotFoundException {  *// Endpoint以杭州为例，其它Region请按实际情况填写。* String endpoint = **"** endpoint **"**;  *// 阿里云主账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用RAM账号进行API访问或日常运维，请登录RAM控制台创建RAM账号。* String accessKeyId = **"** accessKeyId **"**;  String accessKeySecret = **"** accessKeySecret **"**;  String bucketName = **"k9510"**;  *// 1.创建OSSClient实例。* OSS ossClient = **new** OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);  //2.使用OSSClient对象的方法操作对象  *//设置oss实例的访问权限：公共读* ossClient.setBucketAcl(bucketName, CannedAccessControlList.***PublicRead***);  *// 创建上传文件* FileInputStream fis=**new** FileInputStream(**"d://16.jpg"**);  ossClient.putObject(bucketName,**"18.jpg"**,fis);   **//保存数据库中的地址 ------注意：bucket的权限为公共读  //uploadUrl = "http://" + bucketName + "." + endPoint + "/" + fileUrl; // 关闭OSSClient。**ossClient.shutdown();  System.***out***.println(**"成功了"**);  } |

注意：在网页中使用img标签引用图片的路径即可显示，不能只接在浏览器中输入地址访问

<img src="https://k9510.oss-cn-beijing.aliyuncs.com/16.jpg"/>

## 实现文件下载

|  |
| --- |
| **public class** 下载文件 {  **public static void** main(String[] args) **throws** Exception {  *// Endpoint以杭州为例，其它Region请按实际情况填写。* String endpoint = **"** endpoint **"**;  *// 阿里云主账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用RAM账号进行API访问或日常运维，请登录RAM控制台创建RAM账号。* String accessKeyId = **"** accessKeyId **"**;  String accessKeySecret = **"** accessKeySecret **"**;  String bucketName = **"k9510"**; *// 创建OSSClient实例。* OSS ossClient = **new** OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);   *//设置oss实例的访问权限：公共读  //ossClient.setBucketAcl(bucketName, CannedAccessControlList.PublicRead);*   **// 创建上传文件  // 下载OSS文件到本地文件。如果指定的本地文件存在会覆盖，不存在则新建。  ossClient.getObject(new GetObjectRequest(bucketName, "16.jpg"), new File("d://k.jpg"));**  System.***out***.println(**"下载成功了"**);  } } |

## 实现文件上传

|  |
| --- |
| **public class** 删除文件 {  **public static void** main(String[] args) **throws** FileNotFoundException {  *// Endpoint以杭州为例，其它Region请按实际情况填写。* String endpoint = **"** endpoint **"**;  *// 阿里云主账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用RAM账号进行API访问或日常运维，请登录RAM控制台创建RAM账号。* String accessKeyId = **"** accessKeyId **"**;  String accessKeySecret = **"** accessKeySecret **"**;  String bucketName = **"k9510"**; *// 创建OSSClient实例。* OSS ossClient = **new** OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);  **// 创建上传文件  ossClient.deleteObject(bucketName,"18.jpg");**  *// 关闭OSSClient。* ossClient.shutdown();  System.***out***.println(**"删除成功了"**);  } } |

## 扩展:创建RAM子用户管理OSS(便于权限管理)





（1）添加用户组





注意: 创建用户时需要记住: AccessKeyID\AccessKey Secret

（3）添加用户到用户组

（4）设置用户组权限：AliyunOSSFullAccess

（5）获取子用户AccessKeyId,AccessKeySecret

也可以创建子用户访问OSS对象

## Springboot操作oss配置和属性文件

## springboot配置

|  |
| --- |
| #阿里云 OSS  #不同的服务器，地址不同  aliyun.oss.file.endpoint=your endpoint  aliyun.oss.file.keyid=your accessKeyId  aliyun.oss.file.keysecret=your accessKeySecret  #bucket可以在控制台创建，也可以使用java代码创建  aliyun.oss.file.bucketname=k9510  aliyun.oss.file.filehost=avatar #存储空间的文件路径 |

## 读取配置

|  |
| --- |
| /\*\*\* 常量类，读取配置文件application.properties中的配置\*/@Component[//@PropertySource("classpath:application.properties")](mailto://@PropertySource(%22classpath:application.properties%22)) 加载属性文件public class ConstantPropertiesUtil implements InitializingBean {@Value("${aliyun.oss.file.endpoint}")private String endpoint;@Value("${aliyun.oss.file.keyid}")private String keyId;@Value("${aliyun.oss.file.keysecret}")private String keySecret;@Value("${aliyun.oss.file.filehost}")private String fileHost;@Value("${aliyun.oss.file.bucketname}")private String bucketName;public static String END\_POINT;public static String ACCESS\_KEY\_ID;public static String ACCESS\_KEY\_SECRET;public static String BUCKET\_NAME;public static String FILE\_HOST ;@Overridepublic void afterPropertiesSet() throws Exception {END\_POINT = endpoint;ACCESS\_KEY\_ID = keyId;ACCESS\_KEY\_SECRET = keySecret;BUCKET\_NAME = bucketName;FILE\_HOST = fileHost;}} |

## 在工具类中引用配置常量即可