# 申 明

本文档根据阿里云官方相关文档编写而成,仅作为***成都伟航创达科技有限公司***内部技术交流使用,请勿用做商业用途。

文档编写人：何昆林 杨绍平。

文档编写时间：2017年5月28日。

# 目 录

[申 明 1](#_Toc483726888)

[目 录 2](#_Toc483726889)

[第1章 OSS介绍 1](#_Toc483726890)

[1.1 对象存储 1](#_Toc483726891)

[1.2 功能概述 1](#_Toc483726892)

[1.3 生命周期 3](#_Toc483726893)

[1.4 数据管理 3](#_Toc483726894)

[1.5 数据安全 3](#_Toc483726895)

[1.6 加速与跨域 3](#_Toc483726896)

[1.7 数据处理 3](#_Toc483726897)

[1.8 使用场景 4](#_Toc483726898)

[第2章 使用指南 6](#_Toc483726899)

[2.1 重要概念 6](#_Toc483726900)

[2.2 存储类型 7](#_Toc483726901)

[2.3 OSS控制台 7](#_Toc483726902)

[2.4 补充说明 8](#_Toc483726903)

[第3章 JAVA快速开发 9](#_Toc483726904)

[3.1 概述 9](#_Toc483726905)

[3.2 开发准备 9](#_Toc483726906)

[3.3 快速开发 10](#_Toc483726907)

[3.3.1 创建Bucket 10](#_Toc483726908)

[3.3.2 上传文件 10](#_Toc483726909)

[3.3.3 元信息设置 13](#_Toc483726910)

[3.3.4 上传进度条 15](#_Toc483726911)

[3.3.5 上传回调 16](#_Toc483726912)

[3.3.6 下载文件 17](#_Toc483726913)

[3.3.7 管理文件 17](#_Toc483726914)

[3.3.8 其他 17](#_Toc483726915)

[3.4 API预览 18](#_Toc483726916)

[3.5 视频专区 18](#_Toc483726917)

[第4章 OSS工具示例 19](#_Toc483726918)

[4.1 OSS Utils 19](#_Toc483726919)

# 第1章 OSS介绍

## 1.1 对象存储

阿里云对象存储服务（Object Storage Service，简称 OSS），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。您可以通过调用 API，在任何应用、任何时间、任何地点上传和下载数据，也可以通过 Web 控制台对数据进行简单的管理。

## 1.2 功能概述

以下是阿里OSS所有功能简易列表：

| **功能** | **描述** |
| --- | --- |
| [创建存储空间](https://help.aliyun.com/document_detail/31842.html) | 在上传任何文件到 OSS 之前，您需要首先创建存储空间来存储文件。 |
| [删除存储空间](https://help.aliyun.com/document_detail/31846.html) | 如果您不再需要存储空间，请将其删除以免进一步产生费用。 |
| [修改存储空间读写权限](https://help.aliyun.com/document_detail/31843.html) | OSS 提供权限控制 ACL（Access Control List），您可以在创建存储空间的时候设置相应的 ACL 权限  控制，也可以在创建之后修改 ACL。 |
| [设置静态网站托管](https://help.aliyun.com/document_detail/31872.html) | 将存储空间配置成静态网站托管模式，并通过存储空间域名访问该静态网站。 |
| [设置日志](https://help.aliyun.com/document_detail/31868.html) | 提供自动记录和保存Bucket上所有访问日志的功能。 |
| [设置防盗链](https://help.aliyun.com/document_detail/31869.html) | OSS 是按使用量收费的服务，为了减少您存储于 OSS 的数据被其他人盗链而产生额外费用，OSS 支持  设置基于 HTTP header 中表头字段 referer 的防盗链方法。 |
| [管理域名](https://help.aliyun.com/document_detail/31836.html) | 将自定义域名绑定到OSS外网域名上，实现自定义域名访问存储空间下的文件，也可同时一键配置阿里  云CDN实现加速功能。 |
| [管理跨域资源共享](https://help.aliyun.com/document_detail/31870.html) | OSS 提供 HTML5 协议中的跨域资源共享 CORS 设置，帮助您实现跨域访问。 |
| [设置生命周期](https://help.aliyun.com/document_detail/31863.html) | 定义和管理存储空间内所有对象或对象的某个子集的生命周期。设置生命周期一般用于文件的批量管理和  自动碎片删除等操作。 |
| [设置跨区域复制](https://help.aliyun.com/document_detail/31864.html) | 将源存储空间的数据实时同步到目标存储空间。 |
| [设置回源规则](https://help.aliyun.com/document_detail/31865.html) | 设置回源规则对获取数据的请求以多种方式进行回源读取，满足数据热迁移、特定请求的重定向等需求。 |
| [上传文件](https://help.aliyun.com/document_detail/31848.html) | 您可以上传任意类型文件到存储空间中。 |
| [新建文件夹](https://help.aliyun.com/document_detail/31910.html) | 您可以像管理 Windows 文件夹一样管理 OSS 文件夹。 |
| [搜索文件](https://help.aliyun.com/document_detail/31911.html) | 在存储空间或文件夹中搜索具有相同的名称前缀的文件。 |
| [获取文件访问地址](https://help.aliyun.com/document_detail/31912.html) | 通过获取已上传文件的地址进行文件的分享和下载。 |
| [设置文件HTTP头](https://help.aliyun.com/document_detail/31913.html) | 单个或批量设置文件的 HTTP 头。 |
| [删除文件](https://help.aliyun.com/document_detail/31862.html) | 删除单个文件或批量删除文件。 |
| [删除文件夹](https://help.aliyun.com/document_detail/31915.html) | 删除单个文件夹或批量删除文件夹。 |
| [修改文件读写权限](https://help.aliyun.com/document_detail/52284.html) | OSS 提供权限控制 ACL（Access Control List），您可以在上传文件的时候设置相应的 ACL 权限控制，  也可以在上传之后修改 ACL。 |
| [碎片管理](https://help.aliyun.com/document_detail/31916.html) | 删除存储空间内的全部或部分碎片文件。 |
| [事件通知](https://help.aliyun.com/document_detail/52656.html) | 通过设置事件通知功能，对您所关心的OSS资源上的相关操作及时进行消息通知。 |
| [服务实时监控](https://help.aliyun.com/document_detail/31876.html) | 实时了解OSS服务使用情况，设置相关的报警规则。 |
| [图片服务](https://help.aliyun.com/document_detail/44686.html) | 对保存在OSS上的图片进行格式转换、剪裁、缩放、旋转、水印、样式封装等各种处理。 |
| [API](https://help.aliyun.com/document_detail/31947.html) | 提供 OSS支持的 RESTful API 操作和相关示例。 |
| [SDK](https://help.aliyun.com/document_detail/52834.html) | 提供主流语言 SDK 的开发操作和相关示例。 |

## 1.3 生命周期

**生命周期**，不同于BOS、阿里OSS在这一块介绍的不够详细，提到了通过XML格式配置。

下面是**BOS**(百度对象存储)对于生命周期的控制：

文件在上传之初可能会频繁访问、但是随着时间的推移可能访问会趋于减少直到删除。所以BOS提供了文件的生命周期管理这一重要功能。

BOS提供的生命周期管理支持以下三种功能：

a)归档转“冷”的数据，即转换存储类型从标准存储到低频存储；

b)删除不再需要的数据；

c)清除过期的三步上传数据。

阿里OSS生命周期管理和百度OSS大同小异、需要了解更多细节请参考官方文档及控制台配合理解并使用。

## 1.4 数据管理

官方的说明为用户可以通过管理控制台、各种语言SDK、各类客户端工具对OSS资源进行管理。

## 1.5 数据安全

具体参见<https://help.aliyun.com/document_detail/31867.html?spm=5176.doc31842.6.603.4o6vlY>

## 1.6 加速与跨域

**加速**：OSS作为海量文件存储源，静态图片、视频文件、下载包、APP更新包等均放在OSS上。通过开启CDN加速并使用OSS作为源站，用户通过CDN节点就近访问文件。

<https://help.aliyun.com/document_detail/31936.html?spm=5176.doc32012.6.629.5Uu1Pw>

**跨域**：跨域资源共享(CORS)允许web端的应用程序访问不属于本域的资源。OSS提供接口方便开发者控制跨域访问的权限

加速和跨域我参考了BOS、相比而言个人认为BOS介绍的比较详细。

## 1.7 数据处理

1：图片处理

OSS支持图片处理，用户将原始图片上传到OSS上，通过调用简单的RESTful接口，实现**对图片的实时处理**，如缩略、裁剪、格式转换、旋转、加文字/图片水印。

2：其他处理

参见官方文档及说明

<https://help.aliyun.com/product/31815.html?spm=5176.doc47605.6.52.Ol8BU3>

## 1.8 使用场景

A）图片与媒体文件，如图1-1所示。

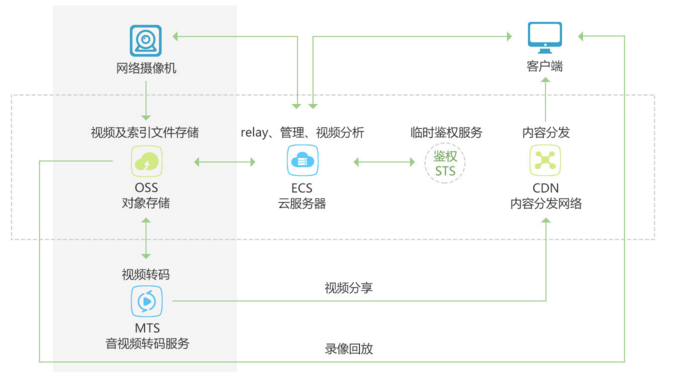


图 1-1

B）网页或者移动应用的静态动态资源分离，如图1-2所示。

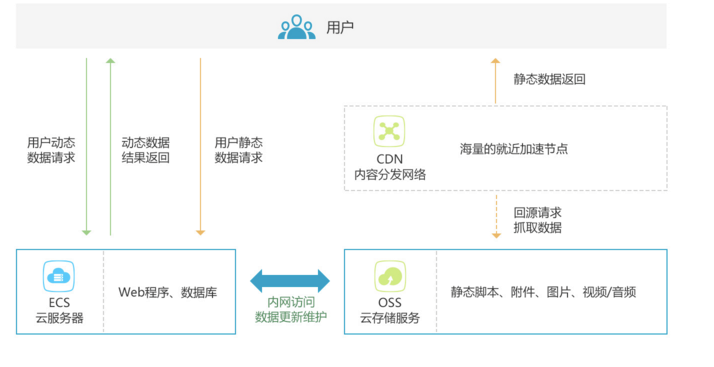
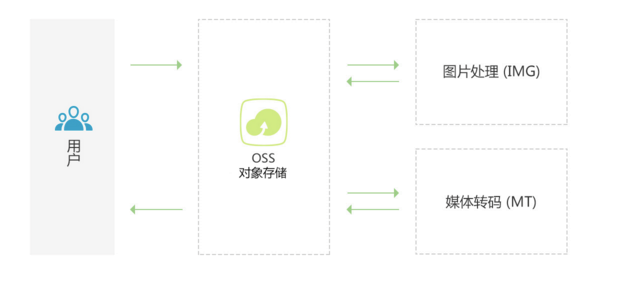


图 1-2

C）云端数据处理，如图1-3所示。

图 1-3

# 第2章 使用指南

## 2.1 重要概念

官方详细说明：

<https://help.aliyun.com/document_detail/31827.html?spm=5176.doc31890.6.566.ulHXYp>

以下是一些关键字：

存储空间（Bucket）

对象/文件（Object）

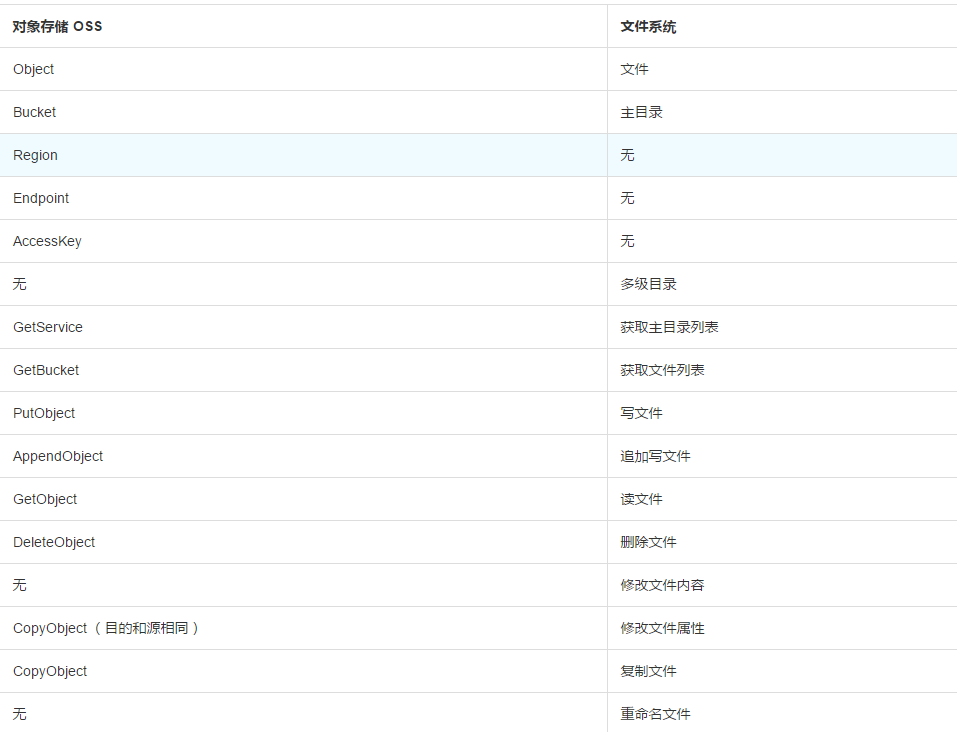
Region（地域）

Endpoint（访问域名）

AccessKey（访问密钥）

强一致性

1）OSS与文件系统的对比



2）OSS术语表



## 2.2 存储类型



## 2.3 OSS控制台

控制台可以实现对OSS的服务管理或者说是对OSS服务的一个从使用到维护升级的一整套解决方案。

详细使用方法参见官方说明，此处不做过多赘述。

## 2.4 补充说明

对于不同的开发者而言，使用OSS的方式可能有些不一致，这里主要讲JAVA如何使用OSS提供的服务。

其他开发环境，如Ptython、PHP、C.HTML#、Android等相关开发请参见官方相关说明。

# 第3章 JAVA快速开发

## 3.1 概述

这里主要详细介绍JAVA使用OSS服务相关步骤和注意事项。

## 3.2 开发准备

传统加入JAR包方式：

1.开发环境要求：Java SDK工具包可在jdk1.6、jdk1.7、jdk1.8环境下运行。

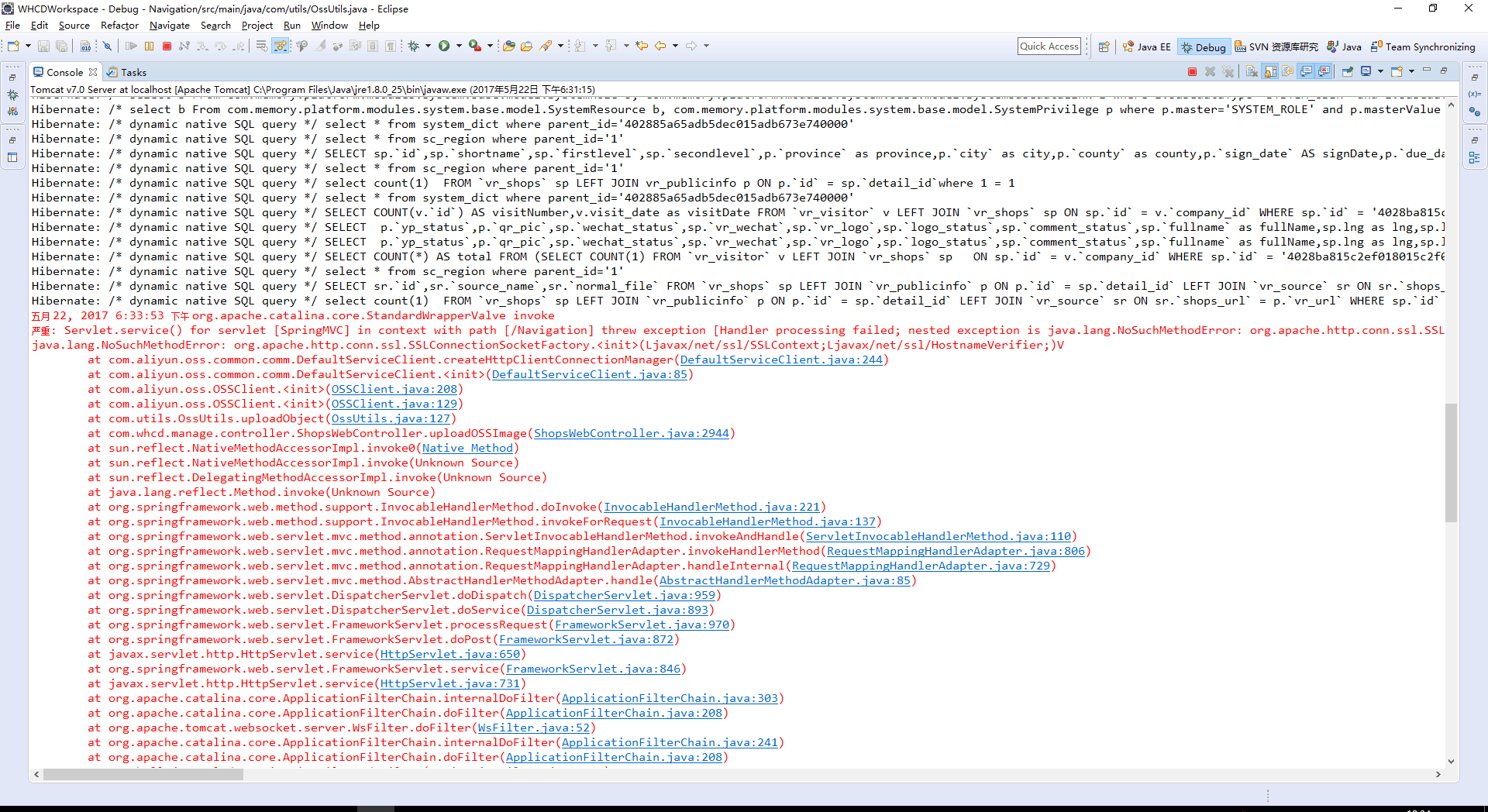
2.下载JAVA SDK并解压，具体有如下JAR包。



3.解压并复制到工程中，如Eclipse右键“工程 -> Properties -> Java Build Path -> Add JARs”。

4.添加SDK工具包lib/bce-java-sdk-version.jar和第三方依赖工具包third-party/\*.jar。其中，version为版本号。

这里可能会出现一个问题:



原因是httpclient包冲突、解决办法是去掉低版本、保留高版本、更新Maven项目即可。在1.3.8中[常见问题](https://help.aliyun.com/document_detail/32024.html?spm=5176.doc32023.6.669.7FL3jn)有详细说明

**对比阿里云:**对于大多数Maven项目可以直接将Maven包引入即可，如下所示：

*<dependency>*

*<groupId>com.aliyun.oss</groupId>*

*<artifactId>aliyun-sdk-oss</artifactId>*

*<version>2.6.0</version>*

*</dependency>*

## 3.3 快速开发

下面为初始化OSSClient对象时需要的一些些参数、这些在OSS里配置后即可。



### 3.3.1 创建Bucket

<https://help.aliyun.com/document_detail/32012.html?spm=5176.doc31848.6.656.hSkVxS>

### 3.3.2 上传文件

----上传字符串

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 上传字符串

String content = "Hello OSS";

ossClient.putObject("<yourBucketName>", "<yourKey>", new ByteArrayInputStream(content.getBytes()));

// 关闭client

ossClient.shutdown();

----上传字符串

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId,accessKeySecret);

// 上传

byte[] content = "Hello OSS".getBytes();

ossClient.putObject("<yourBucketName>", "<yourKey>", new ByteArrayInputStream(content));

// 关闭client

ossClient.shutdown();

----上传网络流

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 上传

InputStream inputStream = new URL("https://www.aliyun.com/").openStream();

ossClient.putObject("<yourBucketName>", "<yourKey>", inputStream);

// 关闭client

ossClient.shutdown();

----上传文件流

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 上传文件流

InputStream inputStream = new FileInputStream("localFile");

ossClient.putObject("<yourBucketName>", "<yourKey>", inputStream);

// 关闭client

ossClient.shutdown();

----上传本地文件

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 上传文件

ossClient.putObject("<yourBucketName>", "<yourKey>", new File("localFile"));

// 关闭client

ossClient.shutdown();

### 3.3.3 元信息设置

以下为常用的Http Header



**示例：**

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

String content = "Hello OSS";

// 创建上传Object的Metadata

ObjectMetadata meta = new ObjectMetadata();

// 设置上传文件长度

meta.setContentLength(content.length());

// 设置上传MD5校验

String md5 = BinaryUtil.toBase64String(BinaryUtil.calculateMd5(content.getBytes()));

meta.setContentMD5(md5);

// 设置上传内容类型

meta.setContentType("text/plain");

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 上传文件

ossClient.putObject("<yourBucketName>", "<yourKey>", new ByteArrayInputStream(content.getBytes()), meta);

// 关闭client

ossClient.shutdown();

**自定义元信息(对Object进行描述)**

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

String content = "Hello OSS";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// 创建上传Object的Metadata

ObjectMetadata meta = new ObjectMetadata();

// 设置自定义元信息name的值为my-data

meta.addUserMetadata("property", "property-value");

// 上传文件

ossClient.putObject("<yourBucketName>", "<yourKey>", new ByteArrayInputStream(content.getBytes()), meta);

// 关闭client

ossClient.shutdown();

**拓展：模拟文件夹**

**说明：**OSS是没有文件夹这个概念的，所有元素都是以Object来存储。创建模拟文件夹本质上来说是创建了一个size为0的Object。对于这个Object可以上传下载，只是控制台会对以”/“结尾的Object以文件夹的方式展示。

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

final String keySuffixWithSlash = "parent\_directory/";

ossClient.putObject("<bucketName>", keySuffixWithSlash, new ByteArrayInputStream(new byte[0]));

// 关闭client

ossClient.shutdown();

### 3.3.4 上传进度条

----上传进度条

完整参考Gihub路径：[GitHub](https://github.com/aliyun/aliyun-oss-java-sdk/blob/master/src/samples/GetProgressSample.java?spm=5176.doc32013.2.11.v4b7FM&file=GetProgressSample.java)

static class PutObjectProgressListener implements ProgressListener {

private long bytesWritten = 0;

private long totalBytes = -1;

private boolean succeed = false;

@Override

public void progressChanged(ProgressEvent progressEvent) {

long bytes = progressEvent.getBytes();

ProgressEventType eventType = progressEvent.getEventType();

switch (eventType) {

case TRANSFER\_STARTED\_EVENT:

System.out.println("Start to upload......");

break;

case REQUEST\_CONTENT\_LENGTH\_EVENT:

this.totalBytes = bytes;

System.out.println(this.totalBytes + " bytes in total will be uploaded to OSS");

break;

case REQUEST\_BYTE\_TRANSFER\_EVENT:

this.bytesWritten += bytes;

if (this.totalBytes != -1) {

int percent = (int)(this.bytesWritten \* 100.0 / this.totalBytes);

System.out.println(bytes + " bytes have been written at this time, upload progress: " + percent + "%(" + this.bytesWritten + "/" + this.totalBytes + ")");

} else {

System.out.println(bytes + " bytes have been written at this time, upload ratio: unknown" + "(" + this.bytesWritten + "/...)");

}

break;

case TRANSFER\_COMPLETED\_EVENT:

this.succeed = true;

System.out.println("Succeed to upload, " + this.bytesWritten + " bytes have been transferred in total");

break;

case TRANSFER\_FAILED\_EVENT:

System.out.println("Failed to upload, " + this.bytesWritten + " bytes have been transferred");

break;

default:

break;

}

}

public boolean isSucceed() {

return succeed;

}

}

public static void main(String[] args) {

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

String accessKeyId = "<accessKeyId>";

String accessKeySecret = "<accessKeySecret>";

String bucketName = "<bucketName>";

String key = "object-get-progress-sample";

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

try {

// 带进度条的上传

ossClient.putObject(new PutObjectRequest(bucketName, key, new FileInputStream("<yourLocalFile>")).

<PutObjectRequest>withProgressListener(new PutObjectProgressListener()));

} catch (Exception e) {

e.printStackTrace();

}

ossClient.shutdown();

}

### 3.3.5 上传回调

----上传回调

完整参考Gihub路径： [GitHub](https://github.com/aliyun/aliyun-oss-java-sdk/blob/master/src/samples/CallbackSample.java?spm=5176.doc32013.2.13.JTfGjg&file=CallbackSample.java)

// endpoint以杭州为例，其它region请按实际情况填写

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// accessKey请登录https://ak-console.aliyun.com/#/查看

String accessKeyId = "<yourAccessKeyId>";

String accessKeySecret = "<yourAccessKeySecret>";

String bucketName = "<yourBucketName>";

// 您的回调服务器地址，如http://oss-demo.aliyuncs.com或http://127.0.0.1:9090

String callbackUrl = "<yourCallbackServerUrl>";

// 创建OSSClient实例

OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

String content = "Hello OSS";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, "key",

new ByteArrayInputStream(content.getBytes()));

// 上传回调参数

Callback callback = new Callback();

callback.setCallbackUrl(callbackUrl);

callback.setCallbackHost("oss-cn-hangzhou.aliyuncs.com");

callback.setCallbackBody("{\\\"mimeType\\\":${mimeType},\\\"size\\\":${size}}");

callback.setCallbackBodyType(CallbackBodyType.JSON);

callback.addCallbackVar("x:var1", "value1");

callback.addCallbackVar("x:var2", "value2");

putObjectRequest.setCallback(callback);

PutObjectResult putObjectResult = ossClient.putObject(putObjectRequest);

// 读取上传回调返回的消息内容

byte[] buffer = new byte[1024];

putObjectResult.getCallbackResponseBody().read(buffer);

// 一定要close，否则会造成连接资源泄漏

putObjectResult.getCallbackResponseBody().close();

// 关闭client

ossClient.shutdown();

### 3.3.6 下载文件

请参考官方说明文件：[OSS文件下载](https://help.aliyun.com/document_detail/32014.html?spm=5176.doc32014.6.658.uwuf4X)

### 3.3.7 管理文件

请参考官方说明文件：[OSS管理文件](https://help.aliyun.com/document_detail/32015.html?spm=5176.doc32014.6.659.so8VUk)

### 3.3.8 其他

[授权访问](https://help.aliyun.com/document_detail/32016.html?spm=5176.doc32015.6.660.w0vqrh)

[生命周期管理](https://help.aliyun.com/document_detail/32017.html?spm=5176.doc32016.6.661.01KFee)

[跨域资源共享](https://help.aliyun.com/document_detail/32018.html?spm=5176.doc32017.6.662.07qPop)

[设置访问日志](https://help.aliyun.com/document_detail/32019.html?spm=5176.doc32018.6.663.LHyvl2)

[静态网站托管](https://help.aliyun.com/document_detail/32020.html?spm=5176.doc32019.6.664.zO05Di)

[防盗链](https://help.aliyun.com/document_detail/32021.html?spm=5176.doc32020.6.665.4StphG)

[跨区域复制](https://help.aliyun.com/document_detail/32022.html?spm=5176.doc32021.6.666.oCrhps)

[图片处理](https://help.aliyun.com/document_detail/47505.html?spm=5176.doc32022.6.667.qrpfmJ)

[异常处理](https://help.aliyun.com/document_detail/32023.html?spm=5176.doc47505.6.668.BqNS9o)

[常见问题](https://help.aliyun.com/document_detail/32024.html?spm=5176.doc32023.6.669.7FL3jn)

## 3.4 API预览

OSS常见API参考：[OSS API](https://help.aliyun.com/document_detail/31948.html?spm=5176.doc31947.6.832.CUrGAA)

## 3.5 视频专区

[视频专区](https://help.aliyun.com/video_list/39687.html?spm=5176.doc31948.6.1062.C6bLcl)

# 第4章 OSS工具示例

## 4.1 OSS Utils

package com.utils;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.InputStream;

import java.net.URISyntaxException;

import java.net.URL;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import java.util.UUID;

import org.springframework.web.multipart.MultipartFile;

import com.aliyun.oss.HttpMethod;

import com.aliyun.oss.OSSClient;

import com.aliyun.oss.model.GeneratePresignedUrlRequest;

import com.aliyun.oss.model.ObjectMetadata;

public class OssUtils {

/\* private static final String EDP\_POINT = "oss-cn-shenzhen.aliyuncs.com"; //OSS外网域名

private static final String ACCESS\_KEY\_ID = "xhXdQd0RR8PBpA1X"; //AccessKey 获取方法文档中有说明

private static final String ACCESS\_KEY\_ID\_SCRECT = "5FqhYwIMMLUGSErWLESmPQC7atFVGn"; //ScrectKey 获取方法文档中有说明

private static final String BUCKET\_NAME = "wehang-qj"; //空间名 创建空间时自定义的名称

\*/

private static final String EDP\_POINT = "oss-cn-shenzhen.aliyuncs.com"; //OSS外网域名

private static final String ACCESS\_KEY\_ID = "LTAIBu1n6STvMIym"; //AccessKey 获取方法文档中有说明

private static final String ACCESS\_KEY\_ID\_SCRECT = "q5PIq6KivS2HTXnl3TJiOyTdTRQ1mW"; //ScrectKey 获取方法文档中有说明

private static final String BUCKET\_NAME = "yangshaoping"; //空间名 创建空间时自定义的名称

public static void main(String[] args) throws FileNotFoundException {

System.out.println(ImageSample("0bd162d9f2d3572c21a659248d13632763d0c386.jpg", "image/resize,w\_10", 50));

}

/\*\*

\* 图片处理后的访问路径

\* @author yangshaoping 2017年5月20日 下午5:02:43

\* @param key

\* @param style 图片处理样式 https://help.aliyun.com/document\_detail/44688.html?spm=5176.doc44687.6.932.taR4en

\* @param monutes 过期时间 分钟数

\* @return

\*/

public static String ImageSample(String key, String style, int monutes) {

OSSClient ossClient = new OSSClient(EDP\_POINT, ACCESS\_KEY\_ID, ACCESS\_KEY\_ID\_SCRECT);

// 过期时间

Date expiration = new Date(new Date().getTime() + 1000 \* 60 \* monutes);

GeneratePresignedUrlRequest req = new GeneratePresignedUrlRequest(BUCKET\_NAME, key, HttpMethod.GET);

req.setExpiration(expiration); //设置过期时间

req.setProcess(style); //设置图片处理样式

URL signedUrl = ossClient.generatePresignedUrl(req);

ossClient.shutdown();

return signedUrl.toString();

}

/\*\*

\* 根据key获取图片路径

\* @author yangshaoping 2017年5月20日 下午3:22:51

\* @param key

\* @return

\*/

public static String getResourceUrl(String key) {

OSSClient client = new OSSClient(EDP\_POINT, ACCESS\_KEY\_ID, ACCESS\_KEY\_ID\_SCRECT);

Date expiration = new Date(new Date().getTime() + 36000 \* 1000); // URL过期时间设置为10个小时

URL url = client.generatePresignedUrl(BUCKET\_NAME, key, expiration);

try {

return url.toURI().toString();

} catch (URISyntaxException e) {

return null;

} finally {

client.shutdown();

}

}

/\*\*

\* 上传文件 根据图片绝对路径上传（本地测试使用）

\* @author yangshaoping 2017年5月20日 下午5:08:39

\* @param filePath

\* @return

\* @throws FileNotFoundException

\*/

public static String putObject(String filePath) throws FileNotFoundException {

// 初始化OSSClient

OSSClient client = new OSSClient(EDP\_POINT, ACCESS\_KEY\_ID, ACCESS\_KEY\_ID\_SCRECT);

// 获取指定文件的输入流

File file = new File(filePath);

InputStream content = new FileInputStream(file);

// 创建上传Object的Metadata

ObjectMetadata meta = new ObjectMetadata();

// 必须设置ContentLength

meta.setContentLength(file.length());

// 上传Object.

client.putObject(BUCKET\_NAME, file.getName(), content, meta);

// 打印ETag

// System.out.println(result.getETag());

// System.out.println(getResourceUrl(file.getName()));

client.shutdown();

return file.getName();

}

/\*\*

\* 上传文件

\* @author yangshaoping 2017年5月20日 下午3:23:35

\* @param file

\* @return

\* @throws Exception

\*/

public static Map<String, Object> uploadObject(MultipartFile file) throws Exception {

// 初始化OSSClient

OSSClient client = new OSSClient(EDP\_POINT, ACCESS\_KEY\_ID, ACCESS\_KEY\_ID\_SCRECT);

// 获取指定文件的输入流

InputStream content = file.getInputStream();

// 创建上传Object的Metadata

ObjectMetadata meta = new ObjectMetadata();

// 必须设置ContentLength

meta.setContentLength(file.getSize());

String fileName = file.getOriginalFilename();

String fileType = fileName.substring(fileName.lastIndexOf(".")+1);

String newFileName = UUID.randomUUID().toString().replaceAll("-", "")+"."+fileType;

// 上传Object.

client.putObject(BUCKET\_NAME, newFileName, content, meta);

Map<String, Object> date = new HashMap<String, Object>();

date.put("key", newFileName);

date.put("url", getResourceUrl(newFileName));

// 打印ETag

// System.out.println(result.getETag());

// System.out.println(getResourceUrl(newFileName));

client.shutdown();

return date;

}

}