

ALIBABA CLOUD

阿里云

对象存储 OSS
快速入门

文档版本：20220210

 阿里云

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.开始使用OSS	05
2.控制台快速入门	06
2.1. 控制台使用流程	06
2.2. 创建存储空间	06
2.3. 上传文件	08
2.4. 下载文件	10
2.5. 分享文件	11
2.6. 使用自有域名访问OSS资源	12
3.命令行工具ossutil快速入门	14
4.图形化管理工具ossbrowser快速入门	17
5.Java SDK快速入门	18
6.后续指引	24

1.开始使用OSS

阿里云对象存储OSS（Object Storage Service）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、音视频在内的各类数据文件。

使用控制台

您可以通过OSS控制台创建Bucket，并将文件上传至Bucket。上传完成后，将文件（Object）下载至本地或者通过生成签名URL的方式将文件分享给第三方，供其下载或预览。更多信息，请参见[控制台使用流程](#)。

观看以下视频快速了解如何通过控制台使用OSS。

使用命令行管理工具ossutil

ossutil是OSS的命令行工具，支持Windows、Linux、macOS系统。您可以通过ossutil提供的方便、简洁、丰富的Bucket和Object命令管理您的OSS。更多信息，请参见[命令行工具ossutil快速入门](#)。

使用图形化管理工具ossbrowser

ossbrowser是OSS的图形化工具，支持Windows、Linux、macOS系统。您可以通过ossbrowser的图形化界面方便直观地管理Bucket、上传下载Object和文件夹（目录）、简化Policy授权等操作。更多信息，请参见[图形化管理工具ossbrowser快速入门](#)。

ossbrowser是桌面式图形化工具，所以传输速度和性能不如命令行工具ossutil。

使用API和SDK

OSS提供Java、Python、PHP、Go等多种语言的API和SDK包，方便您快速进行二次开发。各语言SDK示例，请参见[OSS SDK示例](#)。各API接口的详细信息，请参见[OSS API文档](#)。

基于OSS的文件系统管理

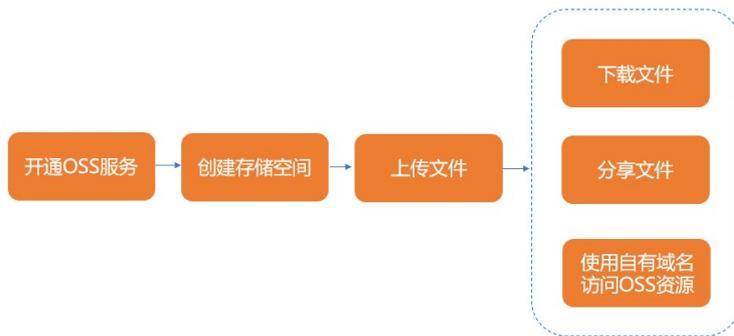
OSS的存储空间内部是扁平的，没有文件系统的目录等概念，所有的对象都直接隶属于其对应的存储空间。如果您想要像使用本地文件夹和磁盘那样来使用OSS存储服务，可以通过配置云存储网关来实现。通过云存储网关提供的NFS、SMB（CIFS）、iSCSI协议，OSS的存储资源会以Bucket为基础映射成本地文件夹或者磁盘。您可以通过文件读写操作访问OSS资源，无缝衔接基于POSIX和块访问协议的应用，降低应用改造和学习成本。更多信息，请参见[通过云存储网关挂载OSS](#)。

2.控制台快速入门

2.1. 控制台使用流程

您可以通过OSS控制台完成常见的基础操作，例如创建存储空间（Bucket）、上传和下载文件（Object）等。

OSS控制台使用流程图如下所示：



1. [开通OSS服务](#)
2. [创建存储空间](#)
3. [上传文件](#)

文件上传至目标Bucket后，您可以执行如下操作：

- 将文件下载至本地。详情请参见[下载文件](#)。
- 通过生成签名URL的方式将文件分享给第三方，供其下载或预览。详情请参见[分享文件](#)。
- 通过自定义域名（自有域名）访问文件，需要将自定义域名绑定至文件所在的Bucket。详情请参见[使用自有域名访问OSS资源](#)。

2.2. 创建存储空间

存储空间（Bucket）是用于存储对象（Object）的容器。在上传任意类型的Object前，您需要先创建Bucket。本文主要介绍如何通过控制台方式创建Bucket。

前提条件

已开通OSS服务。详情请参见[开通OSS服务](#)。

操作步骤

1. 登录[OSS管理控制台](#)。
2. 单击Bucket列表，然后单击创建Bucket。
3. 在创建Bucket面板，按如下说明配置必要参数。其他参数均可保持默认配置，也可以在Bucket创建完成后单独配置。

参数	描述
----	----

参数	描述
Bucket 名称	<p>Bucket 的名称。Bucket 一旦创建，则无法更改其名称。</p> <p>命名规则如下：</p> <ul style="list-style-type: none"> ◦ Bucket 名称必须全局唯一。 ◦ 只能包括小写字母、数字和短划线 (-)。 ◦ 必须以小写字母或者数字开头和结尾。 ◦ 长度必须在3~63字符之间。
地域	<p>Bucket 的数据中心。Bucket 一旦创建，则无法更改其所在地域。</p> <p>如需通过ECS内网访问OSS，请选择与ECS相同的地域。更多信息，请参见OSS访问域名使用规则。</p>
同城冗余存储	<p>OSS同城冗余存储采用多可用区（AZ）机制，将用户的数据以冗余的方式存放在同一地域（Region）的3个可用区。当某个可用区不可用时，仍然能够保障数据的正常访问。</p> <ul style="list-style-type: none"> ◦ 启用：开启同城冗余存储，则Bucket内的Object将以同城冗余的方式进行存储。例如，Bucket存储类型为标准存储，则该Bucket内的Object默认为标准存储（同城冗余）。详情请参见同城冗余存储。 <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> 注意</p> <ul style="list-style-type: none"> ▪ 仅支持开启同城冗余存储。 ▪ 仅允许创建Bucket时开启同城冗余存储。开启后不支持关闭，请谨慎操作。 </div> <ul style="list-style-type: none"> ◦ 关闭：默认不开启同城冗余存储，则Bucket内的Object将以本地冗余的方式进行存储。例如，Bucket存储类型为标准存储，则该Bucket内的Object默认为标准存储（本地冗余）。

4. 单击**确定**。

其他实现方式

操作	实现方式
	命令行工具ossutil
	图形化管理工具ossbrowser
	API接口
	Java SDK
	Python SDK
	Go SDK

创建存储空间	实现方式
	C++ SDK
	PHP SDK
	C SDK
	.NET
	Node.js
	Android
	iOS

后续步骤

[上传文件](#)

2.3. 上传文件

您可以通过OSS控制台上上传不超过5 GB大小的文件。

前提条件

已创建存储空间（Bucket）。详情请参见[创建存储空间](#)。

操作步骤

1. 登录[OSS管理控制台](#)。
2. 单击左侧导航栏的Bucket列表，然后单击目标Bucket名称。
3. 在文件管理页签，单击上传文件。
4. 在上传文件面板，按如下说明配置各项参数。

参数	说明
上传到	设置文件上传到OSS后的存储路径。 <ul style="list-style-type: none">○ 当前目录：将文件上传到当前目录。○ 指定目录：将文件上传到指定目录，您需要输入目录名称。若输入的目录不存在，OSS将自动创建对应的文件夹并将文件上传到该文件夹中。

参数	说明
文件ACL	<p>选择文件的读写权限。</p> <ul style="list-style-type: none"> 继承Bucket：以Bucket读写权限为准。 私有（推荐）：只有文件Owner拥有该文件的读写权限，其他用户没有权限操作该文件。 公共读：文件Owner拥有该文件的读写权限，其他用户（包括匿名访问者）都可以对文件进行访问，这有可能造成您数据的外泄以及费用激增，请谨慎操作。 公共读写：任何用户（包括匿名访问者）都可以对文件进行访问，并且向该文件写入数据。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。 <p>有关文件ACL的更多信息，请参见Object ACL。</p>
待上传文件	<p>选择您需要上传的文件或文件夹。</p> <p>您可以单击扫描文件或扫描文件夹选择本地文件或文件夹，或者直接拖拽目标文件或文件夹到待上传文件区域。</p> <p>如果上传文件夹中包含了无需上传的文件，请单击目标文件右侧的移除将其移出文件列表。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> 注意</p> <ul style="list-style-type: none"> 如果上传的文件与存储空间中已有的文件重名，则会覆盖已有文件。 使用拖拽方式上传文件夹时，OSS会保留文件夹内的所有文件和子文件夹。 文件上传过程中，请勿刷新或关闭页面，否则上传任务会被中断且列表会被清空。 </div>

5. 单击上传文件。

此时，您可以在[上传列表](#)页签查看各个文件的上传进度。上传完成后，您可以在目标路径下查看上传文件的文件名、文件大小以及存储类型等信息。

其他实现方式

操作	实现方式
	命令行工具ossutil
	图形化管理工具ossbrowser
	API接口
	Java SDK
	Python SDK
	Go SDK
	C++ SDK

操作文件	实现方式
	C SDK
	PHP SDK
	.NET SDK
	Node.js SDK
	Browser.js
	Android SDK
	iOS SDK

后续步骤

- 您可以将已上传的文件下载至浏览器默认路径或者本地指定路径。更多信息，请参见[下载文件](#)。
- 您可以将文件URL分享给第三方，供其下载或预览。更多信息，请参见[分享文件](#)。
- 您可以对存储在OSS中的数据进行分析处理。更多信息，请参见[数据处理介绍](#)。
- 在文件上传到OSS上后，您还可以通过上传回调向指定的应用服务器发起回调请求。更多信息，请参见[上传回调](#)。

2.4. 下载文件

当文件（Object）上传至存储空间（Bucket）后，您可以将文件下载至浏览器默认路径或本地指定路径。本文主要介绍如何通过控制台方式下载文件。

前提条件

对于归档或者冷归档存储类型的Object，需要确认Object处于“解冻”状态才能对其进行下载。

操作步骤

1. 登录[OSS管理控制台](#)。
2. 单击左侧导航栏的Bucket列表，然后单击目标Bucket名称。
3. 单击左侧导航栏的文件管理，下载单个或多个文件。
 - 下载单个文件
 - 方式一：单击目标文件右侧的更多 > 下载。
 - 方式二：单击目标文件的文件名或其右侧的详情，在弹出的详情面板中单击下载。
 - 下载多个文件
 - 选中多个文件，选择批量操作 > 下载。通过OSS控制台可一次批量下载最多100个文件。

其他实现方式

操作	实现方式
	命令行工具ossutil

操作	实现方式
下载文件	图形化管理工具ossbrowser
	API接口
	Java SDK
	Python SDK
	PHP SDK
	Go SDK
	C++ SDK
	C SDK
	Node.js
	Browser.js
	Android SDK
	iOS SDK

2.5. 分享文件

文件（Object）上传至存储空间（Bucket）后，您可以将文件URL分享给第三方，供其下载或预览。本文主要介绍如何通过控制台方式分享文件。

操作步骤

1. 登录[OSS管理控制台](#)。
2. 单击左侧导航栏的[Bucket列表](#)，然后单击目标Bucket名称。
3. 单击左侧导航栏的[文件管理](#)，然后单击目标文件的文件名或其右侧的[详情](#)。
4. 在[详情面板](#)，单击[复制文件URL](#)。

如果您要分享私有文件，则在获取文件URL时还需要设置[过期时间](#)。默认的过期时间为3600秒（1小时），最大值为32400秒（9小时）。如果您希望获取更长时效的文件URL，请使用[命令行工具ossutil](#)、[图形化工具ossbrowser](#)或[OSS SDK](#)。

通过文件URL预览

将文件URL分享给第三方时，如需确保第三方访问文件时是预览行为，您需要绑定自定义域名并添加CNAME记录。详情请参见[使用自有域名访问OSS资源](#)。

通过文件URL下载

将文件URL分享给第三方时，如需确保第三方访问文件时是下载行为，您需要将文件HTTP头中的Content-Disposition字段设置为 `attachment`。详情请参见[设置文件元信息](#)。

其他实现方式

操作	实现方式
分享文件	API接口
	Java SDK
	Python SDK
	PHP SDK
	Go SDK
	C++ SDK
	C SDK
	Node.js SDK
	Browser.js SDK

2.6. 使用自有域名访问OSS资源

文件（Object）上传至存储空间（Bucket）后，OSS会自动生成文件URL，您可以通过文件URL（即Bucket外网访问域名）访问该文件。若您希望通过自定义域名（自有域名）访问这些文件，需要将自定义域名绑定至文件所在的Bucket。

前提条件

已完成域名备案，详情请参见[备案](#)。

操作步骤

1. 绑定自定义域名。
 - i. 登录[OSS管理控制台](#)。
 - ii. 单击Bucket列表，之后单击目标Bucket名称。
 - iii. 单击传输管理 > 域名管理。
 - iv. 单击绑定域名。
 - v. 在绑定域名面板，输入要绑定的域名。

若提示域名冲突，表示该域名已绑定至其他Bucket。此时，您可以更换域名或通过验证域名所有权强制绑定域名。验证域名所有权会解除域名与其他Bucket的绑定关系。详情请参见[验证域名所有权](#)。

2. 添加CNAME记录。
 - o. 如果添加的域名为当前账号下管理的域名，开启自动添加CNAME记录。
 - a. 在绑定域名面板，打开自动添加CNAME记录开关。

 **注意** 若您绑定的域名已配置过CNAME，则自动添加的CNAME记录会覆盖原有的CNAME记录。

- b. 单击提交。

- 如果添加的域名为非当前账号下的域名，手动添加CNAME记录。

若您的域名为非阿里云托管的域名，需在对应的域名解析商处配置云解析，如腾讯云解析（原DNSPod）或新网，详情请参见[配置CNAME](#)。

此处以非当前账号下阿里云托管的域名为例，手动添加CNAME记录步骤如下：

- 登录[云解析DNS控制台](#)。
- 在域名解析列表中，单击目标域名右侧的解析设置。
- 单击添加记录，填写域名解析信息。

参数	说明
记录类型	选择域名指向的类型。此处选择CNAME。
主机记录	根据域名前缀填写主机记录。 <ul style="list-style-type: none"> ■ 如果是顶级域名，例如 <code>aliyun.com</code>，输入@。 ■ 如果是二级域名，输入二级域名的前缀。例如域名为 <code>help.aliyun.com</code>，输入help。 ■ 如果需要所有的二级域名都指向Bucket外网访问域名，输入*。
解析线路	解析域名时使用的线路。建议选择默认，系统将自动选择最佳线路。
记录值	填写Bucket外网访问域名。Bucket外网访问域名结构为 <code>BucketName.Endpoint</code> ，例如华东1（杭州）地域创建了名为 <code>examplebucket</code> 的存储空间，外网Endpoint为 <code>oss-cn-hangzhou.aliyuncs.com</code> ，则填写为 <code>examplebucket.oss-cn-hangzhou.aliyuncs.com</code> 。
TTL	域名的更新周期，保留默认值即可。

- 单击确认。

新增CNAME记录实时生效，修改CNAME记录最多72小时内生效。

3. 通过自有域名访问OSS资源。

绑定自定义域名后，文件URL的格式为 `https://YourDomainName/ObjectName`。

例如您在华东1（杭州）地域下创建了目标存储空间examplebucket，并在examplebucket中存放了exampleobject.jpg的文件，自定义域名为 `example.com`，此时您可以使用 `https://example.com/exampleobject.jpg` 访问目标文件。

3. 命令行工具ossutil快速入门

本文旨在引导您通过命令行工具ossutil快速创建目标存储空间（Bucket），然后将本地文件上传至Bucket。上传完成后，将文件（Object）下载至本地或者通过生成签名URL的方式将文件分享给第三方，供其下载或预览。

前提条件

已安装ossutil。详情请参见[下载和安装](#)。

注意事项

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。例如对于Windows 64位系统，请将./ossutil64替换成ossutil64.exe。各系统对应的Binary名称如下。

系统	Binary名称
Linux 64位	./ossutil64
Linux 32位	./ossutil32
Windows 64位	ossutil64.exe
Windows 32位	ossutil32.exe
macOS 64位	./ossutilmac64
macOS 32位	./ossutilmac32
ARM 64位	./ossutilarm64
ARM 32位	./ossutilarm32

创建Bucket

- 命令格式

```
./ossutil64 mb oss://bucket
```

- 使用示例

创建名为examplebucket的存储空间。

```
./ossutil64 mb oss://examplebucket
```

以下输出结果表明已成功创建examplebucket。

```
0.668238(s) elapsed
```

有关创建Bucket的更多示例，请参见[mb（创建存储空间）](#)。

上传文件

- 命令格式

```
./ossutil64 cp local_file oss://bucket
```

- 使用示例
 - 上传单个文件examplefile.txt至目标存储空间examplebucket。

```
./ossutil64 cp examplefile.txt oss://examplebucket
```

- 上传单个文件examplefile.txt至目标存储空间examplebucket，并将文件重命名为exampleobject.txt。

```
./ossutil64 cp examplefile.txt oss://examplebucket/exampleobject.txt
```

以下输出结果表明文件已成功上传至目标Bucket。

```
0.720812(s) elapsed
```

有关上传文件的更多示例，请参见[cp（上传文件）](#)。

下载文件

- 命令格式

```
./ossutil64 cp cloud_url local_file
```

- 使用示例

将文件examplefile.txt从目标存储空间examplebucket下载至本地localfolder文件夹下。

```
./ossutil64 cp oss://examplebucket/examplefile.txt localfolder/
```

将文件examplefile.txt从目标存储空间examplebucket下载至本地localfolder文件夹下，并将文件重命名为exampleobject.txt。

```
./ossutil64 cp oss://examplebucket/examplefile.txt localfolder/exampleobject.txt
```

以下输出结果表明文件已成功下载至本地目标文件夹。

```
0.720812(s) elapsed
```

有关下载文件的更多示例，请参见[cp（下载文件）](#)。

生成签名URL

- 命令格式

```
./ossutil64 sign cloud_url [--timeout <value>]
```

- 使用示例

对目标文件 `oss://examplebucket/exampleobject.txt` 生成超时时间为3600秒的文件签名URL。

```
./ossutil64 sign oss://examplebucket/exampleobject.txt --timeout 3600
```

以下输出结果表明已成功生成文件签名URL。

```
https://examplebucket.ss-cn-hangzhou.aliyuncs.com/exampleobject.txt?Expires=1608282224&OSSAccessKeyId=LTAI4G33piUmgRN1DXx9****&Signature=jo4%2FGykfuc1A4fvyyvKRpRyymYH****  
0.368676(s) elapsed
```

有关生成签名URL的更多示例，请参见[sign（生成签名URL）](#)。

4. 图形化管理工具ossbrowser快速入门

ossbrowser是阿里云官方提供的OSS图形化管理工具，提供类似Windows资源管理器的功能。使用ossbrowser，您可以快速完成存储空间（Bucket）和文件（Object）的相关操作。

前提条件

已安装并登录ossbrowser。具体操作，请参见[安装并登录ossbrowser](#)。

Bucket或Object相关操作

ossbrowser支持的Bucket或Object级别的操作与控制台支持的操作类似，请按照ossbrowser界面指引完成Bucket或Object相关操作。

分类	配置说明
Bucket相关操作	<ul style="list-style-type: none">• 创建Bucket Bucket是您用于存储Object的容器。在上传任何文件到OSS之前，您必须先创建存储空间。有关创建Bucket时如何填写Bucket名称、选择所在地域、ACL权限和存储类型信息，请参见创建存储空间。• 删除Bucket 如果您不再需要Bucket，请将其删除，以免产生额外费用。有关删除Bucket的注意事项，请参见删除存储空间。
Object相关操作	<p>ossbrowser支持上传、下载、预览、移动或复制文件、分享文件等操作。操作过程中有如下注意事项：</p> <ul style="list-style-type: none">• ossbrowser默认使用分片上传和断点续传上传文件，上传文件最大不能超过48.8 TB。若您因意外中断了文件上传的过程，且未继续完成该文件的上传，则已上传的部分会以碎片（Part）的形式存储在OSS的存储空间（Bucket）中。若您不再需要这些Part，建议您通过以下方式删除，以免产生额外的存储费用。<ul style="list-style-type: none">◦ 手动删除Part的具体操作，请参见删除碎片。◦ 通过生命周期规则自动删除Part的具体操作，请参见设置生命周期规则。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 10px 0;"><p> 注意 如果上传的文件与Bucket中已有的文件重名，则覆盖已有文件。</p></div> <ul style="list-style-type: none">• 移动或复制文件最大不能超过5 GB，5 GB以上文件的移动或复制操作建议使用ossutil。• 文件上传至Bucket后，您可以将文件URL分享给第三方，供其下载或预览。具体操作，请参见分享文件。 <p>Object各类操作的其他注意事项，请参见控制台用户指南对应文档。</p>

5.Java SDK快速入门

本文介绍如何快速使用OSS Java SDK完成常见操作，如创建存储空间（Bucket）、上传文件（Object）、下载文件等。

前提条件

已安装Java SDK。详情请参见[安装](#)。

示例工程

OSS Java SDK提供了基于Maven和Ant的示例工程。您可以在本地设备上编译和运行示例工程，或者以示例工程为基础开发您的应用。工程的编译和运行方法，请参见工程目录下的README.md。

- Maven示例工程：[aliyun-oss-java-sdk-demo-mvn.zip](#)
- Ant示例工程：[aliyun-oss-java-sdk-demo-ant.zip](#)

创建存储空间

存储空间是OSS的全局命名空间，相当于数据的容器，可以存储若干文件。

以下代码用于创建存储空间。

```
public class Demo {
    public static void main(String[] args) throws Exception {
        // Endpoint以华东1（杭州）为例，其它Region请按实际情况填写。关于其他Region对应的Endpoint
        // 信息，请参见访问域名和数据中心。
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // 阿里云账号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用户进行API访
        // 问或日常运维，请登录RAM控制台创建RAM用户。
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // 填写Bucket名称，例如examplebucket。
        String bucketName = "examplebucket";
        // 创建OSSClient实例。
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret
    );

    try {
        // 创建存储空间。
        ossClient.createBucket(bucketName);
    } catch (OSSEException oe) {
        System.out.println("Caught an OSSEException, which means your request made it to
OSS, "
            + "but was rejected with an error response for some reason.");
        System.out.println("Error Message:" + oe.getErrorMessage());
        System.out.println("Error Code:" + oe.getErrorCode());
        System.out.println("Request ID:" + oe.getRequestId());
        System.out.println("Host ID:" + oe.getHostId());
    } catch (ClientException ce) {
        System.out.println("Caught an ClientException, which means the client encounter
ed "
            + "a serious internal problem while trying to communicate with OSS, "
            + "such as not being able to access the network.");
        System.out.println("Error Message:" + ce.getMessage());
    } finally {
        if (ossClient != null) {
            ossClient.shutdown();
        }
    }
}
}
```

上传文件

以下代码用于通过流式上传的方式将文件上传到OSS。

```

import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import java.io.ByteArrayInputStream;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Endpoint以华东1（杭州）为例，其它Region请按实际情况填写。
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // 阿里云账号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用户进行API访问或日常运维，请登录RAM控制台创建RAM用户。
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // 填写Bucket名称，例如examplebucket。
        String bucketName = "examplebucket";
        // 填写Object完整路径，例如exampledir/exampleobject.txt。Object完整路径中不能包含Bucket名称。
        String objectName = "exampledir/exampleobject.txt";
        // 创建OSSClient实例。
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
    };
    try {
        String content = "Hello OSS";
        ossClient.putObject(bucketName, objectName, new ByteArrayInputStream(content.getBytes()));
    } catch (OSSException oe) {
        System.out.println("Caught an OSSException, which means your request made it to OSS, "
            + "but was rejected with an error response for some reason.");
        System.out.println("Error Message:" + oe.getErrorMessage());
        System.out.println("Error Code:" + oe.getErrorCode());
        System.out.println("Request ID:" + oe.getRequestId());
        System.out.println("Host ID:" + oe.getHostId());
    } catch (ClientException ce) {
        System.out.println("Caught an ClientException, which means the client encountered "
            + "a serious internal problem while trying to communicate with OSS, "
            + "such as not being able to access the network.");
        System.out.println("Error Message:" + ce.getMessage());
    } finally {
        if (ossClient != null) {
            ossClient.shutdown();
        }
    }
}
}

```

下载文件

以下代码用于通过流式下载方式从OSS下载文件。

```

import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;

```

```

import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.OSSObject;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Endpoint以华东1（杭州）为例，其它Region请按实际情况填写。
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // 阿里云账号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用户进行API访问或日常运维，请登录RAM控制台创建RAM用户。
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // 填写Bucket名称，例如examplebucket。
        String bucketName = "examplebucket";
        // 填写Object完整路径，例如exampledir/exampleobject.txt。Object完整路径中不能包含Bucket名称。
        String objectName = "exampledir/exampleobject.txt";
        // 创建OSSClient实例。
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

        try {
            // 调用ossClient.getObject返回一个OSSObject实例，该实例包含文件内容及文件元信息。
            OSSObject ossObject = ossClient.getObject(bucketName, objectName);
            // 调用ossObject.getObjectContent获取文件输入流，可读取此输入流获取其内容。
            InputStream content = ossObject.getObjectContent();
            if (content != null) {
                BufferedReader reader = new BufferedReader(new InputStreamReader(content));
                while (true) {
                    String line = reader.readLine();
                    if (line == null) break;
                    System.out.println("\n" + line);
                }
                // 数据读取完成后，获取的流必须关闭，否则会造成连接泄漏，导致请求无连接可用，程序无法正常工作。
                content.close();
            }
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to OSS, "
                + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encountered "
                + "a serious internal problem while trying to communicate with OSS, "
                + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}

```

```
        }  
    }  
}
```

列举文件

以下代码用于列举examplebucket存储空间下的文件。默认列举100个文件。

```
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.aliyun.oss.model.OSSObjectSummary;
import com.aliyun.oss.model.ObjectListing;
public class Demo {
    public static void main(String[] args) throws Exception {
        // Endpoint以华东1（杭州）为例，其它Region请按实际情况填写。
        String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
        // 阿里云账号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用户进行API访问或日常运维，请登录RAM控制台创建RAM用户。
        String accessKeyId = "yourAccessKeyId";
        String accessKeySecret = "yourAccessKeySecret";
        // 填写Bucket名称，例如examplebucket。
        String bucketName = "examplebucket";
        // 创建OSSClient实例。
        OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);

        try {
            // ossClient.listObjects返回ObjectListing实例，包含此次listObject请求的返回结果。
            ObjectListing objectListing = ossClient.listObjects(bucketName);
            // objectListing.getObjectSummaries获取所有文件的描述信息。
            for (OSSObjectSummary objectSummary : objectListing.getObjectSummaries()) {
                System.out.println("- " + objectSummary.getKey() + " " +
                    "(size = " + objectSummary.getSize() + ")");
            }
        } catch (OSSException oe) {
            System.out.println("Caught an OSSException, which means your request made it to
OSS, "
                + "but was rejected with an error response for some reason.");
            System.out.println("Error Message:" + oe.getErrorMessage());
            System.out.println("Error Code:" + oe.getErrorCode());
            System.out.println("Request ID:" + oe.getRequestId());
            System.out.println("Host ID:" + oe.getHostId());
        } catch (ClientException ce) {
            System.out.println("Caught an ClientException, which means the client encounter
ed "
                + "a serious internal problem while trying to communicate with OSS, "
                + "such as not being able to access the network.");
            System.out.println("Error Message:" + ce.getMessage());
        } finally {
            if (ossClient != null) {
                ossClient.shutdown();
            }
        }
    }
}
```

6. 后续指引

当您完成了创建存储空间（Bucket）、上传和下载文件（Object）等基本操作后，您可以结合自身的业务场景选用OSS提供的一系列常用的高阶功能。

下表汇总了OSS的常用高阶功能：

功能列表	说明
请求者付费模式	请求者付费模式是指由请求者支付访问Bucket内数据时产生的费用，而Bucket拥有者仅支付存储费用。当您希望共享数据，但又不希望支付因共享数据产生的额外费用时，您可以开启此功能。
生命周期	生命周期规则（Lifecycle）用于定期将Object转储为低频访问、归档存储或冷归档存储类型，或将过期的Object和碎片删除，从而节省存储费用。
静态网站托管	静态网站是指所有的网页都由静态内容构成，包括客户端执行的脚本（例如JavaScript）。您可以将您的静态网站托管到OSS的Bucket，并使用Bucket域名访问该网站。
版本控制	版本控制是针对Bucket级别的数据保护功能。开启版本控制后，针对数据的覆盖和删除操作将会以历史版本的形式保存下来。您在错误覆盖或者删除Object后，能够将Bucket中存储的Object恢复至任意时刻的历史版本。
访问控制	针对存放在Bucket中Object的访问，OSS提供了多种权限控制策略。例如基于资源的授权策略Bucket Policy、基于用户的授权策略RAM Policy、读写权限（ACL）、通过STS临时授权访问OSS以及通过防盗链对访问来源设置白名单。
数据加密	服务器端加密： 上传文件时，OSS对收到的文件进行加密，再将得到的加密文件持久化保存；下载文件时，OSS自动将加密文件解密后返回给用户，并在返回的HTTP请求Header中，声明该文件进行了服务器端加密。 客户端加密： 将文件上传到OSS之前在本地进行加密。
跨区域复制	跨区域复制（Cross-Region Replication）是跨不同OSS数据中心（地域）的Bucket自动、异步（近实时）复制Object，它会将Object的创建、更新和删除等操作从源存储空间复制到不同区域的目标存储空间。跨区域复制功能满足Bucket跨区域容灾或用户数据复制的需求。