

ZOS对象存储Android_SDK使用手册

下载与安装

相关资源

相关资源见帮助中心

SDK安装

(1)将SDK压缩包zos-android-sdk.zip解压到项目根路径

(2)修改项目gradle配置文件，增加对SDK的依赖项

```
dependencies {  
    ..... //其它原依赖项  
    implementation files('zos-android-sdk/aws-android-sdk-s3.jar') // 本次新增  
    依赖项  
}
```

(3)修改项目配置文件AndroidManifest.xml，增加权限

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

初始化连接

在正式使用SDK接口之前，需要先连接ZOS，使用以下的示例可以连接ZOS。直接在APP端代码中硬编码或在配置文件中明文存储Access Key和Secret Key有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，以确保安全。

//创建（初始化）一个zosClient实例

```
import com.amazonaws.*;  
import com.amazonaws.auth.BasicAWSCredentials;  
import com.amazonaws.services.s3.AmazonS3Client;  
import com.amazonaws.services.s3.model.*;  
String accessKey = "accessKey"; // 替换为真实的access_key，安全起见，可在配置文件或者环境变量中密文存放  
String secretKey = "secretKey"; // 替换为真实的secret_key，安全起见，可在配置文件或者环境变量中密文存放  
String endpoint = "endpoint"; // 替换为资源池endpoint  
BasicAWSCredentials credentials = new BasicAWSCredentials(accessKey, secretKey);  
AmazonS3Client client = new AmazonS3Client(credentials);  
client.setEndpoint(endpoint);  
// 如有必要，可以设置Client各种相关配置项  
ClientConfiguration clientConfig = new ClientConfiguration();  
client.setConfiguration(clientConfig);  
// 初始化完毕后，则可以使用Client和ZOS通信，进行各种操作，如上传、下载等  
// 使用完毕后，关闭Client实例,当Client关闭后，不可以继续使用Client。如有需要，则重新初始化一个Client实例  
client.shutdown();
```

桶操作

Create Bucket

功能说明

在指定账号下创建一个新桶。

返回结果

无

方法原型

```
createBucket(String bucketName)

createBucket(CreateBucketRequest createBucketRequest)
```

参数说明

参数方法	描述	参数类型
bucketName	桶名	String
ACL	设置桶CannedAccessControlList，可选项： Private，私有（默认）； PublicRead，公共读，但只有桶所有者可以写入和更改ACL； PublicReadWrite，公共读写，但只有桶所有者可以写入和更改ACL； AuthenticatedRead，对象所有者和存储桶所有者可以读取该对象，只有对象所有者可以写入和更改ACL。	CannedAccessControlList
policy	设置桶冗余策略，可选项： "SINGLE-AZ"，单可用区； "MULTI-AZ"，多可用区。	String
storageClass	设置桶存储类型，可选项： STANDARD，标准存储； STANDARD_IA，低频存储； GLACIER，归档存储。	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    secretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    String storage_calss = "STANDARD_IA";
```

```
CreateBucketRequest request = new CreateBucketRequest(BucketName);
request.setStorageClass(storage_calss);
s3client.createBucket(request);
Log.e("CreateBucket","Success create the bucket");
} catch (Exception e) {
    Log.e(TAG, "e:" + e.getMessage());
}
```

List Buckets

功能说明

获取当前用户所有桶信息

返回结果

参数说明	描述	类型
BucketName	桶名	String

示例

方法原型

```
listBuckets()
```

参数说明

无

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    List<Bucket> buckets = s3client.listBuckets();
    for(Bucket bucket : buckets){
        Log.e("Bucket:", bucket.getName());
    }
}catch (Exception e){
    Log.e("ListBucket","e:" +e.getMessage());
}
```

Head Bucket

功能说明

验证桶是否存在，如果桶不存在打印相关错误日志

返回结果

无

方法原型

```
headBucket(HeadBucketRequest headBucketRequest)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    HeadBucketRequest request = new HeadBucketRequest(BucketName);
    s3client.headBucket(request);
    Log.e("HeadBucket","Success Head the bucket:"+ BucketName );
}catch (Exception e){
    Log.e("HeadBucket","e:" +e.getMessage());
}
```

Delete Bucket

功能说明

删除一个桶

返回结果

无

方法原型

```
deleteBucket(String bucketName)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    s3client.deleteBucket(BucketName);
    Log.e("DeleteBucket","Success Delete the bucket");
} catch (Exception e) {
    Log.e("DeleteBucket", "e:" + e.getMessage());
}
```

Set Bucket Tagging

功能说明

设置桶的Tagging

返回结果

无

方法原型

```
setBucketTaggingConfiguration(String bucketName, BucketTaggingConfiguration
bucketTaggingConfiguration)

setBucketTaggingConfiguration(SetBucketTaggingConfigurationRequest
setBucketTaggingConfigurationRequest)
```

参数说明

参数说明	描述	参数类型
set	桶标签列表	TagSet
request	设置桶标签请求	SetBucketTaggingConfigurationRequest

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    List<TagSet> tagSets = new ArrayList<>();
    TagSet set = new TagSet();
    set.setTag("key1","value1");
    set.setTag("key2","value2");
    tagSets.add(set);
    BucketTaggingConfiguration taggingConfiguration = new
    BucketTaggingConfiguration();
    taggingConfiguration.setTagSets(tagSets);
    SetBucketTaggingConfigurationRequest request = new
    SetBucketTaggingConfigurationRequest(BucketName,taggingConfiguration);
    s3client.setBucketTaggingConfiguration(request);
    Log.e("SetBucketTagging","Success set the tagging");
}catch (Exception e){
    Log.e("SetBucketTagging","e:" +e.getMessage());
}
```

Get Bucket Tagging

功能说明

获取桶标签

返回结果

参数说明	描述	类型
configuration	桶标签配置	BucketTaggingConfiguration
set	桶标签	TagSet

方法原型

```
BucketTaggingConfiguration getBucketTaggingConfiguration(String bucketName)

BucketTaggingConfiguration
getBucketTaggingConfiguration(GetBucketTaggingConfigurationRequest
getBucketTaggingConfigurationRequest)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    BucketTaggingConfiguration configuration =
    s3client.getBucketTaggingConfiguration(BucketName);
    List<TagSet> sets = configuration.getAllTagSets();
    for(TagSet set : sets){
        Log.e("GetBucketTagging","tagging:"+set.toString());
    }
}catch (Exception e){
    Log.e("GetBucketTagging","e:" +e.getMessage());
}
```

Delete Bucket Tagging

功能说明

删除桶标签配置

返回结果

无

方法原型

deleteBucketTaggingConfiguration(String bucketName)

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    s3client.deleteBucketTaggingConfiguration(BucketName);
    Log.e("DeleteBucketTagging","Success delete bucket tagging");
}catch (Exception e){
    Log.e("DeleteBucketTagging","e:" +e.getMessage());
}
```

Set Bucket Logging

功能说明

设置桶日志配置

返回结果

无

方法原型

setBucketLoggingConfiguration(SetBucketLoggingConfigurationRequest
setBucketLoggingConfigurationRequest)

参数说明

参数说明	描述	参数类型
BucketName	桶名	String
DestinationBucketName	生成桶日志目标桶名	String
LogPrefix	生成桶日志目标前缀	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String LogPrefix = "prefix";
    String BucketName = "android";
    String DestinationBucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
```

```
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
BucketLoggingConfiguration configuration = new BucketLoggingConfiguration();
configuration.setDestinationBucketName(DestinationBucketName);
configuration.setLogFilePrefix(LogPrefix);
SetBucketLoggingConfigurationRequest request = new
SetBucketLoggingConfigurationRequest(BucketName,configuration);
s3client.setBucketLoggingConfiguration(request);
Log.e("SetBucketLogging","Success Set Bucket Logging");
}catch (Exception e){
Log.e("SetBucketLogging","e:" +e.getMessage());
}
```

Get Bucket Logging

功能说明

获取桶日志配置

返回结果

参数说明	描述	类型
configuration	桶日志配置配置	BucketLoggingConfiguration

方法原型

```
BucketLoggingConfiguration getBucketLoggingConfiguration(String bucketName)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    BucketLoggingConfiguration configuration =
s3client.getBucketLoggingConfiguration(BucketName);
    Log.e("GetBucketLogging","DestinationBucketName:"+
configuration.getDestinationBucketName()
+","Prefix:"+configuration.getLogFilePrefix());
}catch (Exception e){
    Log.e("GetBucketLogging","e:" +e.getMessage());
}
```

Set Bucket Version

功能说明

设置桶多版本规则

返回结果

无

方法原型

```
setBucketVersioningConfiguration(SetBucketVersioningConfigurationRequest
setBucketVersioningConfigurationRequest)
```

参数说明

参数说明	描述	参数类型
request	设置桶多版本请求	SetBucketVersioningConfigurationRequest
BucketName	桶名	String
Status	多版本状态	String(仅支持：Enabled 开启多版本、Suspended 暂停多版本)
configuration	桶多版本配置	BucketVersioningConfiguration

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
```

```
String BucketName = "android";
String Status = "Enabled";
BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
BucketVersioningConfiguration configuration = new
BucketVersioningConfiguration();
configuration.setStatus(Status);
SetBucketVersioningConfigurationRequest request = new
SetBucketVersioningConfigurationRequest(BucketName,configuration);
s3client.setBucketVersioningConfiguration(request);
Log.e("SetBucketVersion","Success set bucket version");
}catch (Exception e){
Log.e("SetBucketVersion","e:" +e.getMessage());
}
```

Get Bucket Version

功能说明

获取桶多版本状态

返回结果

参数说明	描述	参数类型
configuration	桶多版本配置	BucketVersioningConfiguration
Status	多版本状态	String

方法原型

```
BucketVersioningConfiguration getBucketVersioningConfiguration(String
bucketName)

BucketVersioningConfiguration
getBucketVersioningConfiguration(GetBucketVersioningConfigurationRequest
getBucketVersioningConfigurationRequest)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
String AccessKey = "accessKey";
String SecretKey = "secretKey";
String Endpoint = "endPoint";
String BucketName = "android";
```

```
BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
BucketVersioningConfiguration configuration =
s3client.getBucketVersioningConfiguration(BucketName);
String Status = configuration.getStatus();
Log.e("GetBucketVersion","the Version:"+ Status);
}catch (Exception e){
    Log.e("GetBucketVersion","e:" +e.getMessage());
}
```

Set Bucket ACL

功能说明

设置桶的ACL

返回结果

无

方法原型

```
setBucketAcl(String bucketName, AccessControlList acl)

setBucketAcl(SetBucketAclRequest setBucketAclRequest)

setBucketAcl(String bucketName, CannedAccessControlList acl)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String
CannedACL	桶的访问控制权限	CannedAccessControlList: 支持 Private/PublicRead/PublicReadWrite/AuthenticatedRead

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    CannedAccessControlList CannedACL = PublicRead;
    SetBucketAclRequest request = new SetBucketAclRequest(BucketName,CannedACL);
    s3client.setBucketAcl(request);
    Log.e("SetBucketACL","Success set bucket ACL");
}
```

```
}catch (Exception e){
    Log.e("SetBucketACL","e:" +e.getMessage());
}
```

Get Bucket ACL

功能说明

获取桶的访问控制权限

返回结果

参数说明	描述	参数类型
ACL	桶ACL	AccessControlList

方法原型

```
AccessControlList getBucketAcl(String bucketName)

AccessControlList getBucketAcl(GetBucketAclRequest getBucketAclRequest)
```

参数说明

参数说明	描述	参数类型
BucketName	桶名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    AccessControlList ACL = s3client.getBucketAcl(BucketName);
    Log.e("GetBucketACL","the ACL:"+ ACL.toString());
}catch (Exception e){
    Log.e("GetBucketACL","e:" +e.getMessage());
}
```

对象操作

Put Object

功能说明

将一个对象上传至指定的Bucket。

返回结果

参数名称	参数描述	类型
putObjectRequest	上传请求	PutObjectRequest
bucket	桶名	String
key	对象名	String
exampleString	上传对象内容字符串	String
input	输入流	InputStream
metadata	上传对象的元数据	OjectMetadata

方法原型

```
PutObjectResult putObject(PutObjectRequest putObjectRequest)

PutObjectResult putObject(String bucketName, String key, File file)

PutObjectResult putObject(String bucketName, String key, InputStream input,
ObjectMetadata metadata)

PutObjectResult putObject(String bucketName, String key, String content)
```

参数说明

参数名称	参数描述	类型
versionId	对象版本号	String
eTag	对象eTag	String
metadata	对象元数据	ObjectMetadata
StorageClass	对象存储级别	支持：STANDARD/STANDARD_IA/GLACIER
CannedAcl	对象访问权限	支持： Private/PublicRead/PublicReadWrite/AuthenticatedRead

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
```

```
String BucketName = "android";
String ObjectName = "obj_test";
BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
ObjectMetadata metadata = new ObjectMetadata();
metadata.addUserMetadata("my-metadata-key", "my-metadata-value"); // 自定义元
数据
String content = "Hello, this is a test string.";
byte[] bytes = content.getBytes(StandardCharsets.UTF_8);
InputStream input = new ByteArrayInputStream(bytes);
PutObjectRequest request = new
PutObjectRequest(BucketName,ObjectName,input,metadata);
request.setStorageClass("STANDARD_IA");
request.setCannedAcl(PublicRead);
PutObjectResult result = s3client.putObject(request);
String versionId = result.getVersionId();
String eTag = result.getETag();
Log.e("PubObject", "bucketName:" + BucketName + ",versionId:" + versionId +
",Etag:" + eTag);
} catch (Exception e) {
    Log.e("PutObject", "e:" + e.getMessage());
}
```

Get Object

功能说明

下载对象

返回结果

参数名称	参数描述	类型
objectContent	对象内容	S3ObjectInputStream
metadata	对象元数据	ObjectMetadata

方法原型

```
S3Object getObject(GetObjectRequest getObjectRequest)

S3Object getObject(String bucketName, String key)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象版本号	versionId

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "testfilemeta";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    GetObjectRequest request = new GetObjectRequest(BucketName,ObjectName);
    S3Object object= s3client.getObject(request);
    Log.e("GetObject","object:"+object.toString());
} catch (Exception e) {
    Log.e("GetObject", "e:" + e.getMessage());
}
```

Delete Object

功能说明

删除对象

返回结果

无

方法原型

```
deleteObject(String bucketName, String key)

deleteObject(DeleteObjectRequest deleteObjectRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    secretKey);
```

```
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
s3client.deleteObject(BucketName, ObjectName);
Log.e("DeleteObject", "Success delete Object");
} catch (Exception e) {
    Log.e("DeleteObject", "e:" + e.getMessage());
}
```

Delete Objects

功能说明

删除一批对象

返回结果

参数名称	参数描述	类型
result	删除对象结果	DeleteObjectsResult
object	对象信息	DeleteObjectsResult.DeletedObject

方法原型

```
deleteObjects(DeleteObjectsRequest deleteObjectsRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
keys	删除的对象列表	List<DeleteObjectsRequest.KeyVersion>
Key	删除的对象信息	DeleteObjectsRequest.KeyVersion
ObjectName	删除的对象名	String
versionId	删除的对象版本号	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    DeleteObjectsRequest request = new DeleteObjectsRequest(BucketName);
    List<DeleteObjectsRequest.KeyVersion> keys = new ArrayList<>();
    DeleteObjectsRequest.KeyVersion key1 = new
    DeleteObjectsRequest.KeyVersion("obj1");
```

```
        DeleteObjectsRequest.KeyVersion key2 = new
DeleteObjectsRequest.KeyVersion("obj2");
        keys.add(key1);
        keys.add(key2);
        request.setKeys(keys);
        DeleteObjectsResult result = s3client.deleteObjects(request);
        Log.e("DeleteObjects","Success delete objs");
        for(DeleteObjectsResult.DeletedObject object : result.getDeletedObjects()){
            Log.e("DeleteObjects","object:" + object.getKey());
        }
    } catch (Exception e) {
        Log.e("DeleteObjects", "e:" + e.getMessage());
    }
}
```

Delete Object Version

功能说明

删除一个带有版本号的对象

返回结果

无

方法原型

```
deleteVersion(String bucketName, String key, String versionId)
```

```
deleteVersion(DeleteVersionRequest deleteVersionRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	删除的对象名	String
Version	对象版本号	String

示例

```
try {
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    String Version = "null";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    s3client.deleteVersion(BucketName,ObjectName,Version);
}
```

```
Log.e("DeleteVersion","Success delete Object Versrion");
} catch (Exception e) {
    Log.e("DeleteVersion", "e:" + e.getMessage());
}
```

List Object

功能说明

列举桶内对象（如果桶开启多版本，无法列举对象的所有版本）

返回结果

参数名称	参数描述	类型
objectSummaries	对象的列表	List
commonPrefixes	公共前缀(只有指定了delimiter,返回才有commonPrefix)	List
nextMarker	下一次列举的起始对象名	String
isTruncated	是否被截断	boolean

方法原型

```
listObjects(String bucketName)

listObjects(String bucketName, String prefix)

listObjects(ListObjectsRequest listObjectsRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
Prefix	列举前缀	String
marker	列举起始对象名称	String
delimiter	分隔符	String
maxKeys	最大返回条目数	Integer

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String Prefix = "test";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
```

```
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
ListObjectsRequest request = new ListObjectsRequest();
request.setPrefix(Prefix);
request.setBucketName(BucketName);
ObjectListing result = s3client.listObjects(request);
for(S3ObjectSummary object : result.getObjectSummaries()){
    Log.e("obj:", object.getKey());
}
} catch (Exception e){
    Log.e("ListObjects", "e:"+e.getMessage());
}
```

List Object Versions

功能说明

列举桶内对象（包含多版本对象）

返回结果

参数名称	参数描述	类型
versionSummaries	对象的列表	List
commonPrefixes	公共前缀(只有指定了delimiter,返回才有commonPrefix)	List
nextKeyMarker	下一次列举的起始Key	String
nextVersionIdMarker	下一次列举的起始VersionId	String
isTruncated	是否被截断	boolean

方法原型

```
listVersions(ListVersionsRequest listVersionsRequest)

listVersions(String bucketName, String prefix, String keyMarker, String
versionIdMarker, String delimiter, Integer maxKeys)

listVersions(String bucketName, String prefix)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
Prefix	列举前缀	String
keyMarker	列举起始对象名称	String
versionIdMarker	列举起始对象版本号	String
delimiter	分隔符	String
maxResults	最大返回条目数	Integer

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    ListVersionsRequest request = new ListVersionsRequest();
    request.setBucketName(BucketName);
    VersionListing result = s3client.listVersions(request);
    for(S3VersionSummary version_object : result.getVersionSummaries()){
        Log.e("obj:", version_object.getKey());
        Log.e("version:", version_object.getVersionId());
    }
} catch (Exception e){
    Log.e("ListObjectVersions", "e:"+e.getMessage());
}
```

Get Object Metadata

功能说明

获取对象元数据

返回结果

参数名称	参数描述	类型
result	对象元数据	ObjectMetadata

方法原型

```
getObjectMetadata(String bucketName, String key)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象对象版本号	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    ObjectMetadata result = s3client.getObjectMetadata(BucketName, ObjectName);
    Map<String, String> metaMap = result.getUserMetadata();
    metaMap.forEach((key, value) -> {
        Log.e("HeadObject", "meta:" + key + ": " + value);
    });
}catch (Exception e){
    Log.e("GetObjectMetadata", "e:" + e.getMessage());
}
```

Set Object Acl

功能说明

设置对象的ACL

返回结果

无

方法原型

```
setObjectAcl(String bucketName, String key, AccessControlList acl)

setObjectAcl(String bucketName, String key, CannedAccessControlList acl)

setObjectAcl(String bucketName, String key, String versionId, AccessControlList
acl)

setObjectAcl(String bucketName, String key, String versionId,
CannedAccessControlList acl)

setObjectAcl(SetObjectAclRequest setObjectAclRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象对象版本号	String
ACL	对象访问控制权限	支持： Private/PublicRead/PublicReadWrite/AuthenticatedRead

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    CannedAccessControlList ACL = PublicRead;
    SetObjectACLRequest request = new
    SetObjectACLRequest(BucketName,ObjectName,ACL);
    s3client.setObjectACL(request);
    Log.e("SetObjectACL","Success set the ACL");
} catch (Exception e){
    Log.e("SetObjectACL","e:"+e.getMessage());
}
```

Get Object Acl

功能说明

获取对象的ACL

返回结果

参数名称	参数描述	类型
ACL	对象访问控制权限	AccessControlList

方法原型

```
getObjectAcl(GetObjectACLRequest getObjectACLRequest)

getObjectAcl(String bucketName, String key, String versionId)

getObjectAcl(String bucketName, String key)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	版本号	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    GetObjectAclRequest request = new
    GetObjectAclRequest(BucketName,ObjectName);
    AccessControlList ACL = s3client.getObjectAcl(request);
    Log.e("SetObjectACL","the ACL:" + ACL.toString());
} catch (Exception e){
    Log.e("SetObjectACL","e:"+e.getMessage());
}
```

Set Object Tagging

功能说明

设置对象Tagging

返回结果

无

方法原型

setObjectTagging(SetObjectTaggingRequest setObjectTaggingRequest)

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
tagging	对象标签	ObjectTagging
tags	标签列表	List
versionId	对象对象版本号	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    List<Tag> tags = new ArrayList<>();
    Tag tag1 = new Tag("key1","value1");
    Tag tag2 = new Tag("key2","value2");
    tags.add(tag1);
    tags.add(tag2);
    ObjectTagging tagging = new ObjectTagging(tags);
    SetObjectTaggingRequest request = new
SetObjectTaggingRequest(BucketName,ObjectName,tagging);
    s3client.setObjectTagging(request);
    Log.e("SetObjectTagging","Success Set the Object Tagging");
} catch (Exception e){
    Log.e("SetObjectTagging","e:"+e.getMessage());
}
```

Get Object Tagging

功能说明

获取对象的标签

返回结果

参数名称	参数描述	类型
result	获取对象的标签的结果	GetObjectTaggingResult
tagSet	对象标签列表	List
key	对象标签的Key值	String
value	对象标签的Value值	String

方法原型

```
getObjectTagging(GetObjectTaggingRequest getObjectTaggingRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象对象版本号	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    GetObjectTaggingRequest request = new
    GetObjectTaggingRequest(BucketName,ObjectName);
    GetObjectTaggingResult result = s3client.getObjectTagging(request);
    for(Tag tag : result.getTagSet()){
        Log.e("GetObjectTagging","Key:"+ tag.getKey() + ",Value" +
    tag.getValue());
    }
} catch (Exception e){
    Log.e("GetObjectTagging","e:"+e.getMessage());
}
```

Delete Object Tagging

功能说明

删除对象的标签

返回结果

无

方法原型

```
deleteObjectTagging(DeleteObjectTaggingRequest deleteObjectTaggingRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象对象版本号	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    DeleteObjectTaggingRequest request = new
DeleteObjectTaggingRequest(BucketName,ObjectName);
    s3client.deleteObjectTagging(request);
    Log.e("DeleteObjectTagging","Success Delete the Object Tagging");
} catch (Exception e){
    Log.e("DeleteObjectTagging","e:"+e.getMessage());
}
```

Copy Object

功能说明

拷贝对象

返回结果

无

方法原型

```
copyObject(String sourceBucketName, String sourceKey, String
destinationBucketName, String destinationKey)

copyObject(CopyObjectRequest copyObjectRequest)
```

参数说明

参数名称	参数描述	类型
SourceBucketName	源桶名	String
SourceObjectName	源对象名	String
SourceVersionId	源对象版本号	String
DestBucketName	目标桶名	String
DestObjectName	目标对象名	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String SourceBucketName = "android";
    String SourceObjectName = "obj";
    String DestBucketName = "android";
    String DestObjectName = "obj1";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    CopyObjectRequest request = new
CopyObjectRequest(SourceBucketName,SourceObjectName, DestBucketName, DestObjectNam
e);
    s3client.copyObject(request);
    Log.e("CopyObject", "Success Copy the Object");
} catch (Exception e){
    Log.e("CopyObject", "e:"+e.getMessage());
}
```

Restore Object

功能说明

解冻归档对象

返回结果

无

方法原型

restoreObject(String bucketName, String key, int expirationInDays)

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
versionId	对象版本号	String
day	解冻天数	int

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "restore_obj";
    int day = 3;
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    s3client.restoreObject(BucketName,ObjectName,day);
    Log.e("RestoreObject","Success Restore Object ");
} catch (Exception e){
    Log.e("RestoreObject","e:"+e.getMessage());
}
```

Generate Presigned URL

功能说明

生成预期签名链接

返回结果

参数名称	参数描述	类型
url	生成的链接	URL

方法原型

```
generatePresignedUrl(String bucketName, String key, Date expiration, HttpMethod
method)

generatePresignedUrl(GeneratePresignedUrlRequest generatePresignedUrlRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
Date	链接过期时间	Date
HttpMethod	生成预签名的http方法	GET/PUT

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "obj";
    Date CurrentDay = new Date();
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(CurrentDay);
    calendar.add(Calendar.DAY_OF_MONTH,1);
    Date Date = calendar.getTime();
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    GeneratePresignedUrlRequest request = new
GeneratePresignedUrlRequest(BucketName, ObjectName, HttpMethod.GET);
    request.setExpiration(Date);
    URL url = s3client.generatePresignedUrl(request);
    Log.e("GeneratePresignedURL","Success Generate Presigned URL, the URL:+" +
url.toExternalForm());
} catch (Exception e){
    Log.e("GeneratePresignedURL","e:"+e.getMessage());
}
```

分段操作

Init Multipart Upload

功能说明

初始化分段上传

返回结果

参数名称	参数描述	类型
result	初始化分段上传结果	InitiateMultipartUploadResult
uploadId	初始化分段上传的ID	String

方法原型

```
initiateMultipartUpload(InitiateMultipartUploadRequest  
initiateMultipartUploadRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
ACL	设置桶CannedAccessControlList，可选项： Private，私有（默认）； PublicRead，公共读，但只有桶所有者可以写入和更改ACL； PublicReadWrite，公共读写，但只有桶所有者可以写入和更改ACL； AuthenticatedRead，对象所有者和存储桶所有者可以读取该对象，只有对象所有者可以写入和更改ACL。	CannedAccessControlList
StorageClass	对象存储类别	StorageClass(支持：Standard: 标准/StandardInfrequentAccess: 低频/Glacier:归档)
tagging	对象存储tag	ObjectTagging
objectMetadata	对象存储元数据	ObjectMetadata

示例

```
try{  
    String AccessKey = "accessKey";  
    String SecretKey = "secretKey";  
    String Endpoint = "endPoint";  
    String BucketName = "android";  
    String ObjectName = "multi_part";  
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,  
SecretKey);  
    AmazonS3Client s3client = new AmazonS3Client(credentials);  
    s3client.setEndpoint(Endpoint);  
    InitiateMultipartUploadRequest request = new  
InitiateMultipartUploadRequest(BucketName, ObjectName);  
    InitiateMultipartUploadResult result =  
s3client.initiateMultipartUpload(request);  
    Log.e("InitMultipartPartUpload","success init multipart,the upload id:"+  
result.getUploadId());  
} catch (Exception e){  
    Log.e("InitMultipartPartUpload","e:"+e.getMessage());  
}
```

Upload Part

功能说明

上传分段

返回结果

参数名称	参数描述	类型
result	分段上传结果	UploadPartResult
PartNumber	分段号	int
eTag	段的eTag	String

方法原型

uploadPart(UploadPartRequest uploadPartRequest)

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
UploadId	初始化分段上传的ID	String
PartNum	分段上传段号(从1开始)	int
size	分段上传段大小	long
input	分段上传段内容	InputStream

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "multi_part";
    String UploadId = "";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    for(int PartNum = 1; PartNum < 4; ++PartNum){
        int size = 5 * 1024 * 1024;
        byte[] data = new byte[size];
        for (int i = 0; i < data.length; i++) {
            data[i] = (byte) (i % 256);
        }
    }
}
```

```
InputStream input = new ByteArrayInputStream(data);
UploadPartRequest request = new UploadPartRequest();
request.setUploadId(UploadId);
request.setPartNumber(PartNum);
request.setInputStream(input);
request.setBucketName(BucketName);
request.setKey(ObjectName);
request.setPartSize(size);
UploadPartResult result = s3client.uploadPart(request);
Log.e("UploadMultiPart","the part num: "+
result.getPartETag().getPartNumber()+"the Etag:" +
result.getPartETag().getETag());
}
Log.e("UploadMultiPart","Success Upload Multi Part");
} catch (Exception e){
Log.e("UploadMultiPart","e:"+e.getMessage());
}
```

Compelete Multipart Part

功能说明

分段上传合并

返回结果

无

方法原型

```
completeMultipartUpload(CompleteMultipartUploadRequest
completeMultipartUploadRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
UploadId	初始化分段上传的ID	String
partETags	分段上传段的ETag列表	List

示例

```
try{
String AccessKey = "accessKey";
String SecretKey = "secretKey";
String Endpoint = "endPoint";
String BucketName = "android";
String ObjectName = "multi_part";
String UploadId = "";
```

```
List<PartETag> partETags = new ArrayList<>();
PartETag partETag1 = new PartETag(1,"");
PartETag partETag2 = new PartETag(2,"");
PartETag partETag3 = new PartETag(3,"");
partETags.add(partETag1);
partETags.add(partETag2);
partETags.add(partETag3);
BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
SecretKey);
AmazonS3Client s3client = new AmazonS3Client(credentials);
s3client.setEndpoint(Endpoint);
CompleteMultipartUploadRequest request = new
CompleteMultipartUploadRequest();
request.setBucketName(BucketName);
request.setKey(ObjectName);
request.setUploadId(UploadId);
request.setPartETags(partETags);
s3client.completeMultipartUpload(request);
Log.e("CompeleteMultipartPartUpload","Success Compelete Multipart");
} catch (Exception e){
Log.e("CompeleteMultipartPartUpload","e:"+e.getMessage());
}
```

List Parts

功能说明

列举分段上传的段

返回结果

参数名称	参数描述	类型
bucketName	桶名	String
keyMarker	列举起始的key	String
uploadIdMarker	与keyMarker一起指定list的起始位置	String
prefix	列举前缀	String
maxUploads	分段碎片最大返回数	Integer
isTruncated	是否被截断	boolean
nextKeyMarker	下一次列举起始的Key	String
nextUploadIdMarker	下一次列举的起始UploadId	String
multipartUploads	列举结果	List

方法原型

```
listMultipartUploads(ListMultipartUploadsRequest listMultipartUploadsRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
prefix	列举前缀	String
maxUploads	分段碎片最大返回数	Integer
keyMarker	指定列举开头的key	String
uploadIdMarker	与keyMarker一起指定list的起始位置	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
    SecretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    ListMultipartUploadsRequest request = new
    ListMultipartUploadsRequest(BucketName);
    MultipartUploadListing result = s3client.listMultipartUploads(request);
    for(MultipartUpload upload : result.getMultipartUploads()){
        Log.e("ListMultipartPartUpload","the key:"+upload.getKey() + ",the
        uploadId:"+upload.getUploadId());
    }
} catch (Exception e){
    Log.e("ListMultipartPartUpload","e:"+e.getMessage());
}
```

Abort Multipart

功能说明

取消对象分段上传，删除碎片

返回结果

无

方法原型

```
abortMultipartUpload(AbortMultipartUploadRequest abortMultipartUploadRequest)
```

参数说明

参数名称	参数描述	类型
BucketName	桶名	String
ObjectName	对象名	String
UploadId	初始化分段上传的ID	String

示例

```
try{
    String AccessKey = "accessKey";
    String SecretKey = "secretKey";
    String Endpoint = "endPoint";
    String BucketName = "android";
    String ObjectName = "multi_part";
    String UploadId = "";
    BasicAWSCredentials credentials = new BasicAWSCredentials(AccessKey,
secretKey);
    AmazonS3Client s3client = new AmazonS3Client(credentials);
    s3client.setEndpoint(Endpoint);
    AbortMultipartUploadRequest request = new
AbortMultipartUploadRequest(BucketName,ObjectName,UploadId);
    s3client.abortMultipartUpload(request);
    Log.e("AbortMultipartPartUpload","Success Abort Multipart Upload");
} catch (Exception e){
    Log.e("AbortMultipartPartUpload","e:"+e.getMessage());
}
```