

ZOS对象存储IOS_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) 本地安装（版本需求：IOS 9以上）
1、将SDK压缩包zos-ios-sdk.zip解压到根目录
2、配置Profile文件
target 'ZOSDemo' do
  pod 'AWSS3', :path => './zos-ios-sdk'
  pod 'AWSCore', :path => 'zos-ios-sdk'
end
3、执行pod install完成安装
```

初始化连接

在使用SDK接口之前，需要先连接到ZOS，使用以下示例可以连接到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。

```
//引用ios sdk的头文件
#import <AWSS3/AWSS3.h>
//通过sdk使用s3服务的之前需要先完成accesskey, secretkey, endpoint的配置，完成sdk初始化
AWSStaticCredentialsProvider *credential =
[[AWSStaticCredentialsProvider alloc] initWithAccessKey:@"accesskey"
secretKey:@"secretkey"];
    AWSEndpoint *endpoint = [[AWSEndpoint alloc]
initWithURLString:@"endpoint"];
    AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
    [AWSServiceManager
defaultServiceManager].defaultServiceConfiguration = configure;
    AWSS3 *s3 = [AWSS3 defaults3];
```

相关参数说明：

参数名称	参数描述	类型
accesskey	ZOS账号的access key，可从控制台查看	NSString*
secretkey	ZOS账号的secret key，可从控制台查看	NSString*
endpoint	连接到ZOS资源池的地址，需要指定http或https前缀	NSString*

桶操作

Create Bucket

功能说明

创建一个桶

返回结果

无

方法原型

```
- (AWSTask<AWSS3CreateBucketOutput *> *)createBucket:(AWSS3CreateBucketRequest *)request;
```

参数说明

参数名称	参数描述	类型
request	创建桶请求	AWSS3CreateBucketRequest*
bucket	桶名(长度范围为3到63个字符, 支持小写字母、数字、中划线(-)。禁止以中划线(-)开头或结尾。)	NSString*
objectLockEnabledForBucket	是否开启对象合规保留(可选)	NSNumber* (0关闭, 1开启)
ACL	桶访问权限(可选)	AWSS3ObjectCannedACL (支持 AWSS3ObjectCannedACLPrivate、AWSS3ObjectCannedACLPublicRead、AWSS3ObjectCannedACLAuthenticatedRead)
azPolicy	桶AZ策略(可选)	NSString*(SINGLE-AZ、MULTI-AZ)
storageClass	桶存储级别(可选)	AWSS3StorageClass(支持 AWSS3StorageClassStandard(标准存储)、AWSS3StorageClassStandardIa(低频存储)、AWSS3StorageClassGlacier(归档存储))

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint credentialsProvider:credential];
```

```

[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3CreateBucketRequest *request = [[AWSS3CreateBucketRequest alloc] init];
request.bucket = @"ios_test";
[s3 createBucket:request completionHandler:^(AWSS3CreateBucketOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"success create:%@",request.bucket);
    }
}]];

```

Delete Bucket

功能说明

删除一个空桶

返回结果

无

方法原型

```

- (AWSTask *)deleteBucket:(AWSS3DeleteBucketRequest *)request;

```

参数说明

参数名称	参数描述	类型
request	删除桶请求	AWSS3DeleteBucketRequest*
bucket	桶名	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3DeleteBucketRequest *request = [[AWSS3DeleteBucketRequest alloc] init];
request.bucket = @"ios_test";
[s3 deleteBucket:request completionHandler:^(NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else {
        NSLog(@"success delete the bucket :%@",request.bucket);
    }
}]];

```

```
}  
}];
```

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

```
- (void)headBucket:(AWSS3HeadBucketRequest *)request completionHandler:(void (^  
_Nullable)(NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	HeadBucket请求	AWSS3HeadBucketRequest*
bucket	桶名称	NSString*

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider alloc]  
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];  
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];  
AWSServiceConfiguration *configure = [[AWSServiceConfiguration  
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint  
credentialsProvider:credential];  
[AWSServiceManager defaultManager].defaultServiceConfiguration =  
configure;  
AWSS3 *s3 = [AWSS3 defaultS3];  
AWSS3HeadBucketRequest *request = [[AWSS3HeadBucketRequest alloc] init];  
request.bucket = @"ios_test";  
[s3 headBucket:request completionHandler:^(NSError * _Nullable error) {  
    if(error != nil){  
        NSLog(@"Error, %@",error);  
    } else{  
        NSLog(@"Head bucket success, the bucket %@",request.bucket);  
    }  
}];
```

List Bucket

功能说明

列举用户下的所有桶

返回结果

参数名称	参数描述	类型
response	list buckets 返回结果	AWSS3ListBucketsOutput*
buckets	用户下所有桶信息	NSArray<AWSS3Bucket >

方法原型

```
- (void)listBuckets:(AWSRequest *)request completionHandler:(void (^ _Nullable)
(AWSS3ListBucketsOutput * _Nullable response, NSError * _Nullable
error))completionHandler;
```

参数说明

无

示例

```
AWSSStaticCredentialsProvider *credential = [[AWSSStaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSRequest *request = [[AWSRequest alloc] init];
[s3 listBuckets:request completionHandler:^(AWSS3ListBucketsOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    }else{
        for(AWSS3Bucket* bucket in response.buckets){
            NSLog(@"Bucket:%@",bucket.name);
        }
    }
}];
```

Put Bucket Tagging

功能说明

设置桶标签

返回结果

无

方法原型

```
- (void)putBucketTagging:(AWSS3PutBucketTaggingRequest *)request
completionHandler:(void (^ _Nullable)(NSError * _Nullable
error))completionHandler;
```

参数说明

参数名称	参数描述	类型
tagging	桶标签，一个桶可以设置多个标签	AWSS3Tagging *
tag	标签信息 (key,value) 形式	AWSS3Tag*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSS3ServiceConfiguration *configure = [[AWSS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3PutBucketTaggingRequest *request = [[AWSS3PutBucketTaggingRequest
alloc] init];
AWSS3Tag* tag = [[AWSS3Tag alloc] init];
tag.key = @"key1";
tag.value = @"value1";
NSArray<AWSS3Tag*> * tagset = [[NSArray<AWSS3Tag*> alloc] initWithObjects:tag,
nil];

AWSS3Tagging *tagging = [[AWSS3Tagging alloc] init];
tagging.tagSet = tagset;
request.bucket = @"ios_test";
request.tagging = tagging;
[s3 putBucketTagging:request completionHandler:^(NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error: %@",error);
    } else{
        NSLog(@"success set taggging");
    }
}];
```

Get Bucket Tagging

功能说明

获取桶标签

返回结果

参数名称	参数描述	类型
response	查询桶标签的结果	AWSS3GetBucketTaggingOutput*
tagSet	桶标签列表	NSArray<AWSS3Tag >
tag	标签信息 (key,value) 形式	AWSS3Tag*

方法原型

```
- (void)getBucketTagging:(AWSS3GetBucketTaggingRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3GetBucketTaggingOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSS3ServiceConfiguration *configure = [[AWSS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3GetBucketTaggingRequest *request = [[AWSS3GetBucketTaggingRequest
alloc] initWithBucket:@"ios_test"];
request.bucket = @"ios_test";
[s3 getBucketTagging:request completionHandler:^(AWSS3GetBucketTaggingOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSArray<AWSS3Tag *> *tagset = response.tagSet;
        for(AWSS3Tag *tag in tagset){
            NSLog(@"key:%@ value:%@",tag.key,tag.value);
        }
    }
}];
```

Delete Bucket Tagging

功能说明

删除桶标签

返回结果

无

方法原型

```
- (void)deleteBucketTagging:(AWSS3DeleteBucketTaggingRequest *)request  
completionHandler:(void (^ _Nullable)(NSError * _Nullable  
error))completionHandler;
```

参数说明

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc]  
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];  
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];  
AWSServiceConfiguration *configure = [[AWSServiceConfiguration  
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint  
credentialsProvider:credential];  
[AWSServiceManager defaultManager].defaultServiceConfiguration =  
configure;  
AWSS3 *s3 = [AWSS3 defaultS3];  
AWSS3DeleteBucketTaggingRequest *request = [[AWSS3DeleteBucketTaggingRequest  
alloc] init];  
request.bucket = @"ios_test";  
[s3 deleteBucketTagging:request completionHandler:^(NSError * _Nullable error) {  
    if(error != nil){  
        NSLog(@"Error:%@",error);  
    }else{  
        NSLog(@"success delete the bucket taggging");  
    }  
}];
```

Put Bucket Logging

功能说明

设置桶日志参数

返回结果

无

方法原型

```
- (void)putBucketLogging:(AWSS3PutBucketLoggingRequest *)request
completionHandler:(void (^ _Nullable)(NSError * _Nullable
error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	设置桶日志的请求	AWSS3PutBucketLoggingRequest*
bucketLoggingStatus	设置桶日志配置	AWSS3BucketLoggingStatus*
loggingEnabled	日志的配置	AWSS3LoggingEnabled *
targetBucket	生成桶日志目标桶桶名	NSString*
targetPrefix	生成桶日志目标前缀	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3PutBucketLoggingRequest *request = [[AWSS3PutBucketLoggingRequest
alloc] init];
AWSS3BucketLoggingStatus *loggingstatus = [[AWSS3BucketLoggingStatus alloc]
init];
AWSS3LoggingEnabled *loggingenable = [[AWSS3LoggingEnabled alloc] init];
loggingenable.targetBucket = @"ios_test";
loggingenable.targetPrefix = @"test";
loggingstatus.loggingEnabled = loggingenable;
request.bucket = @"ios_test";
request.bucketLoggingStatus = loggingstatus;
[s3 putBucketLogging:request completionHandler:^(NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success set bucket logging");
    }
}];
```

Get Bucket Logging

功能说明

获取桶日志配置

返回结果

参数名称	参数描述	类型
response	获取桶日志的结果	AWSS3GetBucketLoggingRequest*
loggingEnabled	日志的配置	AWSS3LoggingEnabled *
targetBucket	生成桶日志目标桶桶名	NSString*
targetPrefix	生成桶日志目标前缀	NSString*

方法原型

```
- (void)getBucketLogging:(AWSS3GetBucketLoggingRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3GetBucketLoggingOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	获取桶日志的请求	AWSS3GetBucketLoggingRequest*

示例

```
AWSSStaticCredentialsProvider *credential = [[AWSSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetBucketLoggingRequest *request = [[AWSS3GetBucketLoggingRequest alloc]
init];
request.bucket = @"ios_test";
[s3 getBucketLogging:request completionHandler:^(AWSS3GetBucketLoggingOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    }else{
        NSLog(@"Target bucket:%@",response.loggingEnabled.targetBucket);
        NSLog(@"Target prefix:%@",response.loggingEnabled.targetPrefix);
    }
}];
```

Set Bucket Version

功能说明

桶开启/暂停多版本

返回结果

无

方法原型

```
- (void)putBucketVersioning:(AWSS3PutBucketVersioningRequest *)request
completionHandler:(void (^ _Nullable)(NSError * _Nullable
error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	开启/暂停桶多版本请求	AWSS3PutBucketVersioningRequest*
configuration	桶多版本配置	AWSS3VersioningConfiguration*
status	桶多版本状态	AWSS3BucketVersioningStatusEnabled/AWSS3BucketVersioningStatusSuspended

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaults3];
AWSS3PutBucketVersioningRequest *request = [[AWSS3PutBucketVersioningRequest
alloc] init];
AWSS3VersioningConfiguration *configuration = [[AWSS3VersioningConfiguration
alloc] init];
configuration.status = AWSS3BucketVersioningStatusEnabled;
request.bucket = @"ios_test";
request.versioningConfiguration = configuration;
[s3 putBucketVersioning:request completionHandler:^(NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error : %@",error);
    }else{
        NSLog(@"Success set the bucket version");
    }
}];
```

Get Bucket Version

功能说明

获取桶多版本配置

返回结果

参数名称	参数描述	类型
response	获取桶多版本信息结果	AWSS3GetBucketVersioningOutput*
response	获取桶多版本信息结果	AWSS3GetBucketVersioningOutput*

方法原型

```
- (AWSTask<AWSS3GetBucketVersioningOutput *> *)getBucketVersioning:
(AWSS3GetBucketVersioningRequest *)request;
```

参数说明

参数名称	参数描述	类型
response	获取桶多版本信息结果	AWSS3GetBucketVersioningOutput*
status	获取桶版本状态	NS_ENUM

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetBucketVersioningRequest *request = [[AWSS3GetBucketVersioningRequest
alloc] init];
request.bucket = @"ios_test";
[s3 getBucketVersioning:request
completionHandler:^(AWSS3GetBucketVersioningOutput * _Nullable response, NSError
* _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"the bucket status %ld",response.status);
    }
}];
```

Set Bucket Acl

功能说明

设置桶访问权限,有三种方式设置桶的访问权限,三种方式不可同时使用,每次只能给一种参数赋值。其中,通过ACL参数方式进行操作,是设置预定义的固定的ACL,不能针对特定用户进行授权;
AWSS3AccessControlPolicy参数方式和Grant*参数方式则可以针对特定用户进行授权,
AccessControlPolicy方式通过请求体传递,而 Grant*方式通过请求头传递。三种方式都会覆盖原有ACL属性,包括桶所有者自身的权限,如需保留原有 ACL 属性,应将需要保留的原ACL添加到本次操作的授权中 (ACL 参数方式会默认将桶所有者权限设为FULL_CONTROL,而另外两种方式则不会保留任何原 ACL 属性)。

返回结果

无

方法原型

```
- (void)putBucketAcl:(AWSS3PutBucketAclRequest *)request completionHandler:(void (^ _Nullable)(NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	设置桶访问权限请求	AWSS3PutBucketAclRequest*
ACL	桶访问权限请求	AWSS3BucketCannedACL
accessControlPolicy	桶访问权限请求	AWSS3AccessControlPolicy*
grantFullControl	允许grantee read, write, read ACP, 和 write ACP	NSString*
grantRead	授予读桶内对象权限	NSString*
grantReadACP	授予读桶权限	NSString*
grantWrite	授予写桶内对象权限	NSString*
grantWriteACP	授予写桶权限	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration = configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3PutBucketAclRequest *request = [[AWSS3PutBucketAclRequest alloc] init];
request.bucket = @"ios_test";
```

```
request.ACL = AWSS3BucketCannedACLPrivate;
[s3 putBucketAcl:request completionHandler:^(NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success set acl");
    }
}];
```

Get Bucket ACL

功能说明

获取桶访问权限

返回结果

参数名称	参数描述	类型
response	设置桶访问权限请求	AWSS3GetBucketAclOutput*
grants	桶访问权限列表	NSArray<AWSS3Grant *>*

方法原型

```
- (void)getBucketAcl:(AWSS3GetBucketAclRequest *)request completionHandler:(void (^ _Nullable)(AWSS3GetBucketAclOutput * _Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	获取桶访问权限请求	AWSS3GetBucketAclRequest*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetBucketAclRequest *request = [[AWSS3GetBucketAclRequest alloc] init];
request.bucket = @"ios_test";
[s3 getBucketAcl:request completionHandler:^(AWSS3GetBucketAclOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success get the acl");
    }
}];
```

```
    }  
    }];
```

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
response	列举桶内对象返回结果	AWSS3ListObjectsOutput*
commonPrefixes	公共前缀，在计算返回次数时，所有聚集在一个公共前缀中的键都算作一个返回，只有指定了delimiter，响应才包含CommonPrefixes;例如，如果prefix是notes/，delimiter是斜线(/)，如notes/summer/july，则CommonPrefixes是notes/summer/。在计算返回次数时，所有汇总到一个公共前缀中的键都算作一个返回值	NSArray<AWSS3CommonPrefix*>*
contents	列举的每一个对象的信息	NSArray<AWSS3Object*>*
encodingType	响应中对象键的编码方式	AWSS3EncodingType
isTruncated	是否截断	NSNumber*
marker	列举开始的key	NSString*
maxKeys	列举最大返回条目数	NSNumber*
name	桶名称	NSString*
nextMarker	当返回被截断,下一次开始的key	NSString*
prefix	列举前缀	NSString*

方法原型

```
- (void)listObjects:(AWSS3ListObjectsRequest *)request completionHandler:(void (^ _Nullable)(AWSS3ListObjectsOutput * _Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	列举桶内对象请求	AWSS3ListObjectsRequest*
bucket	是	桶名称	NSString*
delimiter	否	分隔符	NSString*
marker	否	列举起始对象名称	NSString*
maxKeys	否	列举返回最大条目数	NSNumber*
prefix	否	列举前缀	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaults3];
AWSS3ListObjectsRequest *request = [[AWSS3ListObjectsRequest alloc] init];
request.bucket = @"ios_test";
[s3 listObjects:request completionHandler:^(AWSS3ListObjectsOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    }else{
        for(AWSS3Object *object in response.contents){
            NSLog(@"the object:%@",object.key);
        }
    }
}];

```

List Object Version

功能说明

列举桶内对象，包含对象的版本信息

返回结果

参数名称	参数描述	类型
response	列举桶内对象返回结果	AWSS3ListObjectsOutput*
commonPrefixes	公共前缀	NSArray<AWSS3CommonPrefix*>*
versions	列举的每一个对象的信息	NSArray<AWSS3ObjectVersion*>*
encodingType	响应中对象键的编码方式	AWSS3EncodingType
isTruncated	是否截断	NSNumber*
keyMarker	列举开始的key	NSString*
versionIdMarker	列举开始版本信息	NSString*
maxKeys	列举最大返回条目数	NSNumber*
name	桶名称	NSString*
nextMarker	当返回被截断,下一次开始的key	NSString*
nextVersionIdMarker	当返回被截断,下一次开始的版本信息	NSString*
prefix	列举前缀	NSString*
deleteMarkers	带有删除标记的对象	NSArray<AWSS3DeleteMarkerEntry*>*

方法原型

```
- (void)listObjectVersions:(AWSS3ListObjectVersionsRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3ListObjectVersionsOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	列举桶内对象请求	AWSS3ListObjectVersionsRequest*
bucket	是	桶名称	NSString*
delimiter	否	分隔符	NSString*
keyMarker	否	列举起始对象名称	NSString*
maxKeys	否	列举返回最大条目数	NSNumber*
prefix	否	列举前缀	NSString*
versionIdMarker	否	列举起始的版本信息	NSString*

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider alloc]
initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaults3];
AWSS3ListObjectVersionsRequest *request = [[AWSS3ListObjectVersionsRequest alloc]
init];
request.bucket = @"ios_test";
[s3 listObjectVersions:request completionHandler:^(AWSS3ListObjectVersionsOutput
* _Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        for(AWSS3ObjectVersion *obj_version in response.versions){
            NSLog(@"key:%@,
version_id:%@",obj_version.key,obj_version.versionId);
        }
    }
}];
```

对象操作

Put Object

功能说明

将一个文件（Object）上传至指定 Bucket。

返回结果

方法原型

```
(void)putObject:(AWSS3PutObjectRequest *)request completionHandler:(void (^
_Nullable)(AWSS3PutObjectOutput * _Nullable response, NSError * _Nullable
error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	上传请求	AWSS3PutObjectRequest*
response	上传的返回结果	AWSS3PutObjectOutput*
error	上传异常返回	NSError*
bucket	桶名	NSString*
key	对象名	NSString*
body	上传对象的内容	Id(通用类型)
ACL	上传对象的访问权限 (可选)	AWSS3ObjectCannedACL (支持 AWSS3ObjectCannedACLPrivate、AWSS3ObjectCannedACLPublicRead)
storageClass	上传对象的存储级别 (可选)	AWSS3StorageClass,支持AWSS3StorageClassStandard (标准存储)、AWSS3StorageClassStandardIa (低频存储)、AWSS3StorageClassGlacier (归档存储)
metadata	上传对象的元数据	NSDictionary<NSString *, NSString *> *

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3PutObjectRequest *request = [[AWSS3PutObjectRequest alloc] init];
NSString *content = @"test content";
NSData *data = [content dataUsingEncoding:NSUTF8StringEncoding];
request.bucket = @"ios_test";
request.key = @"test_file";
request.body = data;
request.contentLength = [NSNumber numberWithInt:data.length];
[s3 putObject:request completionHandler:^(AWSS3PutObjectOutput * _Nullable
reponse, NSError * _Nullable error){
    if(error != nil){
        NSLog(@"error: %@",error);
    } else{
        NSLog(@"success put the obj:%@",request.key);
    }
}];

```

Get Object

功能说明

将一个ZOS桶里的对象下载至本地。该操作需要对目标Object具有读权限

返回结果

参数名称	参数描述	类型
response	下载返回	AWSS3GetObjectOutput*
body	返回数据内容	Id(通用类型)
contentLength	body大小(bytes)	NSNumber *
contentType	类型	NSString *
ETag	下载对象的ETag	NSString*
lastModified	对象最后修改时间	NSDate *
metadata	对象元数据	NSDictionary<NSString *, NSString >
ETag	下载对象的ETag	NSString*

方法原型

```
- (void)getObject:(AWSS3GetObjectRequest *)request completionHandler:(void (^ _Nullable)(AWSS3GetObjectOutput * _Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	下载请求	AWSS3GetObjectRequest*
bucket	桶名	NSString*
key	对象名	NSString*
versionId	对象版本号	NSString*

示例

```
AWSSStaticCredentialsProvider *credential = [[AWSSStaticCredentialsProvider alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration = configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetObjectRequest *request = [[AWSS3GetObjectRequest alloc] init];
```

```

request.bucket = @"ios_test";
request.key = @"test_file";
[s3 getObject:request completionHandler:^(AWSS3GetObjectOutput * _Nullable
response, NSError * _Nullable error){
    if(error != nil){
        NSLog(@"Error: %@",error);
    }else{
        NSLog(@"Success get the obj:%@",request.key);
    }
    // 保存对象至本地
    NSString *filepath = [NSHomeDirectory()
stringByAppendingPathComponent:@"test_file"];
    NSData *body = response.body;
    NSError *err = nil;
    NSString *result = [[NSString alloc] initWithData:body
encoding:NSUTF8StringEncoding];
    BOOL success = [result writeToFile:filepath atomically:YES
encoding:NSUTF8StringEncoding error:&err];
    if(success){
        NSLog(@" write success");
    }else{
        NSLog(@"write fail");
    }
}
]];

```

Delete Object

功能说明

删除桶内指定对象

返回结果

无

方法原型

```

- (void)deleteObject:(AWSS3DeleteObjectRequest *)request completionHandler:(void
(^ _Nullable)(AWSS3DeleteObjectOutput * _Nullable response, NSError * _Nullable
error))completionHandler;

```

参数说明

参数名称	参数描述	类型
request	删除请求	AWSS3GetObjectRequest*
bucket	桶名	NSString*
key	对象名	NSString*
versionId	对象版本号	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc]initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc]initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaults3];
AWSS3DeleteObjectRequest *request = [[AWSS3DeleteObjectRequest alloc]init];
request.bucket = @"ios_test";
request.key = @"test_file";
[s3 deleteObject:request completionHandler:^(AWSS3DeleteObjectOutput * _Nullable
reponse, NSError * _Nullable error){
    if(error != nil){
        NSLog(@"error: %@",error);
    } else{
        NSLog(@"success delete the obj");
    }
}]];

```

Delete Objects

功能说明

删除桶内一批对象

返回结果

无

方法原型

```

- (void)deleteObjects:(AWSS3DeleteObjectsRequest *)request completionHandler:
(void (^)( _Nullable)AWSS3DeleteObjectsOutput * _Nullable response, NSError *
 _Nullable error))completionHandler;

```

参数说明

参数名称	参数描述	类型
request	删除请求	AWSS3DeleteObjectsRequest*
bucket	桶名	NSString*
obj	删除的对象信息	AWSS3ObjectIdentifier*
key	删除对象的对象名	NSString*
versionId	删除对象的版本号	NSString*
objs	删除对象列表	NSArray<AWSS3ObjectIdentifier *> *

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3DeleteObjectsRequest *request = [[AWSS3DeleteObjectsRequest alloc] init];
request.bucket = @"ios_test";
AWSS3ObjectIdentifier *obj = [[AWSS3ObjectIdentifier alloc] init];
AWSS3ObjectIdentifier *obj1 = [[AWSS3ObjectIdentifier alloc] init];
AWSS3ObjectIdentifier *obj2 = [[AWSS3ObjectIdentifier alloc] init];
obj.key = @"test";
obj1.key = @"delete";
obj2.key = @"delete1";
AWSS3Remove *remove = [[AWSS3Remove alloc] init];
NSArray<AWSS3ObjectIdentifier *> *objs = [[NSArray<AWSS3ObjectIdentifier *>
alloc] initWithObjects:obj,obj1,obj2, nil];
remove.objects = objs;
request.remove = remove;
[s3 deleteObjects:request completionHandler:^(AWSS3DeleteObjectsOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error %@",error);
    } else{
        NSLog(@"Success delete the objs");
    }
}
}];
```

Head Object

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
response	获取对象元数据返回结果	AWSS3HeadObjectOutput*
contentType	编码方式	NSString *
contentLength	对象大小	NSNumber*
encodingType	响应中对象键的编码方式	AWSS3EncodingType
deleteMarker	对象是否为删除标记	NSNumber*
marker	列举开始的key	NSString*
ETag	对象ETag信息	NSString*
lastModified	最后更新时间	NSDate *
metadata	元数据信息	NSDictionary<NSString *, NSString *>*
objectLockLegalHoldStatus	对象是否处于合法保留状态	AWSS3ObjectLockLegalHoldStatus
storageClass	对象存储级别	AWSS3StorageClass
versionId	对象版本号	NSString*

方法原型

```
- (void)headObject:(AWSS3HeadObjectRequest *)request completionHandler:(void (^ _Nullable)(AWSS3HeadObjectOutput * _Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	获取对象元数据请求	AWSS3HeadObjectRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
versionId	否	对象版本号	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint credentialsProvider:credential];
```



```
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3HeadObjectRequest *request = [[AWSS3HeadObjectRequest alloc] init];
request.bucket= @"ios_test";
request.key = @"test";
[s3 headObject:request completionHandler:^(AWSS3HeadObjectOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Etag:%@",response.Etag);
        NSLog(@"MetaData");
        for(NSString *key in response.metadata){
            NSString *value = response.metadata[key];
            NSLog(@"Key:%@,value:%@",key,value);
        }
    }
}];
```

Put Object Acl

功能说明

设置Object的ACL，控制对Object的访问权限。该操作需要用户具有WRITE_ACP权限。

有三种方式设置ACL，三种方式不可同时使用，每次只能给一种参数赋值。其中，通过AWSS3ObjectCannedACL参数方式进行操作，是设置预定义的固定的ACL，不能针对特定用户进行授权，且该参数实现的效果，也可以借由另外两种方式实现，该参数使用请求头进行传递；accessControlPolicy参数方式和Grant*参数方式则可以针对特定用户进行授权。三种方式都会覆盖原有ACL属性，包括对象所有者自身的权限，如需保留原有ACL属性，应将需要保留的原ACL添加到本次操作的授权中。

返回结果

无

方法原型

```
- (AWSTask<AWSS3PutObjectAclOutput *> *)putObjectAcl:(AWSS3PutObjectAclRequest
*)request;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	设置对象ACL请求	AWSS3PutObjectAclRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
AWSS3ObjectCannedACL	ACL参数方式则必须, 其他方式, 则不能使用	对象ACL	AWSS3ObjectCannedACL(可取值: AWSS3ObjectCannedACLPrivate、AWSS3ObjectCannedACLPublicRead、AWSS3ObjectCannedACLAuthenticatedRead)
accessControlPolicy	该方式下必须, 其他方式下则不能使用	每个被授权者设置对象ACL权限的元素	AWSS3AccessControlPolicy *
grantRead	该方式下必须, 其他方式下则不能使用	允许被授权者获取对象	NSString*
grantReadACP	该方式下必须, 其他方式下则不能使用	允许被授权者获取ACL	NSString*
grantWrite	该方式下必须, 其他方式下则不能使用	允许被授权者删除、修改对象	NSString*
grantWriteACP	该方式下必须, 其他方式下则不能使用	允许被授权者修改ACL	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];

```

```

[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3PutObjectAclRequest *request = [[AWSS3PutObjectAclRequest alloc] init];
request.bucket = @"ios_test";
request.ACL = AWSS3ObjectCannedACLPrivate;
request.key = @"test";
[s3 putObjectAcl:request completionHandler:^(AWSS3PutObjectAclOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error: %@",error);
    } else{
        NSLog(@"Success, put the acl for the key: %@", request.key);
    }
}];

```

Get Object Acl

功能说明

获取指定Object的ACL。该操作需要READ_ACP权限。

返回结果

参数名称	参数描述	类型
response	获取对象ACL返回结果	AWSS3GetObjectAclOutput*
grants	权限列表	NSArray<AWSS3Grant *>*

方法原型

```

- (void)getObjectAcl:(AWSS3GetObjectAclRequest *)request completionHandler:(void
(^ _Nullable)(AWSS3GetObjectAclOutput * _Nullable response, NSError * _Nullable
error))completionHandler;

```

参数说明

参数名称	是否必须	参数描述	类型
request	是	获取对象ACL请求	AWSS3GetObjectAclRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
versionId	否	对象版本号	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];

```

```

AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc]initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetObjectAclRequest *request = [[AWSS3GetObjectAclRequest alloc]init];
request.bucket = @"ios_test";
request.key = @"test";
[s3 getObjectAcl:request completionHandler:^(AWSS3GetObjectAclOutput * _Nullable
response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        for(AWSS3Grant *grant in response.grants){
            NSNumber *permission = @(grant.permission);
            NSLog(@"the permission: %@",permission);
        }
    }
}];

```

Put Object Tagging

功能说明

设置对象标签

返回结果

参数名称	参数描述	类型
response	设置对象标签返回结果	AWSS3PutObjectTaggingOutput*
versionId	设置标签对象的versionID	NSString*

方法原型

```

- (void)putObjectTagging:(AWSS3PutObjectTaggingRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3PutObjectTaggingOutput * _Nullable
response, NSError * _Nullable error))completionHandler;

```

参数说明

参数名称	是否必须	参数描述	类型
request	是	设置对象标签请求	AWSS3PutObjectTaggingRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
tagging	是	对象标签	AWSS3Tagging *
versionId	否	对象版本号	NSString*

示例

```
AWSSStaticCredentialsProvider *credential = [[AWSSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3PutObjectTaggingRequest *request = [[AWSS3PutObjectTaggingRequest alloc]
init];
AWSS3Tag *tag = [[AWSS3Tag alloc] init];
tag.key = @"obj_key";
tag.value = @"obj_value";
NSArray<AWSS3Tag *> *tags = [[NSArray<AWSS3Tag *> alloc] initWithObjects:tag,
nil];
AWSS3Tagging *tagging = [[AWSS3Tagging alloc] init];
tagging.tagSet = tags;
request.bucket = @"ios_test";
request.key = @"test";
request.tagging = tagging;
[s3 putObjectTagging:request completionHandler:^(AWSS3PutObjectTaggingOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success put the object tagging");
    }
}
}];
```

Get Object Tagging

功能说明

获取对象标签信息

返回结果

参数名称	参数描述	类型
response	获取对象标签返回结果	AWSS3GetObjectTaggingOutput*
versionId	标签对象的versionID	NSString*
tagSet	标签对象的标签列表	NSArray<AWSS3Tag *>*

方法原型

```
- (void)getObjectTagging:(AWSS3GetObjectTaggingRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3GetObjectTaggingOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	获取对象标签请求	AWSS3GetObjectTaggingRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
versionId	否	对象版本号	NSString*

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3GetObjectTaggingRequest *request = [[AWSS3GetObjectTaggingRequest alloc]
init];
request.bucket = @"ios_test";
request.key = @"test";
[s3 getObjectTagging:request completionHandler:^(AWSS3GetObjectTaggingOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        for(AWSS3Tag *tag in response.tagSet){
            NSLog(@"key: %@, value: %@",tag.key,tag.value);
        }
    }
}];
```

Delete Object Tagging

功能说明

删除对象标签

返回结果

参数名称	参数描述	类型
response	删除对象标签返回结果	AWSS3DeleteObjectTaggingOutput*
versionId	标签对象的versionID	NSString*

方法原型

```
- (void)deleteObjectTagging:(AWSS3DeleteObjectTaggingRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3DeleteObjectTaggingOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	是否必须	参数描述	类型
request	是	删除对象标签请求	AWSS3DeleteObjectTaggingRequest*
bucket	是	桶名称	NSString*
key	是	对象名称	NSString*
versionId	否	对象版本号	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSS3ServiceConfiguration *configure = [[AWSS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3DeleteObjectTaggingRequest *request = [[AWSS3DeleteObjectTaggingRequest
alloc] init];
request.bucket = @"ios_test";
request.key = @"test";
[s3 deleteObjectTagging:request
completionHandler:^(AWSS3DeleteObjectTaggingOutput * _Nullable response, NSError
* _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    }else{
        NSLog(@"Success delete the object tagging");
    }
}];
```

Create Multiaprt 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
response	初始化分段上传的返回结果	AWSS3CreateMultipartUploadOutput*
bucket	桶名	NSString*
key	对象名	NSString*
uploadId	分段的Id	NSString*

方法原型

```
- (void)createMultipartUpload:(AWSS3CreateMultipartUploadRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3CreateMultipartUploadOutput *
_Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	初始化分段上传请求	AWSS3CreateMultipartUploadRequest*
bucket	桶名称	NSString*
key	对象名称	NSString*
ACL	对象访问权限	AWSS3ObjectCannedACL(AWSS3ObjectCannedACLPrivate\AWSS3ObjectCannedACLPublicRead\AWSS3ObjectCannedACLAuthenticatedRead)
grantRead	允许该访问对象和该对象的元数据	NSString *
grantReadACP	允许该访问对象ACL	NSString *
grantWriteACP	允许该修改对象ACL	NSString *
metadata	该修改对象的元数据	NSDictionary<NSString *, NSString *>*
storageClass	该对象的存储级别	AWSS3StorageClass, 支持AWSS3StorageClassStandard (标准存储)、AWSS3StorageClassStandardIA (低频存储)、AWSS3StorageClassGlacier (归档存储)

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3CreateMultipartUploadRequest *init_multipart =
[[AWSS3CreateMultipartUploadRequest alloc] init];
init_multipart.bucket = @"ios_test";
init_multipart.key = @"test";
NSString *bucket_name = @"ios_test";
NSString *key_name = @"test";
[s3 createMultipartUpload:init_multipart
completionHandler:^(AWSS3CreateMultipartUploadOutput * _Nullable response,
NSError * _Nullable error) {
```



```
        if(error != nil){
            NSLog(@"Error:%@",error);
        } else{
            NSLog(@"the response upload id is:%@",response.uploadId);
        }
    }
}];
```

Upload Part

功能说明

上传分段

返回结果

参数名称	参数描述	类型
response	分段上传的返回结果	AWSS3UploadPartOutput*
ETag	上传段的ETag	NSString*

方法原型

```
- (void)uploadPart:(AWSS3UploadPartRequest *)request completionHandler:(void (^ _Nullable)(AWSS3UploadPartOutput * _Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
uploadrequest	分段上传请求	AWSS3UploadPartRequest*
body	上传内容	id
bucket	桶名	NSString*
key	对象名称	NSString*
contentLength	上传内容大小	NSNumber*
uploadId	上传ID,用于标识正在进行分段上传的部分	NSString*
partNumber	上传的第几个段(从1开始)	NSNumber*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration = configure;
```

```

AWSS3 *s3 = [AWSS3 defaultS3];
__block NSMutableArray<AWSS3CompletedPart *> *parts = [NSMutableArray array];
__block NSString *bucket_name = @"ios_test";
__block NSString *key_name = @"test";
__block NSString *upload_id = @"uploadId";
for(int i = 1; i < 3 ; ++i){
    AWSS3UploadPartRequest *uploadrequest = [[AWSS3UploadPartRequest alloc]
init];
    uploadrequest.bucket = bucket_name;
    uploadrequest.key = key_name;
    uploadrequest.uploadId = upload_id;
    uploadrequest.partNumber = @(i);
    NSString *filepath = [NSHomeDirectory()
stringByAppendingPathComponent:@"test_file_5m"];
    NSData *content = [NSData dataWithContentsOfFile:filepath];
    uploadrequest.body = content;
    uploadrequest.contentLength = [NSNumber numberWithInt:content.length];
    NSLog(@"the content length:%lu",content.length);
    [s3 uploadPart:uploadrequest completionHandler:^(AWSS3UploadPartOutput *
_Nullable response, NSError * _Nullable error) {
        if(error != nil){
            NSLog(@"Error:%@",error);
        } else{
            AWSS3CompletedPart *complete_part = [[AWSS3CompletedPart alloc]
init];
            complete_part.ETag = response.ETag;
            complete_part.partNumber = @(i);
            [parts addObject: complete_part];
        }
    }]];
}

```

Compelete Multipart Upload

功能说明

完成分段上传

返回结果

参数名称	参数描述	类型
response	分段上传的返回结果	AWSS3UploadPartOutput*
ETag	上传段的ETag	NSString*
bucket	桶名	NSString*
key	上传对象名	NSString*
versionId	版本号	NSString*

方法原型

```
- (void)completeMultipartUpload:(AWSS3CompleteMultipartUploadRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3CompleteMultipartUploadOutput *
_Nullable response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
uploadrequest	分段上传请求	AWSS3UploadPartRequest*
multipartUpload	多段请求的信息	AWSS3CompletedMultipartUpload*
bucket	桶名	NSString*
key	对象名称	NSString*
uploadId	初始化分段上传的ID	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
NSMutableArray<AWSS3CompletedPart *> *parts = [NSMutableArray array];
AWSS3CompletedMultipartUpload *completed_upload = [[AWSS3CompletedMultipartUpload
alloc] init];
completed_upload.parts = parts;
AWSS3CompleteMultipartUploadRequest *completed_multipart_request =
[[AWSS3CompleteMultipartUploadRequest alloc] init];
completed_multipart_request.bucket = @"ios_test";
completed_multipart_request.key = @"test";
completed_multipart_request.uploadId = @"uploadId";
completed_multipart_request.multipartUpload = completed_upload;
[s3 completeMultipartUpload:completed_multipart_request
completionHandler:^(AWSS3CompleteMultipartUploadOutput * _Nullable response,
NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error: fail to completed multipart%@",error);
    } else{
        NSLog(@"Success completmultipart upload, the ETag: %@",response.ETag);
    }
}];
```

List part

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
response	列举分段上传碎片的返回结果	AWSS3ListMultipartUploadsOutput*
bucket	桶名	NSString*
keyMarker	列举的起始key	NSString*
uploadIdMarker	列举的起始uploadIdMarker	NSString*
maxUploads	最大返回段数量	NSNumber*
nextKeyMarker	下一次列举的起始key	NSString*
nextUploadIdMarker	下一次列举的起始UploadIdMarker	NSString*
prefix	列举前缀	NSString*
uploads	分段上传的段	NSArray<AWSS3MultipartUpload*>*

方法原型

```
- (void)listMultipartUploads:(AWSS3ListMultipartUploadsRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3ListMultipartUploadsOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	分段上传list请求	AWSS3ListMultipartUploadsRequest*
bucket	桶名	NSString*
keyMarker	与uploadIdMarker一起指定list的起始位置	NSString*
maxUploads	分段碎片最大返回数	NSNumber*
prefix	指定前缀开头的key	NSString*
uploadIdMarker	与keyMarker一起指定list的起始位置	NSString*

示例

```
AWSSStaticCredentialsProvider *credential = [[AWSSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3ListMultipartUploadsRequest *request = [[AWSS3ListMultipartUploadsRequest
alloc] init];
    request.bucket = @"ios_test";
    [s3 listMultipartUploads:request
completionHandler:^(AWSS3ListMultipartUploadsOutput * _Nullable response, NSError
* _Nullable error) {
        if(error != nil){
            NSLog(@"Error:%@",error);
        } else{
            for(AWSS3MultipartUpload *upload in response.uploads){
                NSLog(@"key:%@",upload.key);
                NSLog(@"upload_id:%@",upload.uploadId);
            }
        }
    }
    }];
```

Abort part

功能说明

取消对象分段上传，删除碎片

返回结果

参数名称	参数描述	类型
response	取消返回结果	AWSS3AbortMultipartUploadOutput*

方法原型

```
- (void)abortMultipartUpload:(AWSS3AbortMultipartUploadRequest *)request
completionHandler:(void (^ _Nullable)(AWSS3AbortMultipartUploadOutput * _Nullable
response, NSError * _Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	分段上传取消请求	AWSS3AbortMultipartUploadRequest*
bucket	桶名	NSString*
key	对象名	NSString*
uploadId	分段上传段的ID	NSString*

示例

```

AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultS3];
AWSS3AbortMultipartUploadRequest *request = [[AWSS3AbortMultipartUploadRequest
alloc] init];
request.bucket = @"ios_test";
request.key = @"test";
request.uploadId = @"uploadId";

[s3 abortMultipartUpload:request
completionHandler:^(AWSS3AbortMultipartUploadOutput * _Nullable response, NSError
* _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"success abort the upload, id:%@",request.uploadId);
    }
}];

```

Restore Object

功能说明

归档对象解冻

返回结果

参数名称	参数描述	类型
response	归档解冻结果	AWSS3RestoreObjectOutput*

方法原型

```
- (void)restoreObject:(AWSS3RestoreObjectRequest *)request completionHandler:
(void (^ _Nullable)(AWSS3RestoreObjectOutput * _Nullable response, NSError *
_Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	归档解冻请求	AWSS3RestoreObjectRequest*
bucket	桶名	NSString*
key	对象名	NSString*
restoreRequest	解冻对象参数	AWSS3RestoreRequest*
versionId	对象版本号	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3RestoreRequest *restore_request = [[AWSS3RestoreRequest alloc] init];
restore_request.days = @1;

AWSS3RestoreObjectRequest *request = [[AWSS3RestoreObjectRequest alloc] init];
request.bucket = @"ios_test";
request.key = @"test";
request.restoreRequest = restore_request;
[s3 restoreObject:request completionHandler:^(AWSS3RestoreObjectOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success restore object");
    }
}];
```

Presign Url

功能说明

请求生成一个临时的预签名下载连接的url,用户可使用该url访问桶内对象

返回结果

参数名称	参数描述	类型
presignedURL	生成的预签名链接	AWSTask<NSURL *> *

方法原型

```
(AWSTask<NSURL *> *)getPreSignedURL:(AWSS3GetPreSignedURLRequest *)getPreSignedURLRequest;
```

参数说明

参数名称	参数描述	类型
request	预签名请求	AWSS3GetObjectRequest*
bucket	桶名	NSString*
key	对象名	NSString*
expiration	预签名链接过期时间	NSDate*
method	预签名链接的HTTP方法	HttpMethod*

示例

```
AWSStaticCredentialsProvider *credential = [[AWSStaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWSEndpoint *endpoint = [[AWSEndpoint alloc] initWithURLString:@"endpoint"];
AWSServiceConfiguration *configure = [[AWSServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWSServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3GetPreSignedURLRequest *request = [[AWSS3GetPreSignedURLRequest
alloc] init];
request.key = @"test_file";
request.bucket = @"ios_test";
NSTimeInterval expire = 1000;
request.HTTPMethod = AWSHTTPMethodGET;
request.expires = [NSDate dateWithTimeIntervalSinceNow:expire];
AWSS3PreSignedURLBuilder *s3pre = [AWSS3PreSignedURLBuilder
defaultS3PreSignedURLBuilder];
[[s3pre getPreSignedURL:request] continueWithBlock:^(AWSTask<NSURL*> *task){
    if (task.error){
        NSLog(@"Error :%@", task.error);
    } else{
        NSURL *presignedURL = task.result;
        NSLog(@"Presigned URL :%@", presignedURL);
    }
    return nil;
}];
```


Copy Object

功能说明

桶源桶拷贝对象到目标桶(源桶需要读权限,目标桶需要写权限)

返回结果

参数名称	参数描述	类型
response	拷贝对象返回结果	AWSS3ReplicateObjectOutput*
AWSS3ReplicateObjectResult	包含目标对象信息(ETag、lastModified)	AWSS3ReplicateObjectResult*
versionId	目标对象的版本	NSString*

方法原型

```
- (void)replicateObject:(AWSS3ReplicateObjectRequest *)request completionHandler:
(void (^ _Nullable)(AWSS3ReplicateObjectOutput * _Nullable response, NSError *
_Nullable error))completionHandler;
```

参数说明

参数名称	参数描述	类型
request	拷贝对象请求	AWSS3ReplicateObjectRequest*
bucket	目标桶名	NSString*
key	目标对象名	NSString*
ACL	目标对象的ACL	AWSS3ObjectCannedACL(AWSS3ObjectCannedACLPrivate\AWSS3ObjectCannedACLPublicRead\AWSS3ObjectCannedACLAuthenticatedRead)
grantRead	允许访问目标对象的数据和元数据	NSString*
grantReadACP	允许访问目标对象的ACL	NSString*
grantWriteACP	允许修改目标对象的ACL	NSString*
metadata	目标对象的元数据	NSDictionary<NSString *, NSString *>*
metadataDirective	从源对象复制元数据还是用请求中提供的元数据替换元数据	AWSS3MetadataDirective(AWSS3MetadataDirectiveCopy\AWSS3MetadataDirectiveReplace)
storageClass	目标对象的存储级别	AWSS3StorageClass, 支持AWSS3StorageClassStandard (标准存储)、AWSS3StorageClassStandardIa (低频存储)、AWSS3StorageClassGlacier (归档存储)
replicateSource	源对象(格式: {bucket}/{key} 或者{bucket}/{key}?versionId={versionId})	NSString*

示例

```
AWSS3StaticCredentialsProvider *credential = [[AWSS3StaticCredentialsProvider
alloc] initWithAccessKey:@"accesskey" secretKey:@"secretkey"];
AWS3Endpoint *endpoint = [[AWS3Endpoint alloc] initWithURLString:@"endpoint"];
AWS3ServiceConfiguration *configure = [[AWS3ServiceConfiguration
alloc] initWithRegion:AWSRegionUSEast1 endpoint:endpoint
credentialsProvider:credential];
[AWS3ServiceManager defaultManager].defaultServiceConfiguration =
configure;
AWSS3 *s3 = [AWSS3 defaultManager];
AWSS3ReplicateObjectRequest *request = [[AWSS3ReplicateObjectRequest alloc]
init];
```

```
request.bucket = @"ios_test";
request.key = @"file";
NSString *source_bucket = @"ios_test";
NSString *source_key = @"new_file";
request.replicateSource = [NSString stringWithFormat:@"%@/%@", source_bucket,
source_key];
[s3 replicateObject:request completionHandler:^(AWSS3ReplicateObjectOutput *
_Nullable response, NSError * _Nullable error) {
    if(error != nil){
        NSLog(@"Error:%@",error);
    } else{
        NSLog(@"Success copy the obj:%@",request.key);
    }
}];
```