

# EXHIBIT 18

---

# Amazon Simple Storage Service

## API Reference

API Version 2006-03-01



## Amazon Simple Storage Service API Reference

### PUT Object

---

## PUT Object

### Description

This implementation of the `PUT` operation adds an object to a bucket. You must have `WRITE` permissions on a bucket to add an object to it.

Amazon S3 never adds partial objects; if you receive a success response, Amazon S3 added the entire object to the bucket.

Amazon S3 is a distributed system. If it receives multiple write requests for the same object simultaneously, it overwrites all but the last object written. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer or use versioning instead.

To ensure that data is not corrupted traversing the network, use the `Content-MD5` header. When you use this header, Amazon S3 checks the object against the provided MD5 value and, if they do not match, returns an error. Additionally, you can calculate the MD5 while putting an object to Amazon S3 and compare the returned `ETag` to the calculated MD5 value.



#### Note

To configure your application to send the Request Headers prior to sending the request body, use the `100-continue` HTTP status code. For `PUT` operations, this helps you avoid sending the message body if the message is rejected based on the headers (e.g., because of authentication failure or redirect). For more information on the `100-continue` HTTP status code, go to Section 8.2.3 of <http://www.ietf.org/rfc/rfc2616.txt>.

### Versioning

If you enable versioning for a bucket, Amazon S3 automatically generates a unique version ID for the object being stored. Amazon S3 returns this ID in the response using the `x-amz-version-id` response header. If versioning is suspended, Amazon S3 always uses `null` as the version ID for the object stored. For more information about returning the versioning state of a bucket, see [GET Bucket versioning \(p. 64\)](#).

If you enable versioning on a bucket, when Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.

To see sample requests that use versioning, see [Sample Request \(p. 158\)](#).

### Reduced Redundancy Storage

Reduced redundancy storage (RRS) enables customers to reduce their costs by storing non-critical, reproducible data at lower levels of redundancy than Amazon S3's standard storage. RRS provides a cost-effective, highly available solution for distributing or sharing content that is durably stored elsewhere, or for storing thumbnails, transcoded media, or other processed data that can be easily reproduced. The RRS option stores objects on multiple devices across multiple facilities, providing 400 times the durability of a typical disk drive, but does not replicate objects as many times as standard Amazon S3 storage. Thus, using RRS is even more cost effective.

To store an object using reduced redundancy, set the `x-amz-storage-class` request header to `REDUCED_REDUNDANCY`. The default value is `STANDARD`.