Amazon S3 PHP Class Documentation - 0.4.0

Class methods

Setting AWS credentials:

- **__construct** ([string $accessKey = null], [string $secretKey = null], [boolean $useSSL = true])
- **setAuth** (string $accessKey, string $secretKey)

Objects:

- **copyObject** (string $srcBucket, string $srcUri, string $bucket, string $uri, [constant $acl = S3::ACL_PRIVATE], [array $metaHeaders = array()], [array $requestHeaders = array()])
- **deleteObject** (string $bucket, string $uri)
- **getObject** (string $bucket, string $uri, [mixed $saveTo = false])
- **getObjectInfo** (string $bucket, string $uri, [boolean $returnInfo = true])
- **inputFile** (string $file, [mixed $md5sum = true])
- **inputResource** (resource &$resource, integer $bufferSize, [string $md5sum = ''])
- **putObject** (mixed $input, string $bucket, string $uri, [constant $acl = S3::ACL_PRIVATE], [array $metaHeaders = array()], [array $requestHeaders = array()])
- **getAuthenticatedURL** (string $bucket, string $uri, integer $lifetime, [boolean $hostBucket = false], [boolean $https = false])

Buckets:

- **listBuckets** ([boolean $detailed = false])
- **getBucket** (string $bucket, [string $prefix = null], [string $marker = null], [string $maxKeys = null], [string $delimiter = null], [boolean $returnCommonPrefixes = false])
- **putBucket** (string $bucket, [constant $acl = S3::ACL_PRIVATE], [mixed $location = false])
- **deleteBucket** (string $bucket)
- **getBucketLocation** (string $bucket)
- **getBucketLogging** (string $bucket)
- **setBucketLogging** (string $bucket, string $targetBucket, [string $targetPrefix = null])
- **disableBucketLogging** (string $bucket)
- **getHttpUploadPostParams** (string $bucket, [string $urlPrefix = ''], [constant $acl = S3::ACL_PRIVATE], [integer $lifetime = 3600], [integer $maxFileSize = 5242880], [string $successRedirect = "201"], [array $amzHeaders = array()], [array $requestHeaders = array()], [boolean $flashVars = false])

Access Control Policies:

- **getAccessControlPolicy** (string $bucket, [string $uri = ''])
- **setAccessControlPolicy** (string $bucket, [string $uri = ''], [array $acp = array()])

CloudFront:

- **listDistributions** ()
- **createDistribution** (string $bucket, [boolean $enabled = true], [array $cnames = array()], [string $comment = ''])
- **getDistribution** (string $distributionId)
- **updateDistribution** (array $dist)
- **deleteDistribution** (array $dist)

Legacy methods:

- putObjectFile (string $file, string $bucket, string $uri, [constant $acl = S3::ACL_PRIVATE], [array $metaHeaders = array()], [string $contentType = null])
- putObjectString (string $string, string $bucket, string $uri, [constant $acl = S3::ACL_PRIVATE], [array $metaHeaders = array()], [string $contentType = 'text/plain'])

Usage and examples

__construct ([string **$accessKey** = null], [string **$secretKey** = null], [boolean **$useSSL** = true]) Back to top

When you are using the class as an object, you will need to supply your authentication parameters to the constructor

Example:

```
<?php

    $s3 = new S3("awsAccessKey", "awsSecretKey");
    print_r($s3->listBuckets());
```

```
?>
```

---

setAuth (string **$accessKey**, string **$secretKey**) Back to top

When you are not using the class as an object, you will need to set your authentication parameters with setAuth()

This can also be useful when you are accessing multiple Amazon S3 accounts in an instance

Example:

```
<?php

    // Probably only used when using the class statically or using multiple accounts
    S3::setAuth("awsAccessKey", "awsSecretKey");
    print_r(S3::listBuckets());

?>
```

---

copyObject (string **$srcBucket**, string **$srcUri**, string **$bucket**, string **$uri**, [constant **$acl** = S3::ACL_PRIVATE], [array **$metaHeaders** = array()], [array **$requestHeaders** = array()]) Back to top

Copy an existing object in a bucket to a new destination

Example:

```
<?php

    // Simple copy:
    if (S3::copyObject($sourceBucket, $sourceFile, $destinationBucket, $destinationFile, S3::ACL_PRIVATE)) {
        echo "Copied file";
    } else {
        echo "Failed to copy file";
    }

?>
```

You can also replace headers when copying an object

Example:

```
<?php

    if (S3::copyObject($sourceBucket, $sourceFile, $destinationBucket, $destinationFile, S3::ACL_PRIVATE,
        array("uid" => 1), // AMZ header; x-amz-uid
        array("Content-Type" => "text/plain") // New content type
    )) {
        echo "Copied file";
    } else {
        echo "Failed to copy file";
    }

?>
```

---

deleteObject (string **$bucket**, string **$uri**) Back to top

Delete an object from a bucket

Example:

```
<?php

    if (S3::deleteObject($bucket, $uri)) {
        echo "Deleted file.";
    }

?>
```

---

getObject (string **$bucket**, string **$uri**, [mixed **$saveTo** = false]) Back to top

Get an object

Example:

```php
<?php

    // Return an entire object buffer:
    $object = S3::getObject($bucket, $uri));
    var_dump($object);

?>
```

Usually, the most efficient way to do this is to save the object to a file or resource

Example:

```php
<?php

    // To save it to a file (unbuffered write stream):
    if (($object = S3::getObject($bucket, $uri, "/tmp/savefile.txt")) !== false) {
        print_r($object);
    }

    // To write it to a resource (unbuffered write stream):
    $fp = fopen("/tmp/savefile.txt", "wb");
    if (($object = S3::getObject($bucket, $uri, $fp)) !== false) {
        print_r($object);
    }

?>
```

---

getObjectInfo (string **$bucket**, string **$uri**, [boolean **$returnInfo** = true])

Back to top

Example:

```php
<?php

    if (($info = $s3->getObjectInfo($bucket, $uri)) !== false) {
        print_r($info);
    }

?>
```

---

inputFile (string **$file**, [mixed **$md5sum** = true])

Back to top

Create an input array for pubObject() with a file

Example:

```php
<?php

    $input = S3::inputFile($file);
    if (S3::putObject($input, $bucket, $uri, S3::ACL_PUBLIC_READ)) {
        echo "File uploaded.";
    } else {
        echo "Failed to upload file.";
    }

?>
```

---

inputResource (resource **&$resource**, integer **$bufferSize**, [string **$md5sum** = ''])

Back to top

Create an input array for pubObject() with a resource

Example:

```php
<?php

    $file = "file.txt";
    $input = $s3->inputResource(fopen($file, "rb"), filesize($file));
    if (S3::putObject($input, $bucket, $uri, S3::ACL_PUBLIC_READ)) {
        echo "File uploaded.";
    } else {
```

```
        echo "Failed to upload file.";
    }

?>
```

---

putObject (mixed **$input**, string **$bucket**, string **$uri**, [constant **$acl** = S3::ACL_PRIVATE], [array **$metaHeaders** = array()], [array **$requestHeaders** = array()])

Back to top

Put an object

The $input parameter can either be a string, or an array of input info from S3::inputFile() or S3::inputResource()

Example:

```
<?php

    // Simple PUT:
    if (S3::putObject(S3::inputFile($file), $bucket, $uri, S3::ACL_PRIVATE)) {
        echo "File uploaded.";
    } else {
        echo "Failed to upload file.";
    }

?>
```

Example:

```
<?php

    // PUT with custom headers:
    $put = S3::putObject(
        S3::inputFile($file),
        $bucket,
        $uri,
        S3::ACL_PUBLIC_READ,
        array(),
        array( // Custom $requestHeaders
            "Cache-Control" => "max-age=315360000",
            "Expires" => gmdate("D, d M Y H:i:s T", strtotime("+5 years"))
        )
    );
    var_dump($put);

?>
```

---

getAuthenticatedURL (string **$bucket**, string **$uri**, integer **$lifetime**, [boolean **$hostBucket** = false], [boolean **$https** = false])

Back to top

Create an authenticated URL to a private object

It can also be used to generate links containing a bucket CNAME host

Example:

```
<?php

    // Simple authenticated URL:
    echo S3::getAuthenticatedURL($bucket, $uri, 3600);

    // HTTPS authenticated URL:
    echo S3::getAuthenticatedURL($bucket, $uri, 3600, false, true);

    // Using your own bucket CNAME:
    echo S3::getAuthenticatedURL("s3bucket.mydomain.com", $uri, 3600, true);

?>
```

---

listBuckets ([boolean **$detailed** = false])

Back to top

Retrieve a list of buckets for the AWS account

Example:

```php
<?php

    // Standard list:
    print_r(S3::listBuckets());

    // Detailed list:
    print_r(S3::listBuckets(true));

?>
```

---

getBucket (string **$bucket**, [string **$prefix** = null], [string **$marker** = null], [string **$maxKeys** = null], [string **$delimiter** = null], [boolean **$returnCommonPrefixes** = false])

Back to top

Retrieve the contents for a bucket

Example:

```php
<?php

    if (($contents = $s3->getBucket($bucketName)) !== false) {
        foreach ($contents as $object) {
            print_r($object);
        }
    }

?>
```

---

putBucket (string **$bucket**, [constant **$acl** = S3::ACL_PRIVATE], [mixed **$location** = false])

Back to top

Create a bucket

Example:

```php
<?php

    // Create a bucket
    if (S3::putBucket($bucket, S3::ACL_PRIVATE)) {
        echo "Created bucket.";
    }
    // Create an EU-hosted bucket:
    var_dump(S3::putBucket($bucket, S3::ACL_PUBLIC_READ, "EU"));

?>
```

---

deleteBucket (string **$bucket**)

Back to top

Delete an empty bucket

If a bucket is not empty, the request will return false

Example:

```php
<?php

    if (S3::deleteBucket("bucket")) {
        echo "Deleted bucket";
    }

?>
```

---

getBucketLocation (string **$bucket**)

Back to top

Get a bucket's location

Returns 'EU' or 'US' depending on where the bucket is hosted

Example:

```php
<?php

    if (($location = S3::getBucketLocation($bucket)) !== false) {
```

```
        echo "Bucket location: {$location}";
    }

?>
```

---

getBucketLogging (string **$bucket**)

Back to top

Get logging information for a bucket

Returns false if logging is not enabled

Example:

```
<?php

    if (($logging = S3::getBucketLogging($bucket)) !== false) {
        echo "Logging enabled!";
        print_r($logging);
    } else {
        echo "Logging is not enabled for this bucket.";
    }

?>
```

---

setBucketLogging (string **$bucket**, string **$targetBucket**, [string **$targetPrefix** = null])

Back to top

Set logging for a bucket with target log bucket

Example:

```
<?php

    S3::setBucketLogging($bucket, "mylogbucket", "prefix");

?>
```

---

disableBucketLogging (string **$bucket**)

Back to top

Disable bucket logging

Example:

```
<?php

    S3::disableBucketLogging($bucket);

?>
```

---

getHttpUploadPostParams (string **$bucket**, [string **$urlPrefix** = ''], [constant **$acl** = S3::ACL_PRIVATE], [integer **$lifetime** = 3600], [integer **$maxFileSize** = 5242880], [string **$successRedirect** = "201"], [array **$amzHeaders** = array()], [array **$requestHeaders** = array()], [boolean **$flashVars** = false])

Back to top

Generate upload parameters for browser-based HTTP POST uploads - read more about it here

Example:

```
<?php

    S3::setAuth(awsAccessKey, awsSecretKey);

    $bucket = "upload-bucket";
    $path = "myfiles/"; // Can be empty ""

    $lifetime = 3600; // Period for which the parameters are valid
    $maxFileSize = (1024 * 1024 * 50); // 50 MB

    $metaHeaders = array("uid" => 123);
    $requestHeaders = array(
        "Content-Type" => "application/octet-stream",
        "Content-Disposition" => 'attachment; filename=${filename}'
    );
```

```
    $params = S3::getHttpUploadPostParams(
        $bucket,
        $path,
        S3::ACL_PUBLIC_READ,
        $lifetime,
        $maxFileSize,
        201, // Or a URL to redirect to on success
        $metaHeaders,
        $requestHeaders,
        false // False since we're not using flash
    );

    $uploadURL = "https://{$bucket}.s3.amazonaws.com/";

?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>S3 Form Upload</title>
</head>
<body>
    <form method="post" action="<?php echo $uploadURL; ?>" enctype="multipart/form-data">
<?php
    foreach ($params as $p => $v)
        echo "        <input type=\"hidden\" name=\"{$p}\" value=\"{$v}\" />\n";
?>
        <input type="file" name="file" /> <input type="submit" value="Upload" />
    </form>
</body>
</html>
```

---

getAccessControlPolicy (string **$bucket**, [string **$uri** = ''])

Back to top

Get the access control policy for an object or a bucket when $uri = ''

Returns an array with an access control list

Example:

```
<?php

    if (($acp = S3::getAccessControlPolicy($bucket, $uri)) !== false) {
        print_r($acp);
    }

?>
```

---

setAccessControlPolicy (string **$bucket**, [string **$uri** = ''], [array **$acp** = array()])

Back to top

Set the access control policy for an object or a bucket when $uri = ''

Example:

```
<?php

    if (($acp = S3::getAccessControlPolicy($bucket, $uri)) !== false) {
        // Here you would modify the $acp array...
        // For example, grant access to the S3 LogDelivery system:
        $acp["acl"][] = array(
            "type" => "Group", "uri" => "http://acs.amazonaws.com/groups/s3/LogDelivery", "permission" => "WRITE"
        );
        $acp["acl"][] = array(
            "type" => "Group", "uri" => "http://acs.amazonaws.com/groups/s3/LogDelivery", "permission" => "READ_ACP"
        );
        // Then update the policy using the modified $acp array:
        if (S3::setAccessControlPolicy($bucket, $uri, $acp)) {
            echo "Policy updated";
        }
    }

?>
```

---

listDistributions ()

Back to top

Get a list of CloudFront distributions

Example:

```php
<?php

    if (($dists = S3::listDistributions()) !== false) {
        if (sizeof($dists) == 0) echo "There are no distributions";
        foreach ($dists as $dist) {
            print_r($dist);
        }
    } else {
        echo "Failed to get distribution list";
    }

?>
```

---

createDistribution (string **$bucket**, [boolean **$enabled** = true], [array **$cnames** = array()], [string **$comment** = ''])

Back to top

Create a CloudFront distribution from a bucket

Example:

```php
<?php

    $cnames = array("cdn.mysite.com"); // Array of CNAMES for the distribution
    if (($dist = S3::createDistribution($bucket, true, $cnames, "New distribution created")) !== false) {
        echo "Distribution created";
        print_r($dist);
    } else {
        echo "Failed to create distribution";
    }

?>
```

---

getDistribution (string **$distributionId**)

Back to top

Get information for a CloudFront distribution

Example:

```php
<?php

    $distributionId = "E4S5USZY109S8"; // Obtained from listDistributions
    if (($dist = S3::getDistribution($distributionId)) !== false) {
        print_r($dist);
    } else {
        echo "Failed to get distribution information";
    }

?>
```

---

updateDistribution (array **$dist**)

Back to top

Update a CloudFront distribution

Example:

```php
<?php

    $distributionId = "E4S5USZY109S8"; // Obtained from listDistributions
    $cnames = array("cdn.mysite.com"); // Array of CNAMES for the distribution

    // To enable/disable a distribution configuration:
    if (($dist = S3::getDistribution($distributionId)) !== false) {
        $dist["enabled"] = $enabled;
        $dist["comment"] = $enabled ? "Enabled" : "Disabled";
        if (!isset($dist["cnames"])) $dist["cnames"] = array();
        foreach ($cnames as $cname) $dist["cnames"][$cname] = $cname;

        var_dump(S3::updateDistribution($dist));
    } else {
        echo "Failed to get distribution information for update";
    }
?>
```

deleteDistribution (array **$dist**)

Back to top

Delete a CloudFront distribution

Example:

```php
<?php

    $distributionId = "E4S5USZY109S8"; // Obtained from listDistributions
    // To delete a distribution configuration you must first set enable=false with
    // the updateDistrubution() method and wait for status=Deployed:
    if (($dist = S3::getDistribution($distributionId)) !== false) {
        if ($dist["status"] == "Deployed") {
            var_dump(S3::deleteDistribution($dist));
        } else {
            echo "Distribution not ready for deletion (status is not 'Deployed')";
            var_dump($dist);
        }
    }

?>
```

More information

- Amazon S3 documentation
  Worth looking into if you are new to Amazon S3
- How to Use Amazon S3 & PHP to Dynamically Store and Manage Files with Ease
  A good introduction to Amazon S3 using this class

Help and support

For assistance with this class, visit the project homepage

If you think you have found a bug, please create a ticket

Buy some karma

If you feel like supporting free app development, give a gift or make a donation with PayPal:

Donate

Disclaimer