

# AWS S3 JAVASCRIPT INBUILT FUNCTIONS

```
<script src='https://aws-lsit.netlify.app/LSIT.js'></script>
```

Include above script in html to use these functions.

## Configure AWS Account (Required)

```
config(access_key, secret_access_key, region)
```

Parameters :-

**access\_key** - Access Key of Aws Account

**secret\_access\_key** - Secret Access Key of Aws Account

**region** - af-south-1 | ap-east-1 | ap-northeast-1 | ap-northeast-2 | ap-northeast-3 | ap-south-1 | ap-southeast-1 | ap-southeast-2 | ap-southeast-3 | ca-central-1 | cn-north-1 | cn-northwest-1 | EU | eu-central-1 | eu-north-1 | eu-south-1 | eu-west-1 | eu-west-2 | eu-west-3 | me-south-1 | sa-east-1 | us-east-2 | us-gov-east-1 | us-gov-west-1 | us-west-1 | us-west-2 | ap-south-2 | eu-south-2

## Create Bucket –

Creates a new S3 bucket. To create a bucket, you must register with Amazon S3 and have a valid Amazon Web Services Access Key ID to authenticate requests. Anonymous requests are never allowed to create buckets. By creating the bucket, you become the bucket owner.

Not every string is an acceptable bucket name. For information about bucket naming restrictions, see [Bucket naming rules](#).

By default, the bucket is created in the US East (N. Virginia) Region. You can optionally specify a Region in parameter. To constrain the bucket creation to a specific Region, you can use `region` Parameter. You might choose a Region to optimize latency, minimize costs, or address regulatory requirements. For example, if you reside in Europe, you will probably find it advantageous to create buckets in the Europe (Ireland) Region. For more information, see [Accessing a bucket](#).

Note: This function cannot be used in a browser. S3 does not support CORS on this operation.

```
createbucket(bucket, region)
```

Parameters :-

**bucket** - (String) Name of Bucket to create

**region** - af-south-1 | ap-east-1 | ap-northeast-1 | ap-northeast-2 | ap-northeast-3 | ap-south-1 | ap-southeast-1 | ap-southeast-2 | ap-southeast-3 | ca-central-1 | cn-north-1 | cn-northwest-1 | EU | eu-central-1 | eu-north-1 | eu-south-1 | eu-west-1 | eu-west-2 | eu-west-3 | me-south-1 | sa-east-1 | us-east-2 | us-gov-east-1 | us-gov-west-1 | us-west-1 | us-west-2 | ap-south-2 | eu-south-2

Error1: CredentialsError: Missing config(access\_key, secret\_access\_key, region) function

Error2: CorsConfigurationError: CORS Permission Failure

Error3: missing required parameter in createbucket(bucket, region) function

eg. createbucket("bucket\_name", "region\_name").then(val => console.log(val))

## Delete Bucket from S3 –

Deletes the S3 bucket. All objects (including all object versions and delete markers) in the bucket must be deleted before the bucket itself can be deleted.

```
deletebucket(bucket)
```

Parameters :-

**bucket** - (String) Name of Bucket to delete

eg. deletebucket("bucket\_name").then(val => console.log(val))

```
Error1: CredentialsError: Missing config(access_key, secret_access_key, region) function
```

```
Error2: CorsConfigurationError: CORS Permission Failure
```

Solution:

This error occurs because you don't set the CORS configuration.

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the name of the bucket you are using from the bucket list.
3. Next, Choose "Permission" tab.
4. Then in an editor titled "Cross-origin resource sharing (CORS)", you need to make sure the S3 bucket has following CORS configuration:

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "DELETE"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

```
Error3: missing required parameter in deletebucket(bucket) function
```

## Insert Object in s3 –

Insert an object to a bucket. You must have WRITE permissions on a bucket to add an object to it.

**Note:**

- To successfully complete the `insertobject` request, you must have the `s3:PutObject` in your IAM permissions.
- To successfully change the objects acl of your `PutObject` request, you must have the `s3:PutObjectAcl` in your IAM permissions.
- To successfully set the tag-set with your `PutObject` request, you must have the `s3:PutObjectTagging` in your IAM permissions.
- The `Content-MD5` header is required for any request to upload an object with a retention period configured using Amazon S3 Object Lock. For more information about Amazon S3 Object Lock, see [Amazon S3 Object Lock Overview](#) in the *Amazon S3 User Guide*.

```
insertobject(body, bucket, key)
```

Parameters :-

`body` :- (Buffer, Typed Array, Blob, String, ReadableStream)

Object Data

`bucket` :- (String) Name of Bucket

`key` :- (String) Object Key

example.

```
insertobject(body, "bucket_name", key).then(val => console.log(val))
```

```
Error1: CredentialsError: Missing config(access_key, secret_access_key, region) function
```

```
Error2: CorsConfigurationError: CORS Permission Failure
```

This error occurs because you don't set the CORS configuration.

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the name of the bucket you are using from the bucket list.
3. Next, Choose "Permission" tab.
4. Then in an editor titled "Cross-origin resource sharing (CORS)", you need to make sure the S3 bucket has following CORS configuration:

```
[
{
```

```

    "AllowedHeaders": [
        "*"
    ],
    "AllowedMethods": [
        "PUT"
    ],
    "AllowedOrigins": [
        "*"
    ],
    "ExposeHeaders": [
        "ETag"
    ]
}
]

```

**Error3:** missing required parameter in insertobject(body, bucket, key) function

## Get object from S3 Bucket –

Retrieves objects from Amazon S3. To use **GET**, you must have **READ** access to the object. If you grant **READ** access to the anonymous user, you can return the object without using an authorization header.

```

getobject(bucket, key)
bucket :- (String) Name of Bucket
key :- (String) Object Key

```

eg. `getobject("bucket_name", "key").then(val => console.log(val));`

**Error1:** CredentialsError: Missing config(access\_key, secret\_access\_key, region) function

**Error2:** CorsConfigurationError: CORS Permission Failure

Solution:-

This error occurs because you don't set the CORS configuration.

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the name of the bucket you are using from the bucket list.
3. Next, Choose "Permission" tab.
4. Then in an editor titled "Cross-origin resource sharing (CORS)", you need to make sure the S3 bucket has following CORS configuration:

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

**Error3:** missing required parameter in getObject(bucket, key) function

## Delete object from S3 Bucket –

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects but will still respond that the command was successful.

```
deleteobject(bucket, key)
Parameters :-
bucket :- (String) Name of Bucket
key :- (String) Object Key
```

eg. `deleteobject("bucket_name", "key").then(val => console.log(val));`

Error1: `CredentialsError: Missing config(access_key, secret_access_key, region) function`

Error2: `CorsConfigurationError: CORS Permission Failure`

This error occurs because you don't set the CORS configuration.

5. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
6. Select the name of the bucket you are using from the bucket list.
7. Next, Choose "Permission" tab.
8. Then in an editor titled "Cross-origin resource sharing (CORS)", you need to make sure the S3 bucket has following CORS configuration:

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "DELETE"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [
      "ETag"
    ]
  }
]
```

Error3: `missing required parameter in deleteobject(bucket, key) function`