

OBJECT STORAGE API

AUTHENTICATION

All API requests require basic authentication using a username and password. Include credentials in the `-u` flag of the cURL command:

```
-u <Username>:<Password>
```

1. CREATE A BUCKET

Endpoint:

```
PUT /_admin/manage/tenants/{tenant_Name}/domains/  
    {domain_Name}/buckets/{Bucket_Name}
```

Request:

```
curl -i -X PUT -u <Username>:<Password>  
<S3_Endpoint_URL>/_admin/manage/tenants/<tenant_Name>/  
domains/<domain_Name>/buckets/<Bucket_Name>
```

Description: Creates a new bucket within a specified domain and tenant.

Response:

```
HTTP/1.1 201 Created  
{  
  "message": "Bucket created successfully",  
  "bucket": "test-bucket"  
}
```

Error Codes:

400 Bad Request - Invalid bucket name.

401 Unauthorized - Authentication failure.

500 Internal Server Error - Server encountered an unexpected error.

2. LIST BUCKETS

Endpoint:

```
GET /_admin/manage/tenants/{tenant_Name}/domains/  
{domain_Name}/buckets
```

Request:

```
curl -i -u <Username>:<Password> <S3_Endpoint_URL>/  
_admin/manage/tenants/<tenant_Name>/domains/  
<domain_Name>/buckets
```

Description: Retrieves a list of all buckets in the specified domain.

Response:

```
HTTP/1.1 200 OK  
{  
  "buckets": ["bucket1", "bucket2", "bucket3"]  
}
```

Error Codes:

401 Unauthorized - Invalid authentication.

500 Internal Server Error - Database or system failure.

3. UPLOAD A FILE

Endpoint:

```
PUT /{Bucket_Name}/{File_Name}
```

Request:

```
curl -i -X PUT -u <Username>:<Password> --data-binary  
"@file.txt" <S3_Endpoint_URL>/<Bucket_Name>/file.txt
```

Description: Uploads a file to the specified bucket.

Response:

```
HTTP/1.1 200 OK  
{  
  "message": "File uploaded successfully",  
  "file": "file.txt"  
}
```

Error Codes:

400 Bad Request - File format not supported.

403 Forbidden - Insufficient permissions.

404 Not Found - Bucket does not exist.

4. LIST OBJECTS IN A BUCKET

Endpoint:

```
GET /{Bucket_Name}/
```

Request:

```
curl -i -u <Username>:<Password> <S3_Endpoint_URL>/  
<Bucket_Name>/
```

Description: Retrieves a list of objects stored in the specified bucket.

Response:

```
HTTP/1.1 200 OK  
{  
  "objects": ["file1.txt", "file2.txt"]  
}
```

Error Codes:

404 Not Found - Bucket does not exist.

500 Internal Server Error - System error.

5. GET AN OBJECT

Endpoint:

```
GET /{Bucket_Name}/{File_Name}
```

Request:

```
curl -i -u <Username>:<Password> <S3_Endpoint_URL>/  
<Bucket_Name>/file.txt
```

Description: Fetches a specific file from a bucket.

Response:

```
HTTP/1.1 200 OK
{
  "file": "file.txt",
  "content": "<file_data>"
}
```

Error Codes:

404 Not Found - File does not exist.

403 Forbidden - Access denied.

6. DELETE AN OBJECT

Endpoint:

```
DELETE /{Bucket_Name}/{File_Name}
```

Request:

```
curl -i -X DELETE -u <Username>:<Password>
<S3_Endpoint_URL>/<Bucket_Name>/file.txt
```

Description: Deletes a specific file from a bucket.

Response:

```
HTTP/1.1 200 OK
{
  "message": "File deleted successfully"
}
```

Error Codes:

404 Not Found - File not found.

403 Forbidden - No delete permissions.

7. DELETE A BUCKET

Endpoint:

```
DELETE /_admin/manage/tenants/{tenant_Name}/domains/  
{domain_Name}/buckets/{Bucket_Name}?recursive=yes
```

Request:

```
curl -i -X DELETE -u <Username>:<Password>  
"<S3_Endpoint_URL>/_admin/manage/tenants/<tenant_Name>/  
domains/<domain_Name>/buckets/<Bucket_Name>?  
recursive=yes"
```

Description: Deletes a bucket and all its contents.

Response:

```
HTTP/1.1 200 OK  
{  
  "message": "Bucket deleted successfully"  
}
```

Error Codes:

404 Not Found - Bucket does not exist.

403 Forbidden - No delete permissions.

500 Internal Server Error - System failure.

8. GET OBJECT METADATA

Endpoint: HEAD /{Bucket_Name}/{File_Name}

Request:

```
curl -i -X HEAD -u <Username>:<Password>  
<S3_Endpoint_URL>/<Bucket_Name>/file.txt
```

Description: Fetches metadata for a specific file without downloading it.

Response:

200 OK: Metadata retrieved successfully.

```
HTTP/1.1 200 OK  
x-amz-meta-custom-header: example-value  
Content-Length: 1024
```

Error Codes:

403 Forbidden: Insufficient permissions.

```
HTTP/1.1 403 Forbidden  
<Error><Code>AccessDenied</Code><Message>Access Denied</  
Message></Error>
```

404 Not Found: File does not exist.

```
HTTP/1.1 404 Not Found  
<Error><Code>NoSuchKey</Code><Message>The specified key  
does not exist.</Message></Error>
```

9. COPY AN OBJECT

Endpoint: PUT /{Bucket_Name}/{New_File_Name}

Request:

```
curl -i -X PUT -u <Username>:<Password> -H "x-amz-copy-source: /<Bucket_Name>/file.txt" <S3_Endpoint_URL>/<Bucket_Name>/file_copy.txt
```

Description: Copies a file to a new location within the same bucket.

Response:

200 OK: Object copied successfully.

```
HTTP/1.1 200 OK
<CopyObjectResult><LastModified>2025-04-03T12:00:00.000Z</LastModified></CopyObjectResult>
```

Error Codes:

403 Forbidden: Insufficient permissions.

404 Not Found: Source object does not exist.

500 Internal Server Error: Issue on the server side.

10. MOVE AN OBJECT (COPY + DELETE)

Endpoints: PUT /{Bucket_Name}/{New_File_Name} DELETE /{Bucket_Name}/{File_Name}

Request:

```
curl -i -X PUT -u <Username>:<Password> -H "x-amz-copy-source: /<Bucket_Name>/file.txt" <S3_Endpoint_URL>/<Bucket_Name>/file_moved.txt  
curl -i -X DELETE -u <Username>:<Password> <S3_Endpoint_URL>/<Bucket_Name>/file.txt
```

Description: Moves a file by copying it to a new location and deleting the original.

Response:

200 OK: Object moved successfully.

Error Codes:

403 Forbidden: Insufficient permissions.

404 Not Found: Source object does not exist.

500 Internal Server Error: Issue on the server side.

11. GET BUCKET VERSIONING

Endpoint: GET /{Bucket_Name}?versioning

Request:

```
curl -i -u <Username>:<Password> "<S3_Endpoint_URL>/<Bucket_Name>?versioning"
```

Description: Checks if versioning is enabled for a bucket.

Response:

200 OK: Versioning status retrieved successfully.

```
HTTP/1.1 200 OK
<VersioningConfiguration><Status>Enabled</Status></
VersioningConfiguration>
```

Error Codes:

403 Forbidden: Insufficient permissions.

404 Not Found: Bucket does not exist.

12. LIST OBJECT VERSIONS

Endpoint: GET /{Bucket_Name}?versions

Request:

```
curl -i -u <Username>:<Password> "<S3_Endpoint_URL>/
<Bucket_Name>?versions"
```

Description: Lists all versions of objects within a bucket.

Response:

200 OK: Object versions retrieved successfully.

```
HTTP/1.1 200 OK
<ListVersionsResult>
  <Version>
    <Key>file.txt</Key>
    <VersionId>123456</VersionId>
    <LastModified>2025-04-03T12:00:00.000Z</LastModified>
  </Version>
</ListVersionsResult>
```

Error Codes:

403 Forbidden: Insufficient permissions.

404 Not Found: Bucket does not exist.

13. GET S3 SERVICE STATUS

Endpoint: GET `/_admin/manage/version`

Request:

```
curl -i -u <Username>:<Password> <S3_Endpoint_URL>/  
_admin/manage/version
```

Description: Retrieves the status of the S3 service.

Response:

200 OK: Service is operational.

```
HTTP/1.1 200 OK  
{ "version": "1.2.3", "status": "running" }
```

Error Codes:

500 Internal Server Error: Issue on the server side.

```
HTTP/1.1 500 Internal Server Error  
<Error><Code>InternalServerError</Code><Message>Unexpected  
server error.</Message></Error>
```

**THIS USER GUIDE FOR API'S WILL HELP YOU USE OBJECT
STORAGE SEAMLESSLY!!!**