Handling of S3 Objects

With the S3 adapter, you can create, delete and list buckets of an S3 endpoint, and retrieve contents and status information of a bucket.

- Storing an Object to a Bucket
- Retrieving an Object From a Bucket
- Listing all Objects Within a Bucket
- Deleting an Object From a Bucket
- Copying an Object Between Buckets
- Retrieving Object Information

Storing an Object to a Bucket

Using the **putObject** operation, you can store an object to a bucket of an S3 endpoint.

- If the bucket is versioned, a new version of the object will be stored.
- If the bucket is not versioned, the object will be updated without any further notice.

There are two ways to define the source of the object to store:

- Provide the object data as a blob in parameter **objectData**.
  ```
  putObject ( bucketName : String, objectName : String, objectData: Blob )
  ```
- Provide a path to a file in the local filesystem (service context) in parameter **filePath**. The contents of this file will be stored then.
  ```
  putObject ( bucketName : String, objectName : String, filePath : String )
  ```

Name	Type	Direction	Description	Allowed Values / Examples
bucketName	String	in	Name of the bucket to put the object to. This must be an existing bucket, otherwise an exception is thrown.	If the bucket is not existing, an exception is thrown.
objectName	String	in	Name of the object to store.	**No** exception will be thrown if the object you want to put is already existing.
filePath	String	in	Path to the file on the local filesystem (in service context) to get the object contents from.	You must specify either **filePath**, or **objectData**. If the file is not available, an exception will be thrown.
objectData	Blob	in	Object data to store as a Blob.	You must specify either **objectData**, or **filePath**.

Retrieving an Object From a Bucket

Using the **getObject** operation, you can retrieve an object from a bucket of an S3 endpoint. You can get the object data

- as a Blob to your service, or
- written to a file in a path relative to the service context.

Retrieve as a Blob

There are two ways to retrieve object data as a Blob depending on whether the bucket is versioned or not.

- Get the object data without specifying a version. This will get the data for unversioned buckets, and the data of the latest version for versioned buckets.
  ```
  getObject ( bucketName : String, objectName : String, objectData: Blob )
  ```
- Get the object data for a specific version.
  ```
  getObject ( bucketName : String, objectName : String, version : String, objectData: Blob )
  ```

Retrieve to a File

There are two ways to retrieve object data into a file depending on whether the bucket is versioned or not.

- Get the object data without specifying a version. This will get the data for unversioned buckets, and the data of the latest version for versioned buckets.
  ```
  getObject ( bucketName : String, objectName : String, filePath :
  String )
  ```
- Get the object data for a specific version.
  ```
  getObject ( bucketName : String, objectName : String, filePath :
  String, version : String )
  ```

If the file path is not valid, an exception will be thrown.

Parameters

Name	Type	Direction	Description	Allowed Values / Examples
bucketName	String	in	Name of the bucket to get an object from.	If the bucket is not existing, an exception is thrown.
objectName	String	in	Name of the object to get.	If the object is not existing, an exception is thrown.
filePath	String	in	Path of the file on the local filesystem (in service context) to store the retrieved object to.	If the file path cannot be accessed, an exception is thrown.
version	String	in	Version of the object to get. If no version is specified, the latest version is retrieved.	If the version is not existing, an exception is thrown.
objectData	Blob	out	Blob containing the contents of the retrieved object data.	

Listing all Objects Within a Bucket

Using the **listObjects** operation of the S3 adapter, you can get a list of all objects that are contained in the given bucket.

- `listObjects ( bucketName: String, objectNames: String[] )`

Name	Type	Direction	Description	Allowed Values / Examples
bucketName	String	in	Name of the bucket to list the contents of.	If the bucket is not existing, an exception is thrown.
objectNames	Array of String	out	List of objects that reside within the given bucket.	

Deleting an Object From a Bucket

Using the **deleteObject** operation of the S3 adapter, you can delete an object from a bucket. The deletion handling differs depending on if the bucket is versioned or not.

Delete an Object From an Unversioned Bucket

If the bucket you want to delete an object from is not versioned, the object is simply deleted. There is no error reported if the object is not existing in the bucket.

Delete an Object From a Versioned Bucket

If the bucket you want to delete an object from is versioned, you can provide a version identifier with the call.

- **Version specified**
  If a version is specified this particular version of the file is deleted from the bucket. If this version cannot be found, no error is reported.
- **No version specified**
  A new delete marker is created and the object is marked as deleted.

Parameters

Name	Type	Direction	Description	Allowed Values / Examples
bucketName	String	in	Name of the bucket to delete an object from.	If the bucket is not existing, an exception is

					thrown.
objectNa me	String	in	Name of the object to delete.		If the object is not existing, **no** exception will be thrown.
version	String	in	For versioned buckets: Specify here the identifier of the object version to delete. If no version is specified for a versioned bucket, a new delete marker is created and the object is marked as deleted (see deleteMark ers).		If the version is not existing, an exception is thrown.

Copying an Object Between Buckets

Using the **copyObject** operation of the S3 adapter, you can copy an object from one bucket to another.

Name	Type	Direction	Description	Allowed Values / Examples
bucketNa me	String	in	Name of the bucket to copy an object from.	If the bucket is not existing, an exception is thrown.
objectName	String	in	Name of the object to be copied.	If the object is not existing, an exception is thrown.
targetBuc ketName	String	in	Name of the bucket to copy the selected object to.	If the bucket is not existing, an exception is thrown. Also, the target bucket must differ from the source bucket.

Retrieving Object Information

Using the **objectStatus** operation of the S3 adapter, you can get some general information on the object including a list of all object versions if the bucket is versioned.

Name	Type	Direction	Description	Allowed Values / Examples
bucketN ame	String	in	Name of the bucket the object resides in.	If the bucket is not existing, an exception is thrown.
objectNa me	String	in	Name of the object to get the object status of.	If the object is not existing, an exception is thrown.
objectSt atus	ObjectSt atus	out	Object containing the status information as described further below.	

The returned status information has the following structure:

Attribute	Type	Description	Examples
name	String	Name of the object.	
size	Integer	Size of the object in bytes.	
type	String	Type of the object as identified by AWS.	**png**, **txt**
creationDate	DateTime	Creation timestamp of the object (last version if versioned).	
latestVersion	String	Number of the latest version.	
versions	Array of String	List of version numbers of the object. The order within the array reflects the order in which the versions have been created. So, **versions[0]** contains the version number of the first version of the object.	
deleteMarke rs	Array of String	List of delete markers of the object.	