Project - BlobStorage

File Name Advanced.cs

Variable declaration for container and Blob prefix.

Function Name

1. CallBlobAdvancedSamples –
2. **Fetching the storage account connection string from app.config**
3. **Create Blob Client**
4. **Declare variable for cloud blob container and service properties.**
5. **Create sample container**
6. **Calling blob service client or fetching blob client**
7. **Calling container**
8. **Blob to assigning the container**
9. **Creating the blobs**
10. **Setting the cross rules.**
11. **If it shows error then calling delete function**
12. **Clearing all values.**
13. CallBlobClientSamples
14. Creating buffer manager
15. List All Containers from storage account without metadata
16. CallContainerSamples
17. Fetch container's properties,metadata and attributes
18. Add metadata to the container.
19. Verifying new added item existing in the containers
20. Setting the container’s public access.
21. Setting container reference using only the URI
22. Change the container’s access to private
23. Delete all unwanted containers
24. CallBlobSamples
25. Fetching the CloudBlockBlob and Writing to the blob creates it on the service
26. check reference to a blob with a request to the server if it not exists then return 404 error.
27. Fetching metada and its attributes.
28. Checking it is blob or not
29. Fetching the continuous token.
30. creating virtual directory name and blob name
31. fetch a reference to the blob
32. Write the blob URI to its contents
33. Set a property or content type on the blob.
34. Set some metadata on the blob.
35. Hierarchical listing virtual directories and blobs
36. Checking the directory is blobDirectory or not. It Call recursively
37. List blobs whose names begin with "0" hierarchically
38. Create a snapshot of a block blob
39. Copy a block blob to another blob in the same container.
40. Upload a blob in blocks.
41. Write an array of random bytes to a block blob
42. CallSasSamples
43. Declare four blob name and its content.
44. Check the container already exists. If it not exists create new one.
45. Create a new shared access policy and assign to its constraints
46. Fetch the container's existing permissions
47. Add new policy to the container's permissions.
48. **ConfigureBlobAnalyticsAsync**
49. ListAllContainers
50. List containers beginning with the specified prefix without metadata.
51. ListContainersWithPrefixAsync

a) List containers beginning with the specified prefix with metadata

1. ListBlobsFromServiceClient
2. The prefix is required when listing blobs from the service client. The prefix must include and the container name.
3. ListBlobsFromServiceClientAsync
4. ListBlobsFromServiceClientAsync
5. CreateSampleContainerAsync
6. AddContainerMetadataAsync
7. Add metadata to the container.
8. SetAnonymousAccessLevelAsync
9. Fetching container’s existing permissions
10. Set the container's public access level
11. PrintContainerPropertiesAndMetadata
12. ManageContainerLeasesAsync
13. create variables for CloudBlobContainers
14. Fetch the lease and delete the leased container.
15. For delete checking four conditions.
16. PrintContainerLeaseProperties
17. DeleteContainersWithPrefixAsync
18. CreateSharedAccessPolicyAsync
19. Create a new shared access policy and assign to its constraints
20. Fetch the container's existing permissions
21. Add new policy to the container's permissions.
22. GetContainerSasUri
23. TestContainerSASAsync
24. PrintBlobPropertiesAndMetadata
25. PrintVirtualDirectoryProperties
26. ListBlobsFlatListingAsync
27. Declare variable for BlobContinuationToken and BlobResultSegment
28. Fetching metada and its attributes.
29. Returens only blobs not vms.
30. Checking it is blob or not
31. Fetching the continuous token.
32. ListBlobsHierarchicalListingAsync
33. Declared variables for BlobContinuationToken, BlobResultSegment
34. Hierarchical listing virtual directories and blobs
35. Checking the directory is blobDirectory. It Call recursively
36. CreateRandomlyNamedBlockBlobAsync
37. Checking CloudBlockBlob is already existing
38. Writing to the blob creates it on the service
39. GetExistingBlobReference
40. GetExistingBlobReferenceAsync

a) check reference to a blob with a request to the server if it not exists then return 404 error.

1. CreateSequentiallyNamedBlockBlobsAsync
2. CreateNestedBlockBlobsAsync
3. the function used to create nested blob structure
4. creating virtual directory name and blob name
5. fetch a reference to the blob
6. Write the blob URI to its contents
7. Set a property or content type on the blob.
8. Set some metadata on the blob.
9. CreateBlockBlobSnapshotAsync
10. Create a new block blob in the container.
11. Add metadata to the blob
12. Upload the blob to create it, with its metadata.
13. Set thread sleeping time
14. specify unique metadata for the snapshot
15. If no metadata is created then base blob's metadata is copied to the snapshot.
16. CopyBlockBlobAsync
17. Fetch a block blob from the container to use as the source.
18. the source blob for the copy operation to prevent another client from modifying it
19. creates an infinite lease.
20. Fetch a reference to a destination blob
21. Checking the source blob exists
22. Fetch the ID of the copy operation
23. Fetch the destination blob's properties
24. Checking the copy status.
25. CopyLargeBlockBlobAsync
26. UploadBlobInBlocksAsync
27. Create an bytes array and specified size.
28. Fetch reference to a new block blob and specify its size.
29. Create a new blob and upload the content.
30. Fetch the blob's block list
31. Read the block list of the newly created.
32. Commit the blocks to a new blob
33. ReadBlockListAsync
34. Fetch the current block blob content
35. Checking its status is Committed
36. Then reading the block blob content from BlockList
37. GetBlobSasUri
38. TestBlobSASAsync
39. UploadByteArrayAsync
40. Write an array of random bytes to a block blob.
41. CorsSample
42. PageRangesSample
43. GetRandomBuffer

Doubt: service properties, pagerangesample, PrintContainerPropertiesAndMetadata, FetchAttributesAsync