**2. Develop for Azure Storage → 2.1 Develop solutions that use Azure Blob Storage → 2.2.1 Set and retrieve properties and metadata**

1. What are blob properties and blob metadata?
2. How do you set blob properties (like content type, cache control)?
3. How do you retrieve blob properties?
4. How do you set custom metadata on a blob?
5. How do you retrieve metadata from a blob?
6. What are best practices when using metadata in Azure Blob Storage?
7. What happens if you overwrite a blob — are properties and metadata preserved?
8. How do you set or update metadata without overwriting the blob content?
9. How do you use Azure SDK (C#, Python) to manage properties and metadata?
10. How can you search or filter blobs by metadata?

---

**1. What are blob properties and blob metadata?**

- **Blob properties** are **system-defined** attributes like Content-Type, Content-Encoding, Cache-Control, and Content-Length.
- **Blob metadata** consists of **user-defined** key-value pairs that describe the blob but do not affect its behavior.

---

**2. How do you set blob properties (like content type, cache control)?**

- When uploading or updating a blob, set properties using `BlobClient.Upload()` with BlobHttpHeaders.
- Example (C# Azure SDK):

```
await blobClient.UploadAsync(fileStream, new BlobHttpHeaders { ContentType = "image/png", CacheControl = "no-cache" });
```

---

**3. How do you retrieve blob properties?**

- Use `BlobClient.GetPropertiesAsync()` method.
- Example (C# Azure SDK):

```
BlobProperties properties = await blobClient.GetPropertiesAsync();
Console.WriteLine(properties.ContentType);
Console.WriteLine(properties.CacheControl);
```

---

**4. How do you set custom metadata on a blob?**

- Use `BlobClient.SetMetadataAsync()` with a dictionary of key-value pairs.
- Example (C# Azure SDK):

```
var metadata = new Dictionary<string, string> { { "author", "john_doe" }, { "category", "images" } };
await blobClient.SetMetadataAsync(metadata);
```

---

**5. How do you retrieve metadata from a blob?**

- Call `BlobClient.GetPropertiesAsync()` and access the Metadata property.
- Example (C# Azure SDK):

```
BlobProperties properties = await blobClient.GetPropertiesAsync();
foreach (var item in properties.Metadata)
{
   Console.WriteLine($"{item.Key}: {item.Value}");
}
```

**6. What are best practices when using metadata in Azure Blob Storage?**

- Keep metadata size small (max 8 KB total per blob).
- Use lowercase keys; metadata keys are case-insensitive.
- Metadata is stored separately; retrieving it requires an extra API call (costs apply).

---

**7. What happens if you overwrite a blob — are properties and metadata preserved?**

- **No**, uploading a blob without explicitly setting metadata and properties will **reset** them to defaults.
- Always reapply desired metadata and properties during overwrite if needed.

---

**8. How do you set or update metadata without overwriting the blob content?**

- Use `BlobClient.SetMetadataAsync()` — it updates metadata without affecting the blob's content.
- No need to re-upload the blob when updating only metadata.

---

**9. How do you use Azure SDK (C#, Python) to manage properties and metadata?**

- **C# Example:**

```
await blobClient.SetMetadataAsync(new Dictionary<string, string> { { "env", "prod" } });
BlobProperties props = await blobClient.GetPropertiesAsync();
Console.WriteLine(props.ContentType);
```

---

**10. How can you search or filter blobs by metadata?**

- Use **Azure Blob Index Tags**, **not** regular metadata.
- With tags, you can query blobs via FindBlobsByTags.
- Example (Azure CLI):

```
az storage blob query-tags --container-name mycontainer --where "tagName = 'value'"
```