

Question #37

Topic 3

## HOTSPOT -

You are developing a web application that makes calls to the Microsoft Graph API. You register the application in the Azure portal and upload a valid X509 certificate.

You create an appsettings.json file containing the certificate name, client identifier for the application, and the tenant identifier of the Azure Active Directory (Azure

AD). You create a method named ReadCertificate to return the X509 certificate by name.

You need to implement code that acquires a token by using the certificate.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

```
AuthenticationConfig config = AuthenticationConfig.ReadFromJsonFile("appsettings.json");
X509Certificate2 certificate = ReadCertificate(config.CertificateName);
var app = .Create(config.ClientId)

    .WithCertificate(certificate)
    .WithAuthority(new Uri(config.Authority))
    .Build();
string[] scopes = new string[] { $"{config.ApiUrl}.default" };
AuthenticationResult result = await app.AcquireTokenForClient().ExecuteAsync();
```

ConfidentialClientApplicationBuilder  
GetAccountAsync()  
GetAccountsAsync()  
ConfidentialClientApplication

scopes  
app  
config

**Correct Answer:****Answer Area**

```
AuthenticationConfig config = AuthenticationConfig.ReadFromJsonFile("appsettings.json");
X509Certificate2 certificate = ReadCertificate(config.CertificateName);
var app = .Create(config.ClientId)

    .WithCertificate(certificate)
    .WithAuthority(new Uri(config.Authority))
    .Build();
string[] scopes = new string[] { $"{config.ApiUrl}.default" };
AuthenticationResult result = await app.AcquireTokenForClient().ExecuteAsync();
```

ConfidentialClientApplicationBuilder  
GetAccountAsync()  
GetAccountsAsync()  
ConfidentialClientApplication

scopes  
app  
config

Box 1: ConfidentialClientApplicationBuilder

Here's the code to instantiate the confidential client application with a client secret: app =

ConfidentialClientApplicationBuilder.Create(config.ClientId)

.WithClientSecret(config.ClientSecret)

.WithAuthority(new Uri(config.Authority))

.Build();

Box 2: scopes -

After you've constructed a confidential client application, you can acquire a token for the app by calling `AcquireTokenForClient`, passing the scope, and optionally forcing a refresh of the token.

Sample code: `result = await app.AcquireTokenForClient(scopes)`

`.ExecuteAsync();`

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/scenario-daemon-app-configuration> <https://docs.microsoft.com/en-us/azure/active-directory/develop/scenario-daemon-acquire-token>

## Topic 4 - Question Set 4

Question #1

Topic 4

You are developing an ASP.NET Core Web API web service. The web service uses Azure Application Insights for all telemetry and dependency tracking. The web service reads and writes data to a database other than Microsoft SQL Server.

You need to ensure that dependency tracking works for calls to the third-party database.

Which two dependency telemetry properties should you use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. `Telemetry.Context.Cloud.RoleInstance`

B. `Telemetry.Id`

C. `Telemetry.Name`

D. `Telemetry.Context.Operation.Id`

E. `Telemetry.Context.Session.Id`

**Correct Answer:** *BD* 

Example:

```
public async Task Enqueue(string payload)
```

```
{
```

```
// StartOperation is a helper method that initializes the telemetry item
```

```
// and allows correlation of this operation with its parent and children. var operation = telemetryClient.StartOperation<DependencyTelemetry>("enqueue " + queueName);
```

```
operation.Telemetry.Type = "Azure Service Bus";
```

```
operation.Telemetry.Data = "Enqueue " + queueName;
```

```
var message = new BrokeredMessage(payload);
```

```
// Service Bus queue allows the property bag to pass along with the message.
```

```
// We will use them to pass our correlation identifiers (and other context)
```

```
// to the consumer.
```

```
message.Properties.Add("ParentId", operation.Telemetry.Id);
```

```
message.Properties.Add("RootId", operation.Telemetry.Context.Operation.Id);
```

Reference:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking>

HOTSPOT -

You are using Azure Front Door Service.

You are expecting inbound files to be compressed by using Brotli compression. You discover that inbound XML files are not compressed. The files are 9 megabytes (MB) in size.

You need to determine the root cause for the issue.

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

| Statement                                       | Yes                   | No                    |
|---|-----------------------|-----------------------|
| The file MIME type is supported by the service. | <input type="radio"/> | <input type="radio"/> |
| Edge nodes must be purged of all cache assets.  | <input type="radio"/> | <input type="radio"/> |
| The compression type is supported.              | <input type="radio"/> | <input type="radio"/> |

Answer Area

|                 | Statement                                       | Yes                              | No                               |
|-----------------|---|----------------------------------|----------------------------------|
| Correct Answer: | The file MIME type is supported by the service. | <input type="radio"/>            | <input checked="" type="radio"/> |
|                 | Edge nodes must be purged of all cache assets.  | <input checked="" type="radio"/> | <input type="radio"/>            |
|                 | The compression type is supported.              | <input checked="" type="radio"/> | <input type="radio"/>            |

Box 1: No -

Front Door can dynamically compress content on the edge, resulting in a smaller and faster response to your clients. All files are eligible for compression.

However, a file must be of a MIME type that is eligible for compression list.

Box 2: No -

Sometimes you may wish to purge cached content from all edge nodes and force them all to retrieve new updated assets. This might be due to updates to your web application, or to quickly update assets that contain incorrect information.

Box 3: Yes -

These profiles support the following compression encodings: Gzip (GNU zip), Brotli

Reference:

<https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching>

**HOTSPOT -**

You are developing an Azure App Service hosted ASP.NET Core web app to deliver video-on-demand streaming media. You enable an Azure Content Delivery

Network (CDN) Standard for the web endpoint. Customer videos are downloaded from the web app by using the following example URL:

`http://www.contoso.com/ content.mp4?quality=1`

All media content must expire from the cache after one hour. Customer videos with varying quality must be delivered to the closest regional point of presence

(POP) node.

You need to configure Azure CDN caching rules.

Which options should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

| Setting                       | Action  |
|-------------------------------|---|
| Caching behavior              | <div><div></div><div>Bypass cache</div><div>Override</div><div>Set if missing</div></div>   |
| Cache expiration duration     | <div><div></div><div>1 second</div><div>1 minute</div><div>1 hour</div><div>1 day</div></div>                                     |
| Query string caching behavior | <div><div></div><div>Ignore query strings</div><div>Bypass caching for query strings</div><div>Cache every unique URL</div></div> |

Answer Area

| Setting         | Action  |
|-----------------|---|
| Correct Answer: | Caching behavior  |
|                 | <div><div></div><div>Bypass cache</div><div>Override</div><div>Set if missing</div></div>   |
|                 | Cache expiration duration   |
|                 | <div><div></div><div>1 second</div><div>1 minute</div><div>1 hour</div><div>1 day</div></div>                                     |
|                 | Query string caching behavior   |
|                 | <div><div></div><div>Ignore query strings</div><div>Bypass caching for query strings</div><div>Cache every unique URL</div></div> |

Box 1: Override -  
Override: Ignore origin-provided cache duration; use the provided cache duration instead. This will not override cache-control: no-cache.  
Set if missing: Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration.  
Incorrect:  
Bypass cache: Do not cache and ignore origin-provided cache-directive headers.

Box 2: 1 hour -  
All media content must expire from the cache after one hour.

Box 3: Cache every unique URL -  
Cache every unique URL: In this mode, each request with a unique URL, including the query string, is treated as a unique asset with its own cache. For example, the response from the origin server for a request for example.ashx?q=test1 is cached at the POP node and returned for subsequent caches with the same query string. A request for example.ashx?q=test2 is cached as a separate asset with its own time-to-live setting.  
Incorrect Answers:  
Bypass caching for query strings: In this mode, requests with query strings are not cached at the CDN POP node. The POP node retrieves the asset directly from the origin server and passes it to the requestor with each request.  
Ignore query strings: Default mode. In this mode, the CDN point-of-presence (POP) node passes the query strings from the requestor to the origin server on the first request and caches the asset. All subsequent requests for the asset that are served from the POP ignore the query strings until the cached asset expires.  
Reference:  
<https://docs.microsoft.com/en-us/azure/cdn/cdn-query-string>



**DRAG DROP -**

You develop a web app that uses tier D1 app service plan by using the Web Apps feature of Microsoft Azure App Service.

Spikes in traffic have caused increases in page load times.

You need to ensure that the web app automatically scales when CPU load is about 85 percent and minimize costs.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Select and Place:

**Actions****Answer Area**

Configure the web app to the Premium App Service tier.

Configure the web app to the Standard App Service tier.

Enable autoscaling on the web app.

Add a Scale rule.

Switch to an Azure App Services consumption plan.

Configure a Scale condition.

**Actions****Answer Area**

Configure the web app to the Premium App Service tier.

Configure the web app to the Standard App Service tier.

Enable autoscaling on the web app.

Add a Scale rule.

Switch to an Azure App Services consumption plan.

Configure a Scale condition.

Configure the web app to the Standard App Service tier.

Enable autoscaling on the web app.

Add a Scale rule.

Configure a Scale condition.

Correct Answer:



Step 1: Configure the web app to the Standard App Service Tier

The Standard tier supports auto-scaling, and we should minimize the cost.

Step 2: Enable autoscaling on the web app

First enable autoscale -

Step 3: Add a scale rule -

Step 4: Add a Scale condition -

Reference:

<https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-autoscale-get-started>

[← Previous Questions](#)

[Next Questions →](#)