Q&A

Up-to-date Questions and Answers from authentic resources to improve knowledge and pass the exam at very first attempt.
----- Guaranteed.

MS-600 Dumps
MS-600 Braindumps
MS-600 Real Questions
MS-600 Practice Test
MS-600 Actual Questions

PASS

Microsoft

MS-600

Building Applications and Solutions with Microsoft 365 Core Services

https://killexams.com/pass4sure/exam-detail/MS-600

Question: 45

HOTSPOT

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
You can use the Adaptive Cards Designer to create Microsoft Outlook actionable messages.	◯	◯
Microsoft Outlook actionable messages are written in JSON.	◯	◯
You need Azure Active Directory (Azure AD) tenant admin privileges to enable Microsoft Outlook actionable messages.	◯	◯

Answer:
Answer Area

Statements	Yes	No
You can use the Adaptive Cards Designer to create Microsoft Outlook actionable messages.	◉	◯
Microsoft Outlook actionable messages are written in JSON.	◉	◯
You need Azure Active Directory (Azure AD) tenant admin privileges to enable Microsoft Outlook actionable messages.	◯	◉

Explanation:

Graphical user interface, text, application

Description automatically generated

Box 1: Yes

The Adaptive Cards Designer provides a drag-and-drop experience to quickly build and tweak adaptive cards.

Outlook Actionable Messages cards are designed using the Adaptive Card format. The Adaptive Card format is a simple yet powerful declarative layout format that provides a lot of flexibility, allowing for visually rich cards. In this topic we'll cover the Outlook-specific features of the Adaptive Card format.

Box 2: Yes

The actionable message card is in JSON format.

Box 3: No

By default, the Tenant Administrator can create, edit, clone, and delete tenants, and manage user accounts.

Note:

To enable Actionable Messages the recipient of the task must be an Office 365 customer with permissions for the SharePoint online site.

No: Office 365 administrators can disable actionable messages via the Set-Organization Config cmdlet. If actionable messages do not render, check with your administrator to make sure the feature is enabled in your organization.

Adaptive Cards Designer Microsoft outlook actionable messages

References:

https://docs.microsoft.com/en-us/outlook/actionable-messages/adaptive-card

https://gingkoapp.com/create-tenant-administrator.html

Question: 46

When developing a tab in Microsoft teams, personal, group, and shared tabs can all be implemented as configurable tabs.

Instructions: Review the underlined BOLD text, if it makes the statement correct, select '' No change is needed if the statement is incorrect selct the answer choice that makes the statement correct.
A. No change is needed.
B. Share the same code
C. Be used in a conversation
D. Have a configuration panel

Answer: A

Question: 47

HOTSPOT

You have an app that queries Azure Active Directory (Azure AD) by using the Microsoft Graph API.

You need to minimize the number of times users are prompted for their credentials.

How should you complete the code? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area

```
...
publicClientApplication. [                    ▼] (accessTokenRequest).then(function(accessTokenResponse) {
                         | acquireTokenSilent            |
                         | acquireTokenPopup             |
                         | acquireTokenByAuthorizationCode|
    callApi(accessTokenResponse.accessToken);
}).catch(function (error) {
    if (error instanceof InteractionRequiredAuthError) {
        publicClientApplication. [                    ▼] (access TokenRequest).then(function(accessTokenResponse) {
                                 | acquireTokenSilent            |
                                 | acquireTokenPopup             |
                                 | acquireTokenByAuthorizationCode|
            callApi(accessTokenResponse.accessToken);
        }).catch(function(error) {
            console.log(error);
        });
    }
});
...
```

Answer:

Answer Area

```
...
publicClientApplication. [                    ▼] (accessTokenRequest).then(function(accessTokenResponse) {
                         | acquireTokenSilent            |
                         | acquireTokenPopup             |
                         | acquireTokenByAuthorizationCode|
    callApi(accessTokenResponse.accessToken);
}).catch(function (error) {
    if (error instanceof InteractionRequiredAuthError) {
        publicClientApplication. [                    ▼] (access TokenRequest).then(function(accessTokenResponse) {
                                 | acquireTokenSilent            |
                                 | acquireTokenPopup             |
                                 | acquireTokenByAuthorizationCode|
            callApi(accessTokenResponse.accessToken);
        }).catch(function(error) {
            console.log(error);
        });
    }
});
```

Explanation:

Graphical user interface, text, application, email

Description automatically generated

Question: 48

You have a single-page application (SPA) named TodoListSPA and a server-based web app named TodoListService.

The permissions for the TodoList SPA API are configured as shown in the TodoList SPA exhibit. (Click the TodoListSPA tab.)

API/PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT REQUIRED
▼ TodoListService-OBO-sample-v2 (1)			
user_impersonation	Delegated	Access TodoListService-OBO-sample-v2	

The permissions for the TodoListService API are configured as shown in the TodoListService exhibit. (Click the TodoListService tab.)

API/PERMISSIONS NAME	TYPE	DESCRIPTION
▼ Microsoft Graph (1)		
UserRead	Delegated	Sign in and read user profile

You need to ensure that TodoListService can access a Microsoft OneDrive file of the signed-in user. The solution must use the principle of least privilege.

Which permission should to grant?
A. the Sites.Read.All delegated permission for TodoListService
B. the Sites.Read.All delegated permission for TodoListSpa
C. the Sites.Read.All application permission for TodoListSPA
D. the Sites.Read.All application permission for TodoListService

Answer: A

Explanation:

A client application gains access to a resource server by declaring permission requests.

Two types are available:

"Delegated" permissions, which specify scope-based access using delegated authorization from the signed-in resource owner, are presented to the resource at run-time as "scp" claims in the client's access token.

"Application" permissions, which specify role-based access using the client application's credentials/identity, are presented to the resource at run-time as "roles" claims in the client's access token.

Reference: https://docs.microsoft.com/en-us/azure/active-directory/develop/developer-glossary#permissions

Question: 49

You are building a custom API.

Client applications will use access tokens to authenticate to the API.

You need to validate the integrity of the tokens.

Which three elements should you verity? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

A. the let claim
B. the aud claim
C. the tld claim
D. the JSON Web Token (JWT) signature
E. the exp laim

Answer: B,D,E

Question: 50

Topic 1, ADatum Corporation

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end on this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Overview

ADatum Corporation develops a software as a service (SaaS) application named E-invoicing.

Existing Environment

Application Architecture

E-invoicing consists of a single-page application (SPA) and a backend web service that provides invoice management and processing functionality.

E-invoicing stores all the details of each invoicing operation in a backend cloud database. E-invoicing generates invoices in PDF format and provides users with the ability to download the PDF after it is generated. Each invoice has a unique identifier named invoiceid.

The users have a common workflow where they sign in to E-invoicing, and then open E-invoicing in multiple tabs of a web browser so they can use different parts of the application simultaneously.

Security Architecture

ADatum uses the principle of least privilege whenever possible. ADatum always uses the latest libraries and

integration endpoints.

Requirements

Business Goals

ADatum wants to integrate E-invoicing, Azure Active Directory (Azure AD), and Microsoft Graph so that their customers can leverage Microsoft Office 365 services directly from within E-invoicing.

Planned Changes

ADatum plans to add the following capabilities to E-invoicing:

– Email the generated invoices to customers on behalf of the current signed-in user. Any emails generated by the system will contain the invoiced.

– Perform as many operations as possible in the browser without having to leave the E-invoicing application.

– Use Azure AD to manage identities, authentication, and authorization.

– Display all emails that contain a specific invoiceid.

Technical Requirements

ADatum identifies the following technical requirements for the planned E-invoicing capabilities:

– Ensure that all operations performed by E-invoicing against Office 365 are initiated by a user. Require that the user authorize E-invoicing to access the Office 365 data the first time the application attempts to access Office 365 data on the user's behalf.

– Send scheduled reminders to customers before a payment due date. Create an administration user interface to enable the scheduled reminders.

– Implement Microsoft Graph change notifications to detect emails from vendors that arrive in a designated mailbox.

– Implement single sign-on (SSO) and minimize login prompts across browser tabs.

– Secure access to the backend web service by using Azure AD.

– Ensure that all solutions use secure coding practices.

Backend Security Planned Changes

ADatum wants to use custom application roles to map user functionality to permissions granted to users.

E-invoicing will have internal logic that will dynamically identify whether the user should be allowed to call the backend API.

SSO JavaScript Script

You plan to implement SSO with Microsoft Authentication Library (MSAL) by using the following code:

```
01 const config ={
02    auth: {
03        clientId: '3b41e6eb-29a1-44cc-8473-e8abfe5c4e07'
04    },
05    cache: {
06      ...
07    }
08 }
09 const myMSALObj = new UserAgentApplication(config);
10 ...
```

Access Token JavaScript Script

You have the following JavaScript code to obtain an access token.

```
01 ...
02 userAgentApplication.acquireTokenSilent(accessTokenRequest).then(
03  function(accessTokenResponse) {
04    let accessToken = accessTokenResponse.accessToken;
05  }).catch(function (error) {
06    if (error.errorMessage.indexOf('interaction_required') !== -1) {
07      userAgentApplication.acquireTokenPopup(accessTokenRequest).then(
08          function(accessTokenResponse) {
09            ...
10          }).catch(function(error) {
11            console.log(error);
12          });
13    }
14    console.log(error);
15 });
```

Change Notification JSON

You have the following JSON message that will be sent by the Microsoft Graph service to detect the vendor emails.

```
01 {
02    "value": [
03       {
04          "subscriptionId":"<subscription_guid>"
05          "subscriptionExpirationDateTime":"<expirationdate>",
06          "clientState":"<secret>",
07          "changeType":"created",
08          "resource":
09             "users/<user_guid>@<tenant_guid>/messages/<long_id_string>",
10          "resourceData":
11       {
12        "@odata.type":"#Microsoft.Graph.Message",
13        "@odata.id":"Users/<user_guid>@<tenant_guid>/Messages/<long_id_string>",
14        "@odata.etag":"W/\"CQAAABYAAADkrWGo7bouTKlsgTZMr9KwAAAUWRHf\"",
15        "id":<long_id_string>"
16       }
17      }
18   ]
19 }
```

DRAG DROP

You need to implement the role functionality for the backend web service calls.

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

Actions	Answer Area
Upload a certificate for the application registration of the backend web service.	1
Modify the manifest that defines the application roles and set Allowed Member Types to **Apps**.	2
Modify the manifest that defines the application roles and set Allowed Member Types to **Users**.	3
Create a new client secret in the application registration of the backend web service.	
Verify for the role claims in the backend web service.	
Assign the roles to the Azure AD group that contains the users who are mapped to the roles.	

Answer:

Actions	Answer Area
Upload a certificate for the application registration of the backend web service.	**1** Create a new client secret in the application registration of the backend web service.
Modify the manifest that defines the application roles and set Allowed Member Types to **Apps**.	**2** Verify for the role claims in the backend web service.
Modify the manifest that defines the application roles and set Allowed Member Types to **Users**.	**3** Assign the roles to the Azure AD group that contains the users who are mapped to the roles.
Create a new client secret in the application registration of the backend web service.	
Verify for the role claims in the backend web service.	
Assign the roles to the Azure AD group that contains the users who are mapped to the roles.	

Question: 51

How can you validate that the JSON notification message is sent from the Microsoft Graph service?
A. The ClientState must match the value provided when subscribing.
B. The user_guid must map to a user ID in the Azure AD tenant of the customer.
C. The tenant ID must match the tenant ID of the customer's Office 365 tenant.
D. The subscription ID must match the Azure subscription used by ADatum.

Answer: A

Explanation:

clientState specifies the value of the clientState property sent by the service in each notification. The maximum length is 128 characters. The client can check that the notification came from the service by comparing the value of the clientState property sent with the subscription with the value of the clientState property received with each notification.

Note: A subscription allows a client app to receive notifications about changes to data in Microsoft Graph.

Reference: https://docs.microsoft.com/en-us/graph/api/resources/subscription

Question: 52

You are building email notifications for an expensing system.

When a user receives an email notification, the email will contain a comment field. When the user submits a comment, the data will be returned to the expensing system for processing.

What should you do to implement the notification by using the minimum amount of development effort?
A. Create a Microsoft Office Add-in that has an action pane to display the notifications
B. Leverage Microsoft Graph notifications

C. Leverage the Azure SignalR Service and implement web notifications
D. Configure the expensing system to send actionable messages

Answer: D

Explanation:

Whether you are filling out a survey, approving an expense report, or updating a CRM sales opportunity, Actionable Messages enable you to take quick actions right from within Outlook. Developers can now embed actions in their emails or notifications, elevating user engagement with their services and increasing organizational productivity.

Office 365 provides two solutions to enhance productivity with Outlook Actionable Messages: actionable messages via email, and actionable messages via Office 365 Connectors.

Reference: https://docs.microsoft.com/en-us/outlook/actionable-messages/

Question: 53

What should you add to a SharePoint Framework (SPFx) solution to ensure that the solution can be used as a Microsoft Teams tab?
A. the TeamsTab value to the componentType property in the manifest file
B. a manifest file to the Teams folder in the solution
C. the TeamsTab value to the supportedHosts property in the manifest file
D. a manifest file to the webparts folder in the solution

Answer: D

Explanation:

You need to update the web part manifest to make it available for Microsoft Teams. Locate the manifest json file for the web part you want to make available to Teams and modify the supportedHosts properties to include "TeamsTab".

Reference: https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/using-web-part-as-msteams-tab

Question: 54

Which URI should you use to query all the emails that relate to an invoice?

```
https://graph.microsoft.com/v1.0/me/messages?$search="{invoiceid}"

https://graph.microsoft.com/v1.0/me/messages?$subject eq {invoiceid}

https://graph.microsoft.com/v1.0/me/messages?${invoiceid}

https://graph.microsoft.com/v1.0/me/messages?$filter=contains(subject,{invoiceid})
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: A

Question: 55

You are developing a Microsoft Teams application for a support staff.

Within a conversation between the support staff, you want the staff to look up support ticket IDs from a list

without opening the ticketing system.

Which feature should the application include?
A. a bot
B. a connector
C. a messaging extension
D. a tab

Answer: C

Explanation:

Messaging extensions allow users to interact with your web service through buttons and forms in the Microsoft Teams client. They can search, or initiate actions, in an external system from the compose message area, the command box, or directly from a message. You can then send the results of that interaction back to the Microsoft Teams client, typically in the form of a richly formatted card.

Reference: https://docs.microsoft.com/en-us/microsoftteams/platform/messaging-extensions/what-are-messaging-extensions