KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

Cloud Development, 2021 fall.
Homework 4

Report

Microsoft Identity Platform

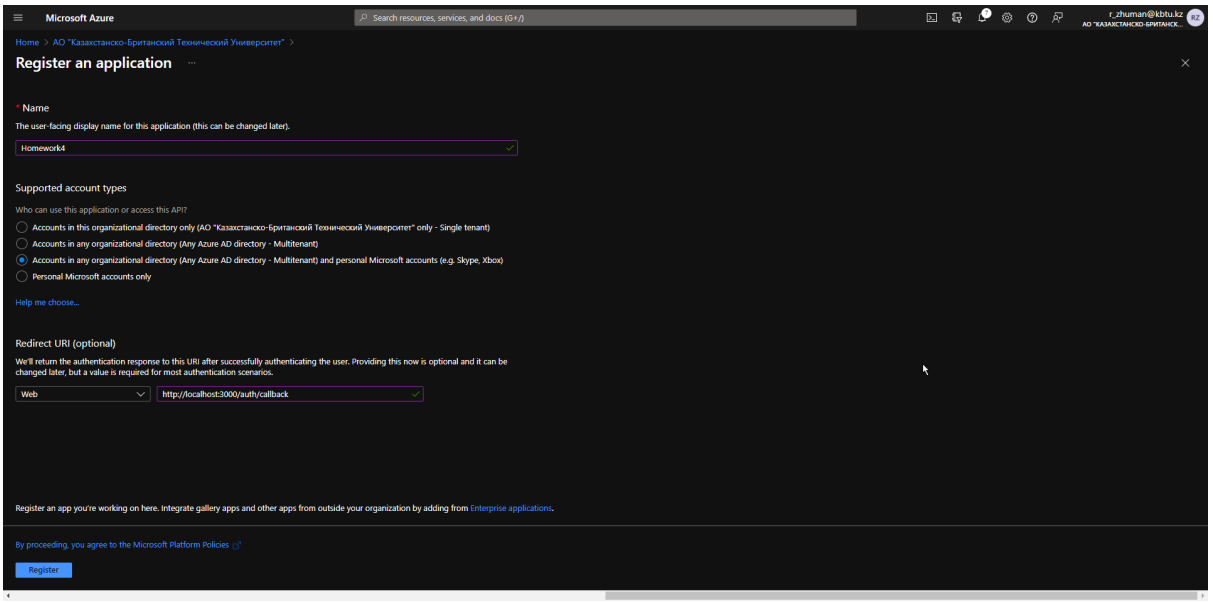
made by Zhuman Rakhat

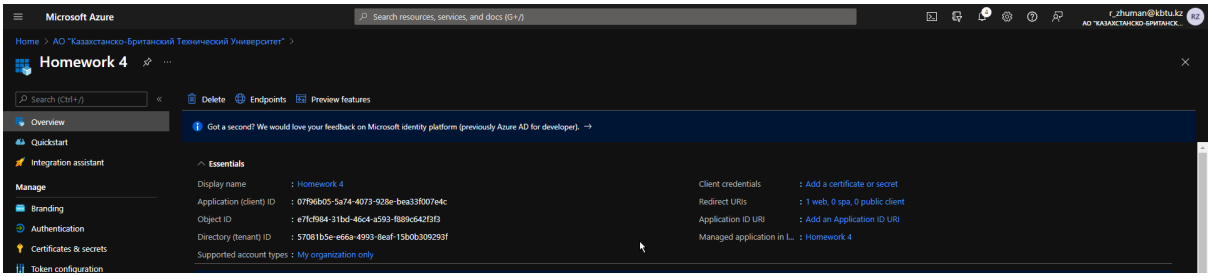
Almaty 2021

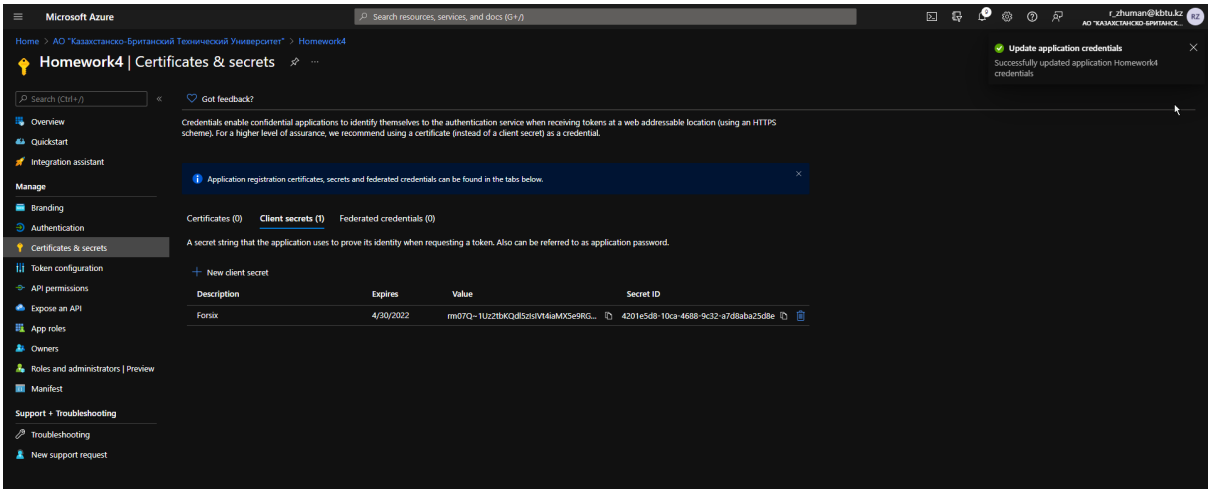
Task

Develop a simple web application, allowing users to be authenticated by Microsoft Identity Platform. Information about current user identity (Full name, for example) has to by reflected on the page headers. Use Microsoft Graph SDK for simplicity. To reduce the amount of work, you can add the mentioned functionality to your existing site from previous homework. Additionally create a blob container, named "securedphotos" (in a Storage Account) with access to blobs allowed only to authenticated users. Your web application have to show images from this container to authorized users only.

Report







>> npx express-generator --hbs graph-tutorial

>> npm install
>> npm install express@4.17.1 http-errors@1.8.0 morgan@1.10.0 debug@4.3.1 hbs@4.1.2
>> npm start
>> npm install dotenv@10.0.0 date-fns@2.23.0 date-fns-tz@1.1.6 connect-flash@0.1.1 express-validator@6.12.1
>> npm install express-session@1.17.2 express-promise-router@4.1.0 isomorphic-fetch@3.0.0
>> npm install @azure/msal-node@1.3.0 @microsoft/microsoft-graph-client@3.0.0 windows-iana@5.0.2

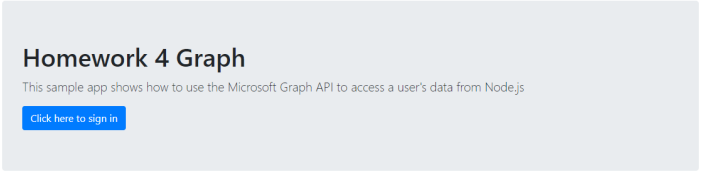
```
const session = require('express-session');
const flash = require('connect-flash');
const msal = require('@azure/msal-node');
```

```
app.use(session({
  secret: 'name',
  resave: false,
  saveUninitialized: false,
  unset: 'destroy'
}));

app.use(flash());

app.use(function(req, res, next) {
  res.locals.error = req.flash('error_msg');
  var errs = req.flash('error');
  for (var i in errs){
    res.locals.error.push({message: 'An error occurred', debug: errs[i]});
  }
  console.log(req.session.user_id)
  if (req.session.userId) {
    res.locals.user = app.locals.users[req.session.userId];
  }

  next();
});
```

Main Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Homework 4</title>

    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
      integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
      crossorigin="anonymous">
    <link rel="stylesheet"
      href="https://use.fontawesome.com/releases/v5.12.1/css/all.css"
      crossorigin="anonymous">
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>

  <body>
    <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
      <div class="container">
        <a href="/" class="navbar-brand">Homework 4</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse"
          aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <ul class="navbar-nav mr-auto">
            <li class="nav-item">
              <a href="/" class="nav-link{{#if active.home}} active{{/if}}">Home</a>
            </li>
            {{#if user}}
              <li class="nav-item" data-turbolinks="false">
                <a href="/images" class="nav-link{{#if active.image}} active{{/if}}">Images</a>
              </li>
            {{/if}}
          </ul>
          <ul class="navbar-nav justify-content-end">
            <li class="nav-item">
              <a class="nav-link" href="https://developer.microsoft.com/graph/docs/concepts/overview" target="_blank">
                <i class="fas fa-external-link-alt mr-1"></i>Docs
              </a>
            </li>
            {{#if user}}
              <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true"
                  aria-expanded="false">
                  {{#if user.avatar}}
                    <img src="{{ user.avatar }}" class="rounded-circle align-self-center mr-2" style="width: 32px;">
                  {{else}}
                    <i class="far fa-user-circle fa-lg rounded-circle align-self-center mr-2" style="width: 32px;"></i>
                  {{/if}}
                </a>
                <div class="dropdown-menu dropdown-menu-right">
```

```
<div class="jumbotron">
  <h1>Homework 4 Graph</h1>
  <p class="lead">This sample app shows how to use the Microsoft Graph API to access a user's data from Node.js</p>
  {{#if user}}
    <h4>Welcome {{ user.displayName }}!</h4>
    <p>Use the navigation bar at the top of the page to get started.</p>
  {{else}}
    <a href="/auth/signin" class="btn btn-primary btn-large">Click here to sign in</a>
  {{/if}}
</div>
```



```
OAUTH_APP_ID=3fb31701-8f13-4817-843d-7a5eb0660a25
OAUTH_APP_SECRET=NfW7Q~VamD_To-kzq43pxtNG9fmILsEhtYQJc
OAUTH_REDIRECT_URI=http://localhost:3000/auth/callback
OAUTH_SCOPES='user.read,mailboxsettings.read'
OAUTH_AUTHORITY=https://login.microsoftonline.com/common/
ACCOUNT=securedphotos
CONTAINER=securedphotos
SECRET=wy2VChkLgRfDUk1cBdqNOucEnFinnzRxzczFW80vM+b35yF4wqi5vhrlDLMKebvsiVpn1MEi8m6gs3nwI20JAw==
```

Msal config

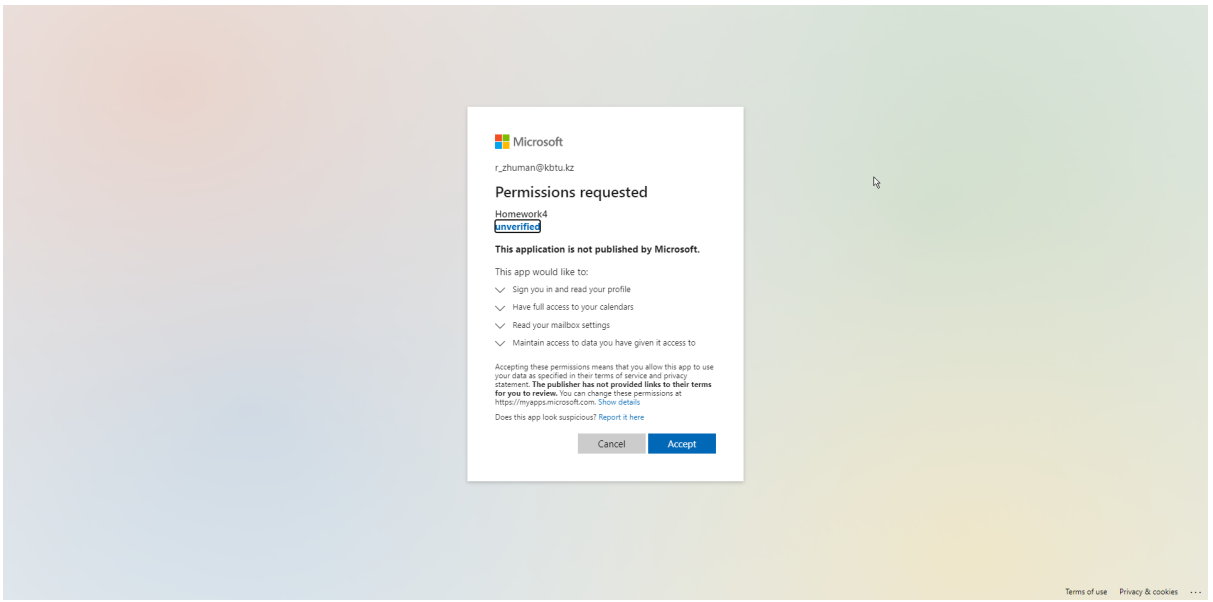
```
app.locals.users = {};
const msalConfig = {
  auth: {
    clientId: process.env.OAUTH_APP_ID,
    authority: process.env.OAUTH_AUTHORITY,
    clientSecret: process.env.OAUTH_APP_SECRET
  },
  system: {
    loggerOptions: {
      loggerCallback(loglevel, message, containsPii) {
        console.log(message);
      },
      piiLoggingEnabled: false,
      logLevel: msal.LogLevel.Verbose,
    }
  }
};

app.locals.msalClient = new msal.ConfidentialClientApplication(msalConfig);
```

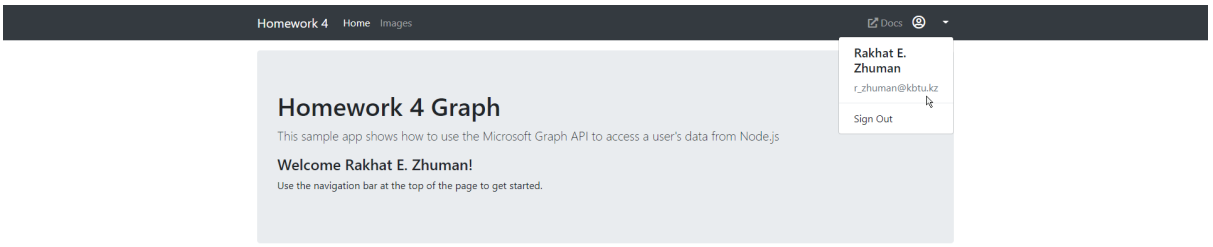
Routes

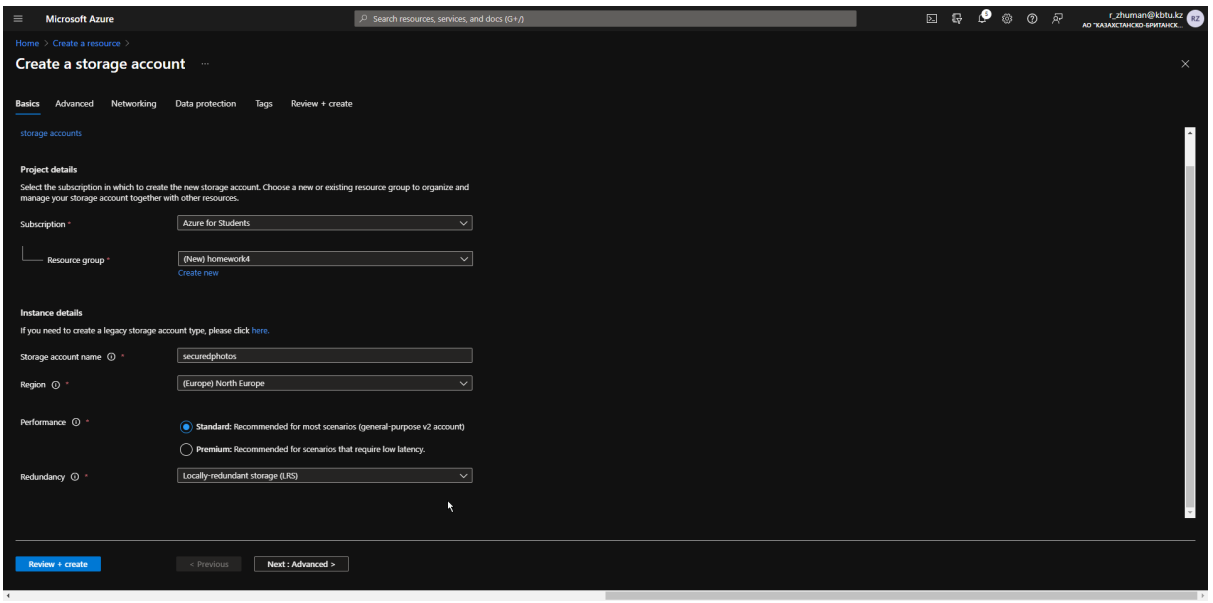
```
const router = require('express-promise-router')();
const graph = require('../graph');

/* GET auth callback. */
router.get('/signin',
  async function (req, res) {
    const urlParameters = {
      scopes: process.env.OAUTH_SCOPES.split(','),
      redirectUri: process.env.OAUTH_REDIRECT_URI
    };

    try {
      const authUrl = await req.app.locals
        .msalClient.getAuthCodeUrl(urlParameters);
      res.redirect(authUrl);
    }
    catch (error) {
      console.log(`Error: ${error}`);
      req.flash('error_msg', {
        message: 'Error getting auth URL',
        debug: JSON.stringify(error, Object.getOwnPropertyNames(error))
      });
      res.redirect('/');
    }
  }
);

router.get('/callback',
  async function(req, res) {
    const tokenRequest = {
      code: req.query.code,
      scopes: process.env.OAUTH_SCOPES.split(','),
      redirectUri: process.env.OAUTH_REDIRECT_URI
    };

    try {
      const response = await req.app.locals
        .msalClient.acquireTokenByCode(tokenRequest);

      // Save the user's homeAccountId in their session
      req.session.userId = response.account.homeAccountId;
      console.log('Token',response.accessToken)
      const user = await graph.getUserDetails(response.accessToken, req.session.userId);
      console.log('User', user)

      // Add the user to user storage
      req.app.locals.users[req.session.userId] = {
        displayName: user.displayName,
        email: user.mail || user.userPrincipalName,
        timeZone: user.mailboxSettings.timeZone
      };
    } catch(error) {
      req.flash('error_msg', {
        message: 'Error completing authentication',
        debug: JSON.stringify(error, Object.getOwnPropertyNames(error))
      });
    }

    res.redirect('/');
  }
);

router.get('/signout',
  async function(req, res) {
    // Sign out
    if (req.session.userId) {
      // Look up the user's account in the cache
      const accounts = await req.app.locals.msalClient
        .getTokenCache()
        .getAllAccounts();

      const userAccount = accounts.find(a => a.homeAccountId === req.session.userId);

      // Remove the account
      if (userAccount) {
        req.app.locals.msalClient
          .getTokenCache()
          .removeAccount(userAccount);
      }
    }

    // Destroy the user's session
    req.session.destroy(function (err) {
      res.redirect('/');
    });
  }
);
```

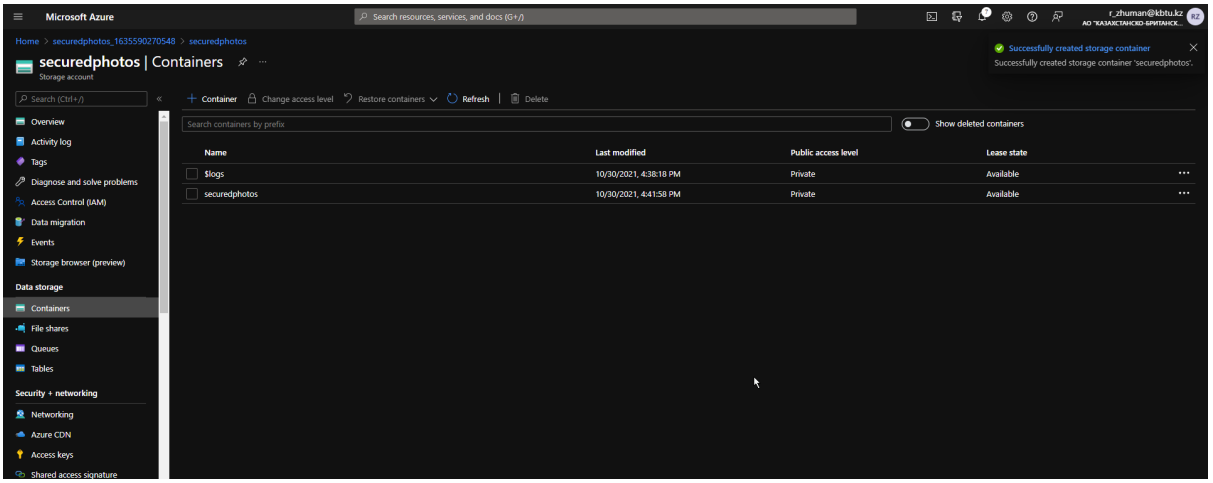
Graph config

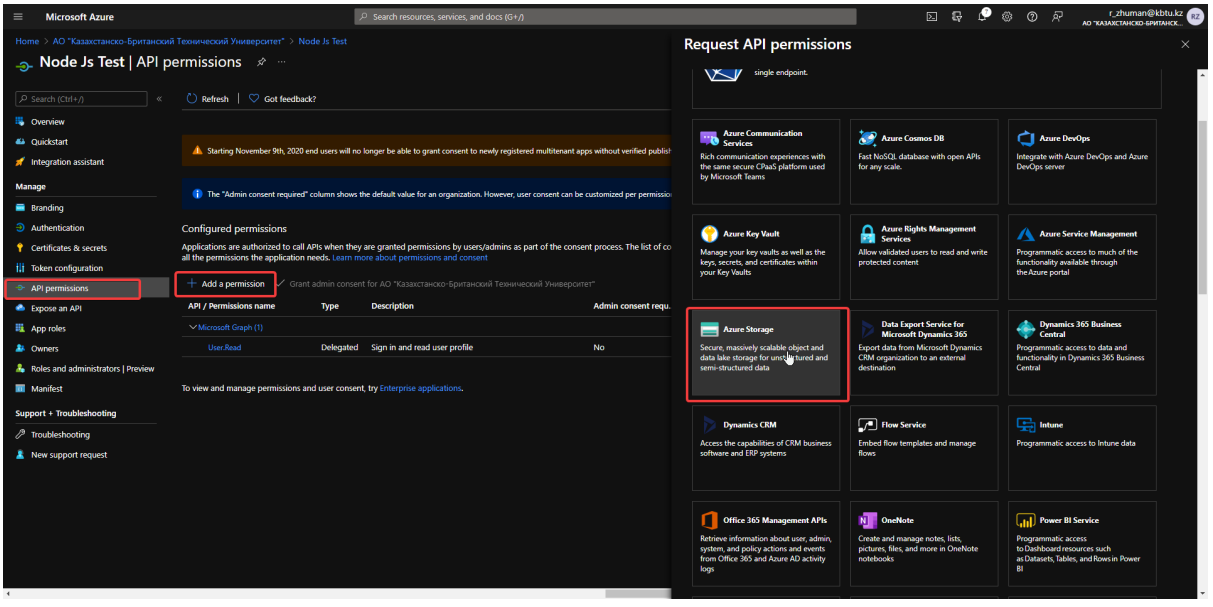
```
var graph = require('@microsoft/microsoft-graph-client');
require('isomorphic-fetch');

module.exports = {
  getUserDetails: async function(msalClient, userId) {
    const client = getAuthenticatedClient(msalClient, userId);

    const user = await client
      .api('/me')
      .select('displayName,mail,mailboxSettings,userPrincipalName')
      .get();
    return user;
  },
};

function getAuthenticatedClient(msalClient, userId) {
  if (!msalClient || !userId) {
    throw new Error(
      `Invalid MSAL state. Client: ${msalClient ? 'present' : 'missing'}, User ID: ${userId ? 'present' : 'missing'}`);
  }

  const client = graph.Client.init({
    authProvider: async (done) => {
      try {
        done(null, msalClient);
      } catch (err) {
        console.log(JSON.stringify(err, Object.getOwnPropertyNames(err)));
        done(err, null);
      }
    }
  });

  return client;
}
```

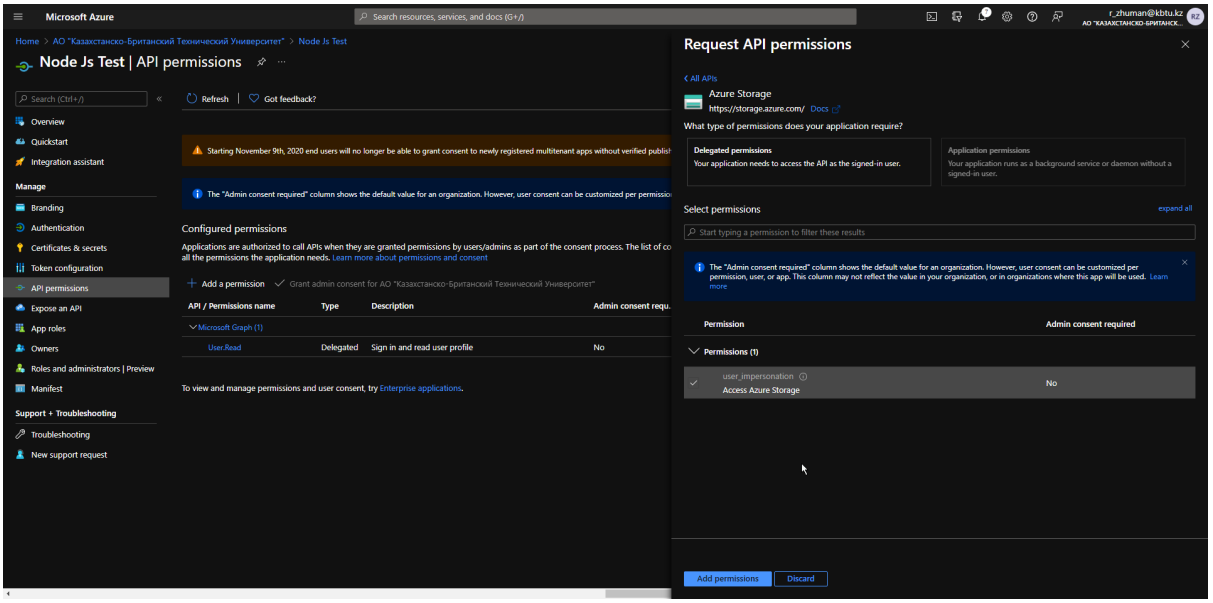





Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
Home > Create a resource >
Create a storage account
Basics
Advanced
Networking
Data protection
Tags
Review + create
storage accounts
Project details
Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.
Subscription *
Azure for Students
Resource group *
(New) homework4
Create new
Instance details
If you need to create a legacy storage account type, please click here.
Storage account name *
securedphotos
Region *
(Europe) North Europe
Performance *
Standard: Recommended for most scenarios (general-purpose v2 account)
Premium: Recommended for scenarios that require low latency.
Redundancy *
Locally-redundant storage (LRS)
Review + create
< Previous
Next : Advanced >


Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
Home > securedphotos_1635590270548 > securedphotos
securedphotos | Containers
Storage account
Successfully created storage container
Successfully created storage container 'securedphotos'.
Search (Ctrl+/)
Container
Change access level
Restore containers
Refresh
Delete
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser (preview)
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Azure CDN
Access keys
Shared access signature
Search containers by prefix
Show deleted containers
Name
Last modified
Public access level
Lease state
$logs
10/30/2021, 4:38:18 PM
Private
Available
securedphotos
10/30/2021, 4:41:58 PM
Private
Available


Microsoft Azure
Search resources, services, and docs (G+/)
r_zhuman@kbtu.kz
Home > АО "Казахстанско-Британский Технический Университет" > Node Js Test
Node Js Test | API permissions
Search (Ctrl+/)
Refresh
Got feedback?
Overview
Quickstart
Integration assistant
Manage
Branding
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators | Preview
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publis
The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permissio
Configured permissions
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of co
all the permissions the application needs. Learn more about permissions and consent
Add a permission
Grant admin consent for АО "Казахстанско-Британский Технический Университет"
API / Permissions name
Type
Description
Admin consent requ.
Microsoft Graph (1)
User.Read
Delegated
Sign in and read user profile
No
To view and manage permissions and user consent, try Enterprise applications.
Request API permissions
single endpoint.
Azure Communication Services
Rich communication experiences with the same secure CPaaS platform used by Microsoft Teams
Azure Cosmos DB
Fast NoSQL database with open APIs for any scale.
Azure DevOps
Integrate with Azure DevOps and Azure DevOps server
Azure Key Vault
Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults
Azure Rights Management Services
Allow validated users to read and write protected content
Azure Service Management
Programmatic access to much of the functionality available through the Azure portal
Azure Storage
Secure, massively scalable object and data lake storage for unstructured and semi-structured data
Data Export Service for Microsoft Dynamics 365
Export data from Microsoft Dynamics CRM organization to an external destination
Dynamics 365 Business Central
Programmatic access to data and functionality in Dynamics 365 Business Central
Dynamics CRM
Access the capabilities of CRM business software and ERP systems
Flow Service
Embed flow templates and manage flows
Intune
Programmatic access to Intune data
Office 365 Management APIs
Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity logs
OneNote
Create and manage notes, lists, pictures, files, and more in OneNote notebooks
Power BI Service
Programmatic access to Dashboard resources such as Datasets, Tables, and Rows in Power BI



Images

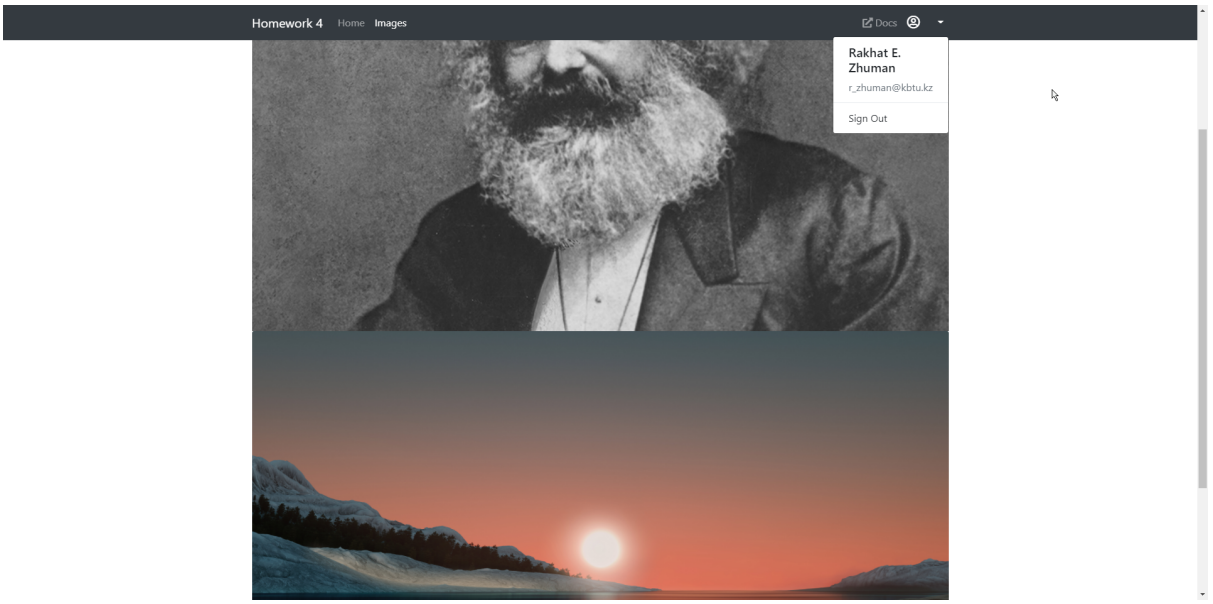
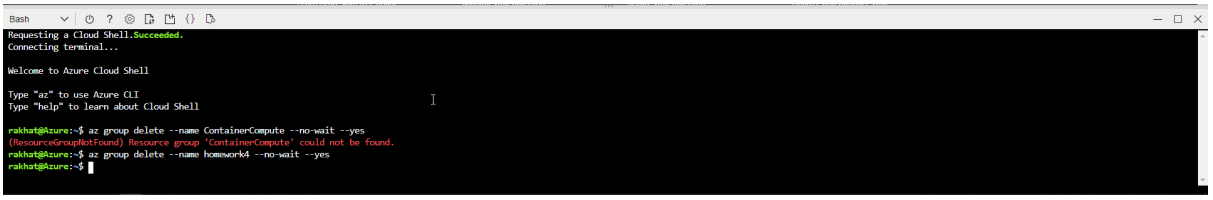
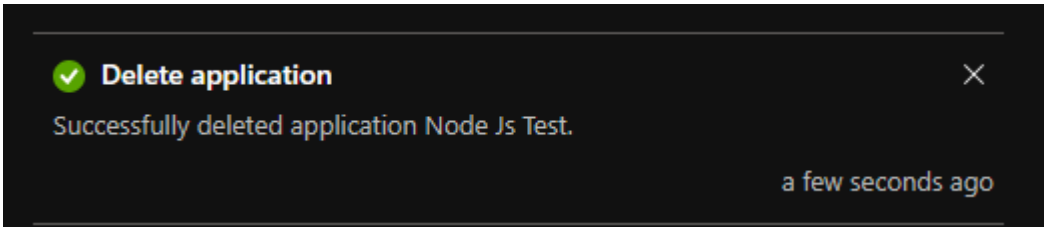
```
const router = require('express-promise-router')();
const graph = require('../graph.js');
const { body, validationResult } = require('express-validator');
const validator = require('validator');
const { retrieve } = require('../blob.js');

router.get('/',
  async function(req, res) {
    if (!req.session.userId) {
      res.redirect('/')
    } else {
      const result = await retrieve()

      const params = {
        active: { image: true },
        data: result
      };
      console.log(params)
      res.render('images', params);
    }
  }
);

module.exports = router;
```

```
const { DefaultAzureCredential, ManagedIdentityCredential } = require("@azure/identity");
const { BlobServiceClient, StorageSharedKeyCredential } = require("@azure/storage-blob");
const { ClientSecretCredential, ChainedTokenCredential } = require("@azure/identity");
const { getBlobSasUri } = require("./functions");

// When an access token is requested, the chain will try each
// credential in order, stopping when one provides a token
// const credential = new ClientSecretCredential(process.env.TENANT, process.env.OAUTH_APP_ID, process.env.OAUTH_APP_SECRET);
const credential = new StorageSharedKeyCredential(process.env.ACCOUNT, process.env.SECRET);
const account = process.env.ACCOUNT;
// const defaultAzureCredential = new ManagedIdentityCredential(userAssignedClientId)

const blobServiceClient = new BlobServiceClient(
  `https://${account}.blob.core.windows.net`, credential
);

const containerName = process.env.CONTAINER;
async function retrieve() {
  const containerClient = blobServiceClient.getContainerClient(containerName);
  console.log(containerClient)
//   const sasuri = await getBlobSasUri(containerClient, account, '538466.marx2x.jpg', credential, null)
    let i = 1;
    let arrayOfBlobs = []
    let blobs = containerClient.listBlobsFlat();
    for await (const blob of blobs) {
        const sasuri = await getBlobSasUri(containerClient, blob.name, credential, null)
        arrayOfBlobs.push(sasuri)
        console.log(`Blob ${i++}: ${blob.name} , ${sasuri}`);
    }
    return arrayOfBlobs
}

module.exports = {
    retrieve
}
```

```js
const { generateBlobSASQueryParameters, BlobSASPermissions } = require('@azure/storage-blob');

function getBlobSasUri(containerClient, blobName, sharedKeyCredential, storedPolicyName) {
    const sasOptions = {
        containerName: containerClient.containerName,
        blobName: blobName
    };

    if (storedPolicyName == null) {
        sasOptions.startsOn = new Date();
        sasOptions.expiresOn = new Date(new Date().valueOf() + 12096e5); // 12096e5 => 14 days => 2 weeks
        sasOptions.permissions = BlobSASPermissions.parse("r");
    } else {
        sasOptions.identifier = storedPolicyName;
    }
    const sasToken = generateBlobSASQueryParameters(sasOptions, sharedKeyCredential).toString();
    console.log(`SAS token for blob is: ${sasToken}`);

    return `${containerClient.getBlockBlobClient(blobName).url}?${sasToken}`;
}

module.exports = {
    getBlobSasUri
}
```

```hbs
<div>
    {{#each data}}
    <img src={{this}} class="card-img-top" alt="...">
    {{/each}}
</div>
```

Result



Cleaning up workspace