CHALLENGE #6 – Access files stored in Azure Blob Storage (LEVEL 2)

Rate this article ★★★★★

FreddyDK (https://social.msdn.microsoft.com/profile/FreddyDK)   June 14, 2018

💬 1 (https://blogs.msdn.microsoft.com/hackathonchallenges/2018/06/14/challenge-6-access-files-stored-in-azure-blob-storage-level-2/#comments)

Share 0   0   0

365 Business Central Sandbox Environment

- ◦ Use http://aka.ms/bcsandbox (http://aka.ms/bcsandbox) to create an Azure VM if you do not have a sandbox environment.
- Visual Studio Code with the AL Extension installed
  - ◦ Azure VMs will have VS Code pre-installed
- The Url for an Azure Blob Storage with unauthenticated access
  - ◦ https://challenge6.blob.core.windows.net/files (https://challenge6.blob.core.windows.net/files)

Expected result
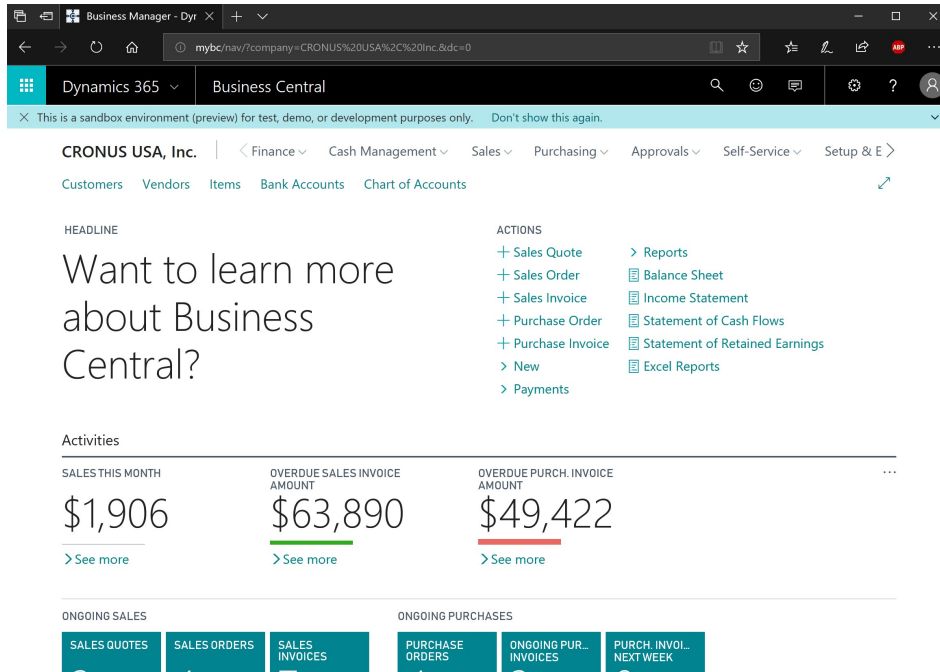
Steps

- Create an empty app
- Create a new table with the relevant fields to describe an Azure Blob
- Create a list page on top the new table. The table should be temporary.
- Create a codeunit to encapsulate the access to Azure Blob Storage and a can populate the new table.
- Add an action to the page that can open the selected document.

Hints

- The documentation for Azure Blob Service REST API can be found here https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api (https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api). Look for the "List Containers" operation.
- In VS Code, use Ctrl+Shift+P and type AL GO and remove the customerlist page extension
- Use the *ttable* and the *tfield* snippets to create the table.
- You can invoke the "List Container" Url directly in a browser to see the returned information.
- Use the *tpage* and the *tpagefield* snippets to create the list page.
- Use the tcodeunit snippet to create the codeunit.
- Use the Hyperlink() method to open the document.

Cheat Sheets

- Create an empty app (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-1-CREATE-AN-EMPTY-APP.pdf)
- Inspect list container response (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-2-INSPECT-LIST-CONTAINER-RESPONSE.pdf)
- Create new table (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-3-CREATE-NEW-TABLE.pdf)
- Code for accessing blob storage (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-4-CODE-FOR-ACCESSING-BLOB-STORAGE.pdf)
- Create list page (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-5-CREATE-LIST-PAGE.pdf)
- Add open action (https://msdnshared.blob.core.windows.net/media/2018/06/CHALLENGE-6-CHEAT-SHEET-6-ADD-OPEN-ACTION.pdf)

Happy coding

Esben Nyhuus Kristoffersen
Principal Achitect

Passwords: Dote6111, Cujy0673, Xymo2826, Sufo3097, Nyko3952, Veci1100

Tags Azure (https://blogs.msdn.microsoft.com/hackathonchallenges/tag/azure/) Blob Storage
(https://blogs.msdn.microsoft.com/hackathonchallenges/tag/blob-storage/) Files
(https://blogs.msdn.microsoft.com/hackathonchallenges/tag/files/) LEVEL 2
(https://blogs.msdn.microsoft.com/hackathonchallenges/tag/level-2/)

Comments (1)

Name *

Email *

Website

☐ Save my name, email, and website in this browser for the next time I comment.

<div align="right">Post Comment</div>

Dave Rendón (https://social.msdn.microsoft.com/profile/Dave+Rend%C3%B3n)
August 6, 2018 at 12:37 pm (https://blogs.msdn.microsoft.com/hackathonchallenges/2018/06/14/challenge-6-
access-files-stored-in-azure-blob-storage-level-2/#comment-175)
Great scenario, thanks!

Reply (https://blogs.msdn.microsoft.com/hackathonchallenges/2018/06/14/challenge-6-access-files-stored-in-
azure-blob-storage-level-2/?replytocom=175#respond)