Deploy WOPI Server

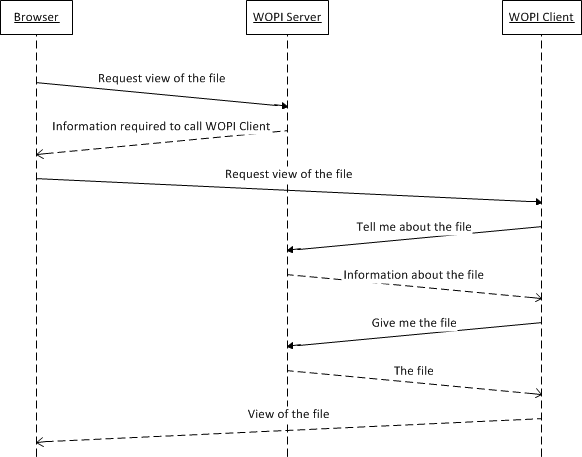
# 1 Goal

Deploy our own WOPI server that enable to view and edit documents by using WOPI application.

1. View/edit file
2. Access token validation
3. Separate file storage server

# 2 Overview

WOPI enables a client to access and change files stored by a server. This allows the client to render files and provide file editing functionality for files stored by the server.



Using WOPI to provide a browser-based viewer for a specific type of file

**WOPI Server**: A document storage location that can be accessed on the web.

1. File storage
2. Web-based services to view and edit documents ([MS-WOPI])

NOTE It’s OK to storage files on a server and provide the web-based services on another server.

**WOPI Client**: It’s the application that can perform actions on files on WOPI server like viewing and editing. In this example, we can use Office Online as the WOPI client.

**Browser**: Wrapping the WOPI application URL.

Example Office Online URL: <http://officeserver/wv/wordviewerframe.aspx?wopisrc=http://wopiserver2/wopi/files/MS-WOPI.docx&access_token=fdjkal>

Reason for not using WOPI application URL:

1. The URL is not friendly looking;
2. The URL is not for WOPI server but the WOPI client, making user confused;
3. The URL exposes the access token, it’s user-specific and time-sensitive.

Because of these reasons, we create a page within WOPI server that contains an iframe points to the WOPI application URL. For example, it likes <http://wopiservername/documents/ms-wopi.docx>

When open from browser, you determine the user requesting the page and issue them an access token. Then direct to the WOPI application URL.

# 3 Environment

## 3.1 WOPI Client

We use Office Online as the WOPI client. Office Online must be installed in domain but not in DC.

Following below steps to install and configure Office Online:

1. Install relative windows features

Add-WindowsFeature Web-Server,Web-Mgmt-Tools,Web-Mgmt-Console,Web-WebServer,Web-Common-Http,Web-Default-Doc,Web-Static-Content,Web-Performance,Web-Stat-Compression,Web-Dyn-Compression,Web-Security,Web-Filtering,Web-Windows-Auth,Web-App-Dev,Web-Net-Ext45,Web-Asp-Net45,Web-ISAPI-Ext,Web-ISAPI-Filter,Web-Includes,InkandHandwritingServices,NET-Framework-Features,NET-Framework-Core,NET-HTTP-Activation,NET-Non-HTTP-Activ,NET-WCF-HTTP-Activation45

1. Install Office Online
2. Install language package for Office Online
3. Restart computer
4. Run PowerShell command: “New-OfficeWebAppsFarm –internal "http://[officeservername]" -allowhttp -editingenabled” to create farm
5. Run PowerShell command: Set-OfficeWebAppsFarm -OpenFromUrlEnabled:$true
6. Run PowerShell command: Set-OfficeWebAppsFarm -ExternalURL “http://[officeserverexternalip]”
7. Verification: can open http://[officeservername]/hosting/discovery and <http://[officeservername]/op/generate.aspx>

## 3.2 WOPI Server

1. Deploy a FTP site as <https://vpsie.com/knowledge-base/how-to-setup-ftp-server-users-on-windows-2012-r2/> (ensure “Application Development” is installed when installing IIS)
   1. Create a local user WOPITestUser for step “2-Create FTP users”
   2. Create folder “C:\FTPStorage” where the FTP directory tree will be saved for step “4-Creating FTP site”
   3. The FTP site can be accessed in address like <ftp://10.172.37.76/>
2. Get source code in <https://github.com/InteropEvents/WOPISolution>
3. Open the source code and update below settings in Web.config

         <!-- Settings related to Office online server start -->

         <add key="WopiDiscovery" value="http://10.172.37.66/hosting/discovery"/>

         <add key="WOPIServerName" value="10.172.37.76:809"/>

     <add key="OfficeServerIP" value="10.172.37.66"/>

     <add key="OfficeServerName" value="Officeserver"/>

     <!-- Settings related to Office online server end -->

     <!-- Settings related to FTP start -->

     <add key="FileServiceUrl" value="ftp://10.172.37.76/"/>

     <add key="FileServiceUserName" value="WOPITestUser"/>

     <add key="FileServiceUserPassword" value="Password01!"/>

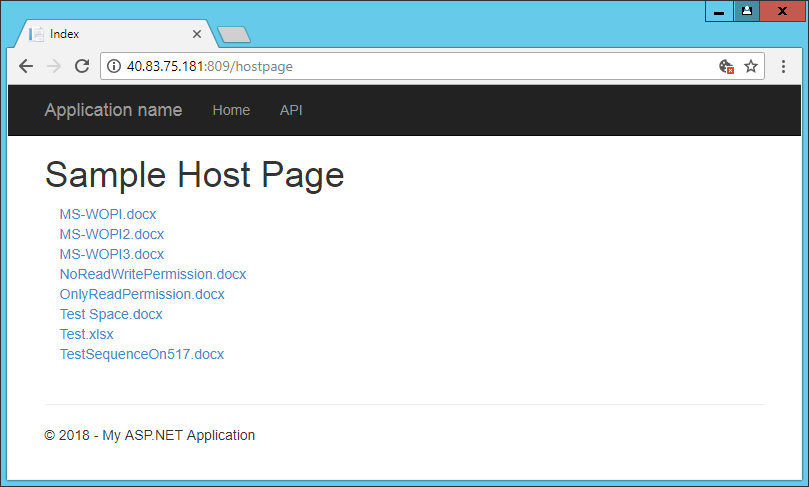
     <!-- Settings related to FTP end -->

1. Set value of “WopiDiscovery” to the Office online server WOPI discovery url
2. Set value of “WOPIServerName” to the WOPI host IP:Port
3. Set value of “OfficeServerIP” to Office online server IP
4. Set value of “OfficeServerName” to Office online server name
5. Set value of “FileServiceUrl” to the address mentioned in step 1
6. Set “FileServiceUserName” to the user name mentioned in step 1
7. Set “FileServiceUserPassword” to the password of user in “FileServiceUserName”
8. Build this solution
9. Create a folder named “WOPIPublish” and publish this solution to the folder.
10. Create a site to point to the publish folder in step 5 and select a port (such as 809) for it.
11. Add inbound rule in firewall to allow port 809 if firewall is open.
12. Access host page in http:// WOPIServerPublicIP:Port/hostpage

# 4 Scenario

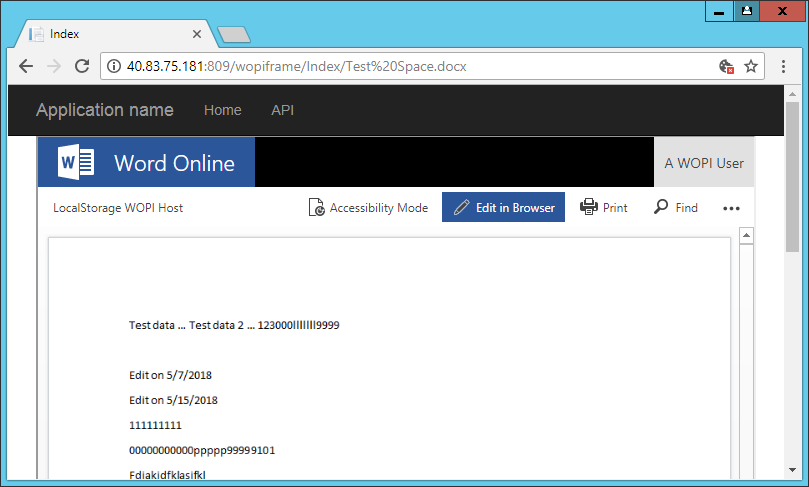
## 4.1 View a document with permission

1. Open the host page in url “http:// [WOPIServerPublicIP:Port]/hostpage”



Host page is coded in [WOPIHost - IISFileServer\WOPIHost\Controllers\HostPageController.cs] and [WOPIHost - IISFileServer\WOPIHost\Views\HostPage\Index.html]

1. Chick on the file name



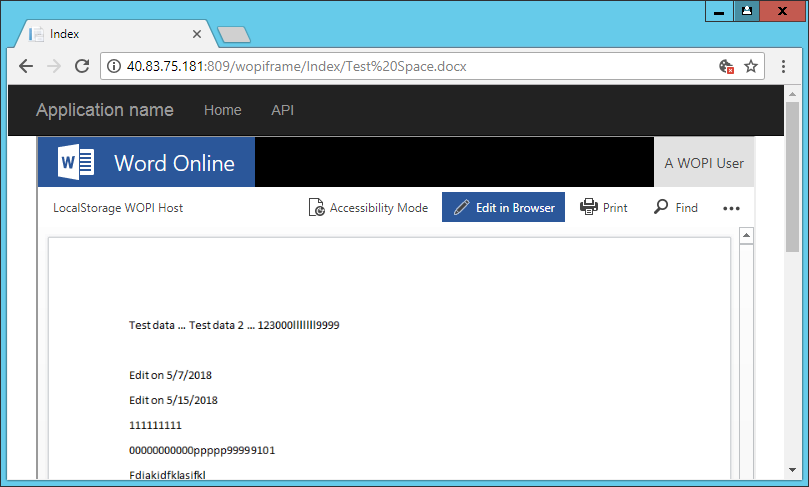
When click on the file name, it will go to WOPI frame page. WOPI frame page is coded in [WOPIHost - IISFileServer\WOPIHost\Controllers\ WopiFrameController.cs] and [WOPIHost - IISFileServer\WOPIHost\Views\WopiFrame\Index.html]

When display the documents, it actives Office Online to send CheckFileInfo and GetFile request to the WOPI server.

CheckFileInfo and GetFile is handled in method “HandleCheckFileInfoRequest” and “HandleGetFileRequest” in [WOPIHost - IISFileServer\WOPIHost\Controllers\WopiHandler.cs]

## 4.2 Edit a document with permission

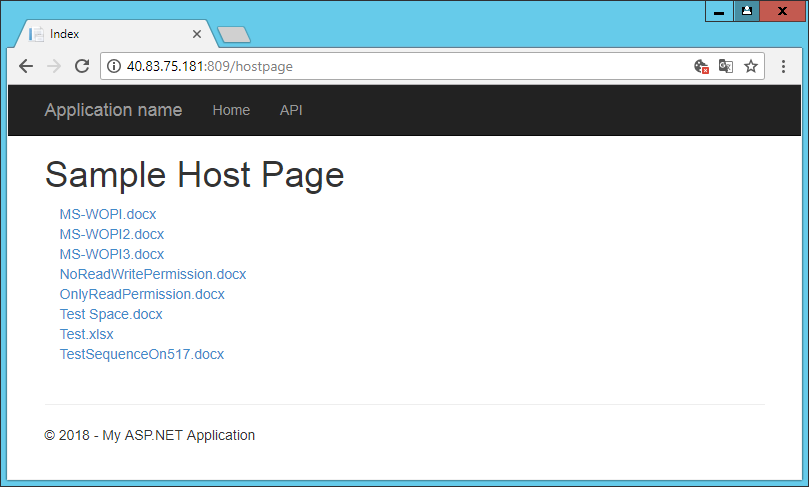
1. Open document with steps in section 4.1
2. Click on “Edit in Browser”



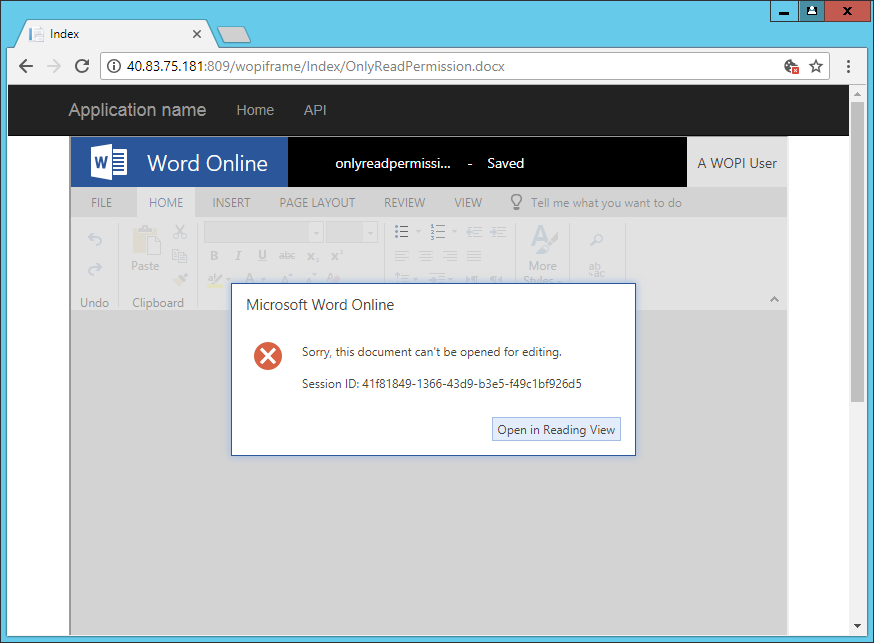
When enter edit mode, Office online will send Lock/GetFile/UnlockAndRelock request, and them are handled in method “HandleLockRequest”/”HandleUnlockAndRelockRequest”/”HandleGetFileRequest” in [WOPIHost - IISFileServer\WOPIHost\Controllers\WopiHandler.cs]. After finish editing, PutFile and Unlock will be invoked, and them are handled in “HandlePutFileRequest”/” HandleUnlockRequest” in WopiHandler.cs

## 4.3 View or edit document without permission

1. Open host page and click on “NoReadWritePermission.docx” or “OnlyReadPermission.docx”



1. Message will alert if there’s no permission



When WOPI server receives a request from WOPI client, it will validate user permission that in an access token. This is implemented in method “ValidateAccess” in WopiHandler.cs. Such as ValidateAccess is invoked in the beginning of HandleCheckFileInfoRequest.



File [Desktop\WOPIHost - IISFileServer\WOPIHost\App\_Data\UserPermissions.xml] stores user permission.

<userpermission>

<user name="TestUser">

<file name="NoReadWritePermission.docx">none</file>

<file name="OnlyReadPermission.docx">Read</file>

</user>

</userpermission>

Permission includes none/read/write. Default, user has read/write permission on all documents. This includes the exceptions on that. In above example, user “TestUser” has read permission for document “OnlyReadPermission.docx” and no any permission for document “NoReadWritePermission.docx”; and “TestUser” has read/write permission for all other documents.

Access token is generated and validated in [WOPIHost - IISFileServer\WOPIHost\Utils\AccessTokenUtil.cs].

# 5 Tracking Problem

1. When install Office Online in Azure VM then deploy WOPI server on our own machine in Microsoft local area network, it fails.

Reason: In this situation, WOPI client cannot find WOPI server that in a local area network. So it cannot request the file information successfully.

Resolution: WOPI client and WOPI server must be able to access each other. Now we put both WOPI client and WOPI server on Azure VM.

1. Cannot view document with space in its name.

Resolution:

1. Encode request url and decode in WOPI server implementation
2. Add below highlight configuration in Web.config

<security>

<requestFiltering allowDoubleEscaping="true" />

</security>

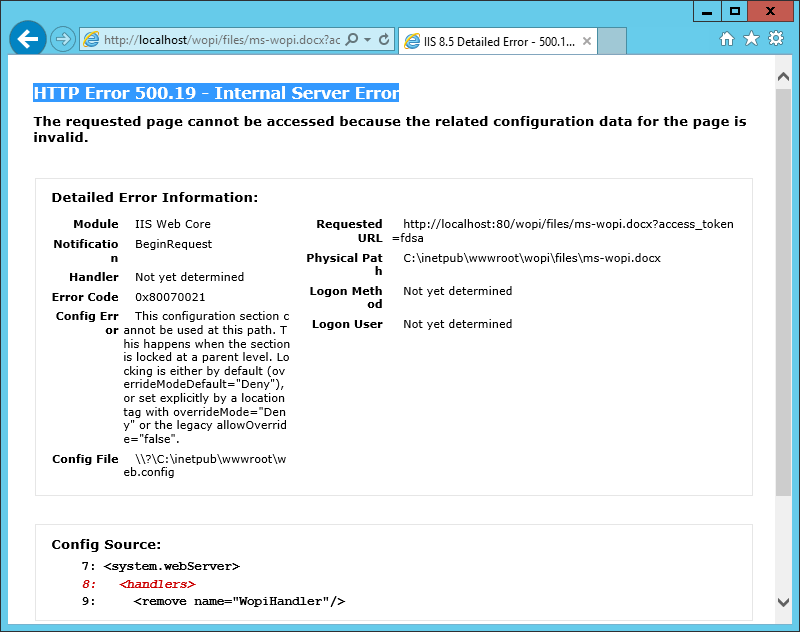
</system.webServer>

1. Cannot view empty Word/PPTX file

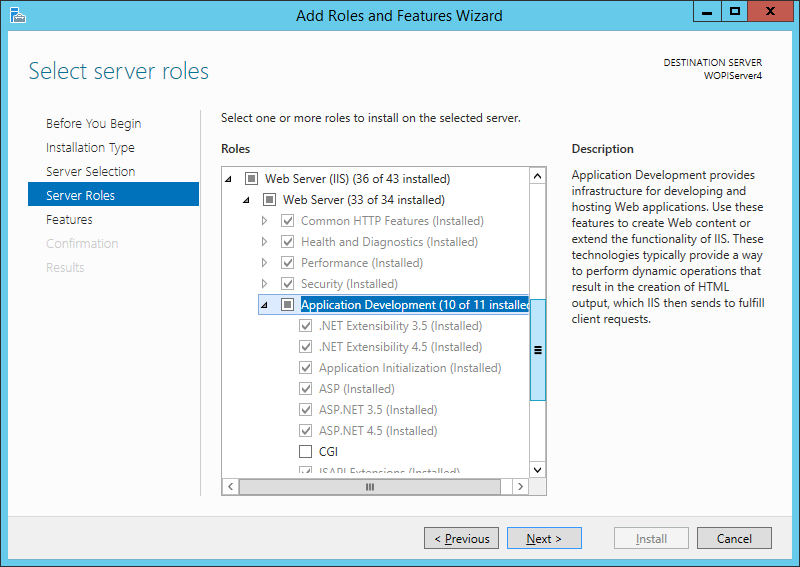
Reason: There’s no stream transferred from WOPI server to WOPI client to be displayed because the size of empty Word/PPTX file is 0 byte.

Resolution: Un-supported

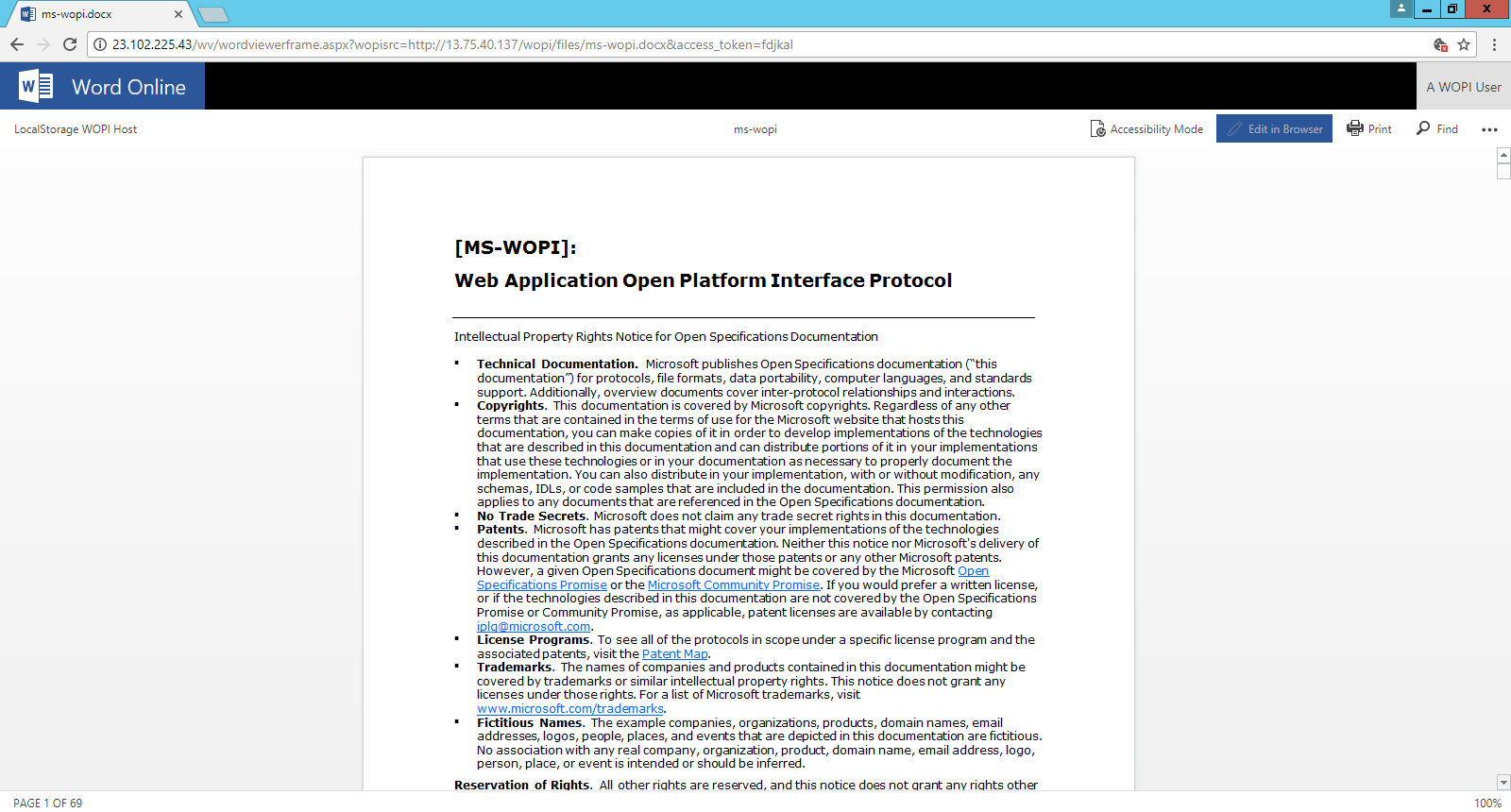
1. After deployed WOPI server, error “HTTP Error 500.19 – Internal Server Error” displays when open URL “http://localhost/wopi/files/ms-wopi.docx?access\_token=fdas”



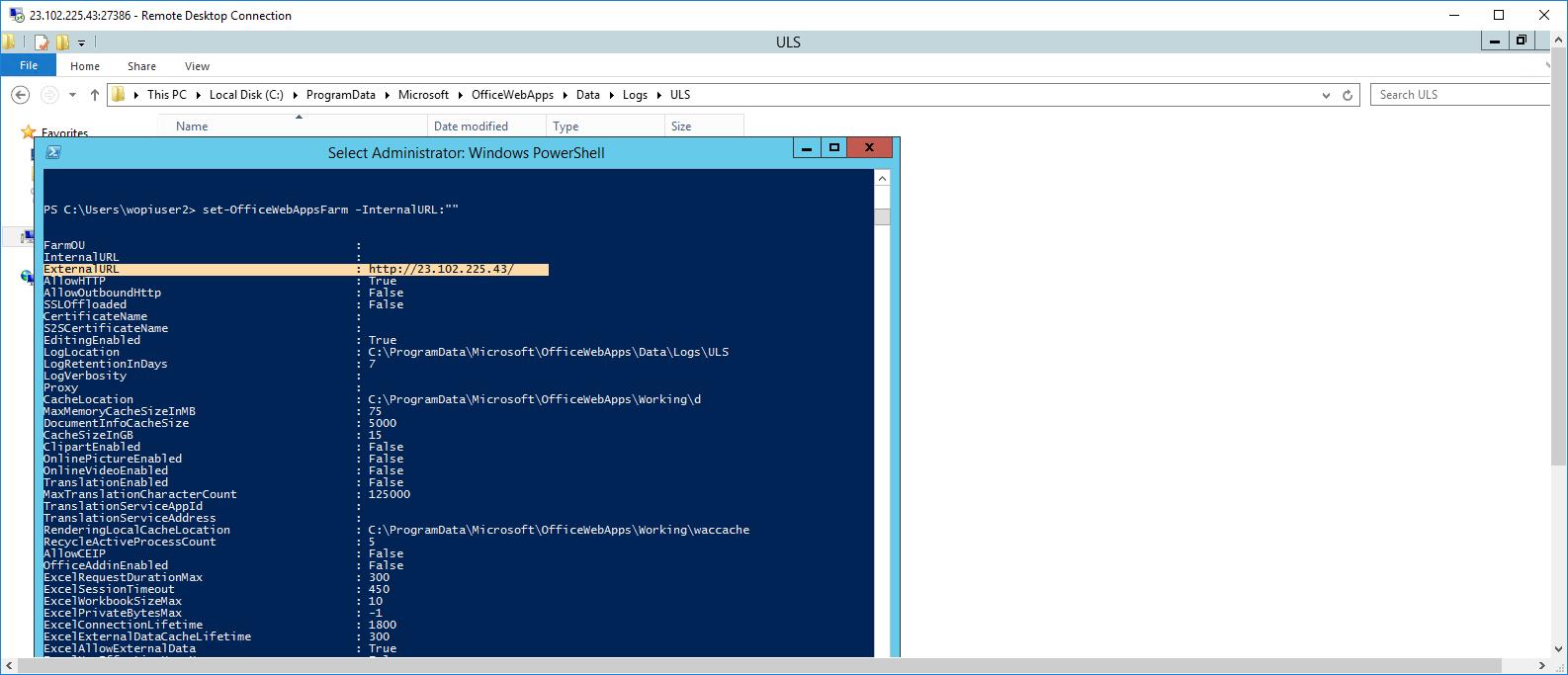
Resolution: Install “Application Development”



1. “Edit in Browser” button is disabled after opened Word document.



Resolution: Only set InternalURL when using machine name in word viewer URL; Only set ExternalURL when using IP in word viewer URL.



1. After edit document successfully, the updates do not display in the view mode.

Resolution:

For .xlsx file, as described in [Integrating with Office Online](http://wopi.readthedocs.io/en/latest/overview.html), file version returned in CheckFileInfo should be changed after file is updated.

For .docx file, (1) same resolution as .xlsx file; (2) set “SHA256” in CheckFileInfo response as specified in [Office Online Integration Documentation](https://media.readthedocs.org/pdf/wopi/latest/wopi.pdf).

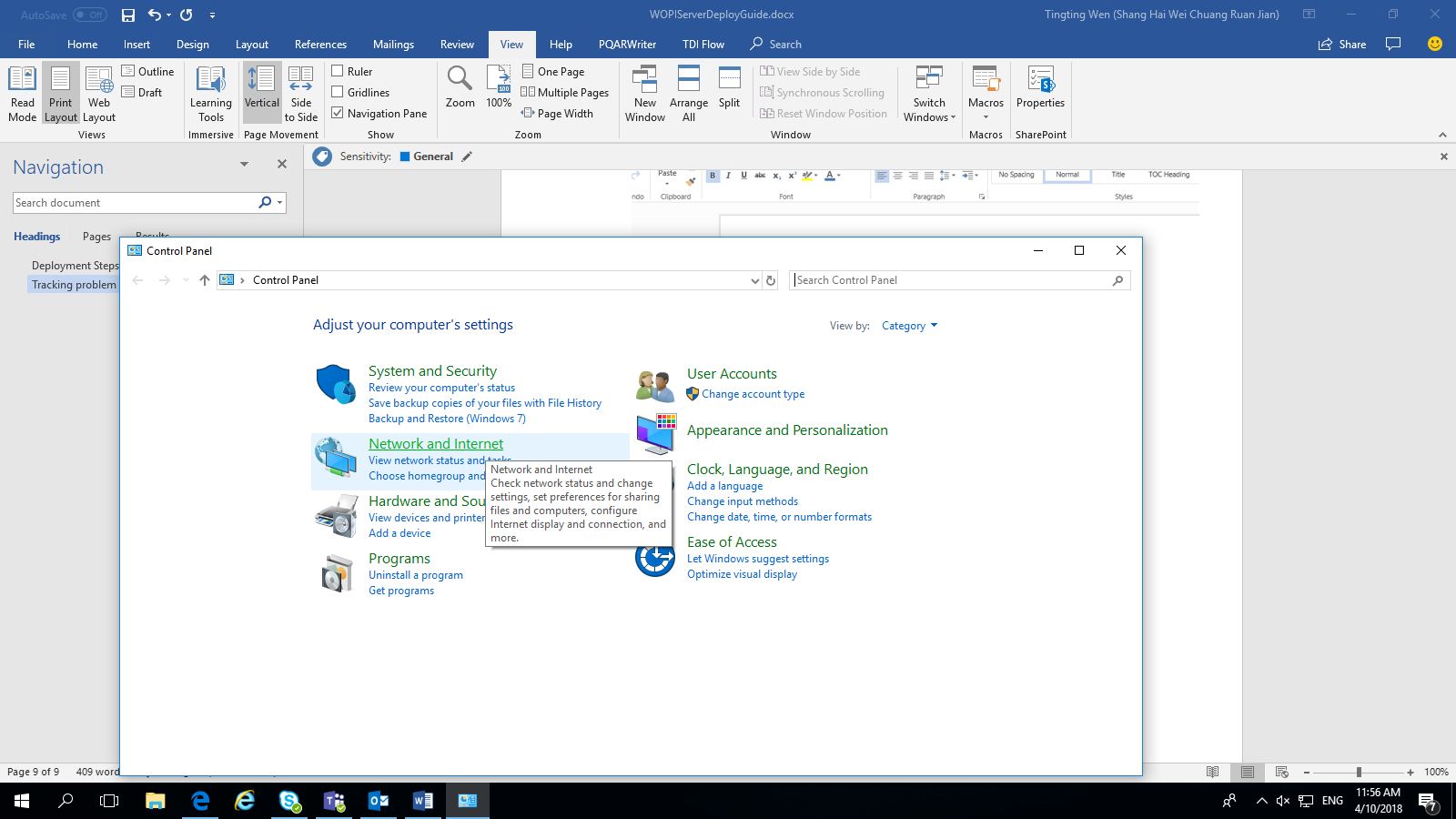
1. File list does not display in the host page as below.



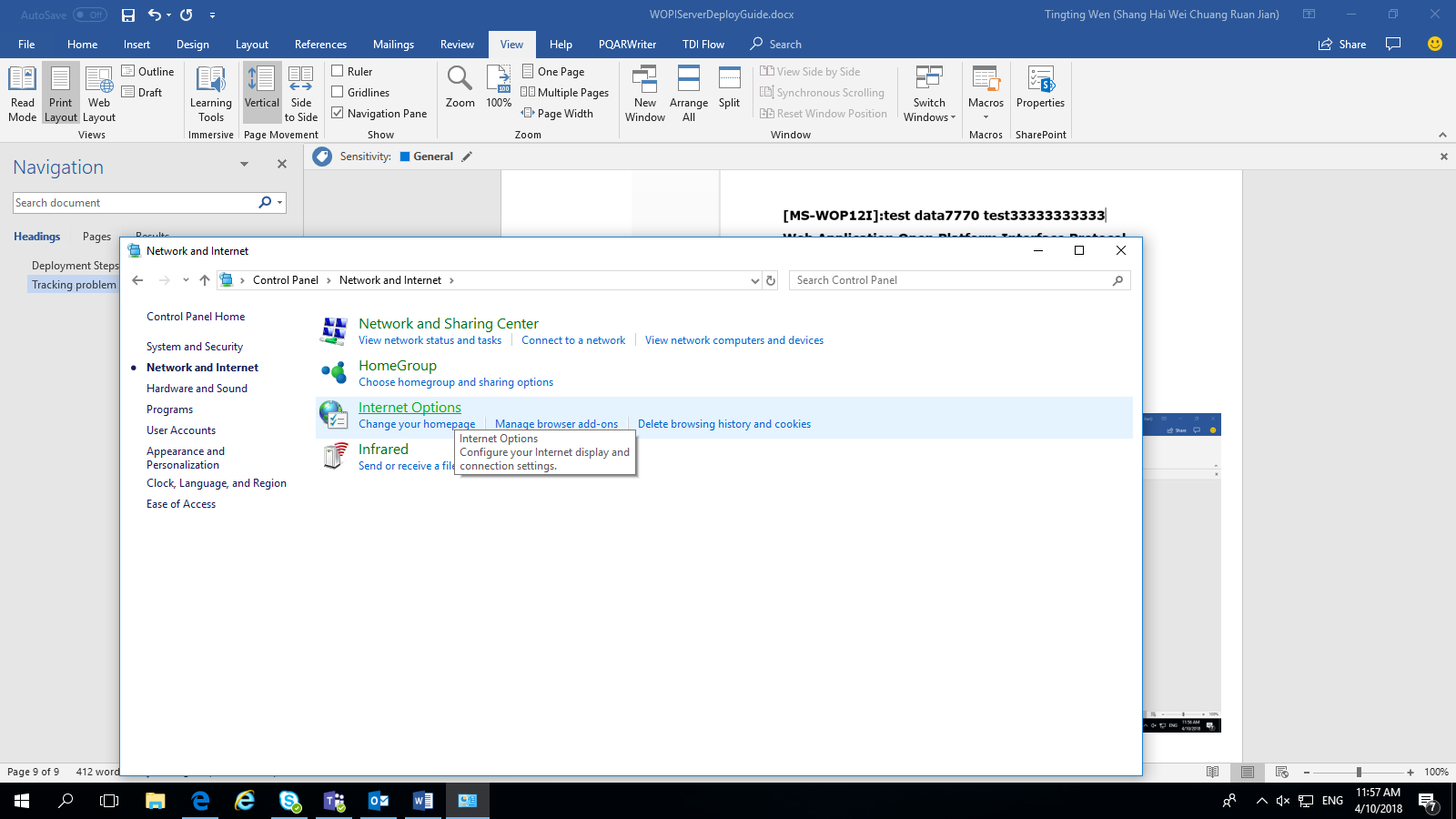
Resolution: Add HTTP header as below.



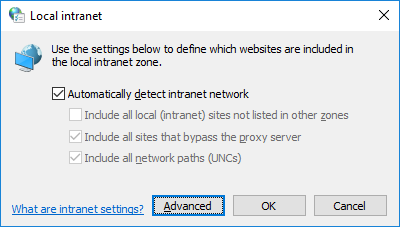
1. Cannot view document with browser “Microsoft Edge” but success with other browser like “Internet Explorer” or “Chrome”.
2. Open Control Panel
3. Click Network and Internet



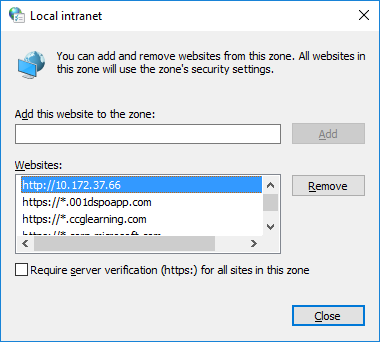
1. Click Internet Options



1. Click Security > Local intranet > Sites, and select as below



1. Click “Advanced” in above steps, and add <http://10.172.37.66> to the zone



1. Restart computer

# 6 Appendix

If you’re not using FTP file storage as specified in this document, you need to take following actions to the source code got in step 2 in section [3.2](#_3.2_WOPI_Server)

1. Add a class into project like xxxxFileStorage and inherit from interface IFileStorage, implement all those functions in the interface.
2. Update Method “CreateFileStorage” in FileStorageFactory.cs to return xxxxFileStorage
3. Rebuild the solution