Setup Guide OfficeCore Headless

1. Grab a copy of the code (there's a **setup.exe.config** in there you should use).
2. Install XM 10.2 with that setup.exe.config (this does install a CD as well as a CM - we don't particularly care about CD, haven't done anything with it yet).
3. Install 'Sitecore Headless Services Server XM 19.0.0 rev 00508' and 'Sitecore ManagementServices 4.0.0 rev. 00411' (you can grab both from dev.sitecore.net).

Error: Installing XM :

[---------- IdentityServer\_StopWebsite : ManageWebsite -----------------------]

[IdentityServer\_StopWebsite]:[Stop] identityserver.headless

The property 'Value' cannot be found on this object. Verify that the property exists.

[TIME] 00:00:07

The property 'Value' cannot be found on this object. Verify that the property exists.

Solution: Delete the website in IIS and try again installing Prereq.

1. (For Daniela, this was not needed – so for now don’t do this! Keep in for later just in case)

In IIS, add the binding for jss.officecore.demo -> use the wildcarded SSL certificate we have (\*.officecore.demo).

Make sure you turn on 'Require Server Name Indication'. (to be clear, this binding needs to be added to your CM website)

1. After that, go to the folder your code lives in using CMD(**in admin mode!!)** and run

npm -i. (npm -install)

1. Then do 'dotnet sitecore login --authority https://[your identity server] --cm https://[your cm server] --allow-write true'

(this logs your CLI in to be able to serialize/deserialize content),

1. Do 'dotnet sitecore ser push' (that pushes the content from disk into Sitecore. If you make changes in Sitecore you can run 'dotnet sitecore ser pull').

Error: Make sure the GraphQL service is installed and available.

Solution : add **https://** in our instance name in user.json (\.sitecore\user.json) and then tried again and it worked for us. Make sure you have https:// in your instance, then try.

* Install JSS CLI
* npm install -g @sitecore-jss/sitecore-jss-cli

1. You might be able to run 'jss deploy config' but **before** check c:\Projects\NextJS\Officecore\ scjssconfig.json that has the right info on instancePath and that deployUrl has https://

Error: permission denied (I think this is just if you don’t open CMD in admin mode)

Solution: you need to give permissions to wwwroot for creations

1. followed by 'jss deploy items -c -d' (you should get some warnings in there which is expected - we don't expect errors there though).

Error: **ERR\_OSSL\_EVP\_UNSUPPORTED**

**Solution: Check the Node.js version and if it’s V17.\* then you need to uninstall it and install V16.13.0 or v16.15.0**

Error: Certificate error (**Most probable you will get this!)**

Solution: Node does not use Windows' cert store, so if it's a trusted cert there it will not be trusted in the JSS CLI. So you would have to use --acceptCertificate [cert thumbprint] parameter . You will receive an error

” Expected server SSL certificate to have thumbprint cm.headless from acceptCertificate, but got 83:01:CC:52:77:6F:06:47:CC:A8:E6:30:88:42:3A:20:B7:AE:98:45 from server.”

**So try this again with this command (**the thumbprint will be different for you, the cmd will give display it in the error message**):**

“jss deploy items --acceptCertificate 83:01:CC:52:77:6F:06:47:CC:A8:E6:30:88:42:3A:20:B7:AE:98:45 “

<https://sitecore.stackexchange.com/questions/16401/unexpected-response-from-import-service-error-unable-to-get-local-issuer-certi>

1. Lastly, you should now be able to run 'jss start:connected' and browse jss.officecore.demo.

Certificate error: do this <https://doc.sitecore.com/xp/en/developers/hd/190/sitecore-headless-development/walkthrough--configuring-sitecore-ca-certificates-for-node-js.html>

API key error fix is <https://doc.sitecore.com/xp/en/developers/hd/190/sitecore-headless-development/create-a-sitecore-api-key.html>

Error Connection to your rendering host failed with a Not Found error. Ensure the POST endpoint at URL [**http://rendering:3000/api/editing/render**](http://rendering:3000/api/editing/render) has been enabled.

Some fix:  add module.exports = {TrailingSlashes: true}; in next.config.js

Error: error - ./src/components/graphql/Header.dynamic.graphql  
Module parse failed: Unexpected token (1:9)  
You may need an appropriate loader to handle this file type, currently no loaders are configured to process this file. See <https://webpack.js.org/concepts#loaders>

Potential Solution:

In brief, the os temp folder is used as the caching folder by default and it could be unstable (Although we never got such a report before from any other client).  
Based on this github conversation, we can try to change the default caching root folder.Please follow the method below:  
1. Go to the "\node\_modules\@sitecore-jss\sitecore-jss-nextjs\dist\cjs\middleware\editing-data-cache.js"2. Modify the code like below:.....  
const tempFolder = '/customFolder'  
class EditingDataDiskCache {  
constructor() {  
// Use gzip compression and store using the OS temp directory (Vercel Serverless Functions have temp directory access)  
this.cache = new sync\_disk\_cache\_1.default('editing-data', { compression: 'gzip', location: tempFolder });  
}  
set(key, editingData) {  
......We changed the os.default.tmpdir() into another custom folder. You can set any file path you like here since it will create a new folder if it doesn't exist. Let's see how it goes.

1. Build the Officecore .sln from the project folderin Visual Studio and copy the .dll in

Inetpub/projectName/bin folder, where your sitecore instance is.

Please contact me if all these doesn’t work!