

How to Run WOPI Validator Tests on O365

The WOPI Validator Tool allows us to make sure our endpoints conform to the WOPI protocol so we can offer an Office 365 integration. This tool is run through Windows PowerShell, but in order to make it work, we need 4 things first:

1. The WOPI source code on our machine
2. A valid Rubex Authentication token
3. The port that we can use to run the WOPI tool with
4. The node id of the file we're running the WOPI too

Please note, number 4 is only needed if you're using a file endpoint from FilesController.cs in Utopia (wopi/files/{id}). There are other endpoints you can use, just be sure to update the powershell command accordingly. This is just an example using the file endpoint.

Source Code Setup (Utopia and WOPI)

1. Clone the [WOPI source code](#) onto your machine in a place easily accessible to you. A good option is just your source folder where your other repositories are.
 2. Once the WOPI code is cloned, open the solution at the path: wopi-validator-core\WopiValidator.sln
 3. Build the project in **Release** mode. This will add the necessary files and directories to enable you to run the tool in PowerShell later.
 4. In Utopia, navigate to Utopia/Controllers/Wopi/WopiBaseController.cs - modify the ValidateRequestAsync method to always return true. We cannot run the proofkey portion of the test locally.
-

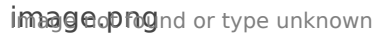
Rubex Authentication Token

1. Run Utopia
2. Go to the UtopiaDB and view the data in the DbUserSessions table.
3. Copy the token under the AccessToken column header on your current session row (i.e. 07680FAC-90D8-419B-AF81-5C14BFA23F4F). This token is the valid Rubex Auth Token

you need to run the WOPI tool. Copy it and paste it somewhere that will be easily accessible later.

1. Alternatively, open up devtools networking tab in the Utopia window. Find an API call and copy the access token from the Authorization header
4. **Important: Keep your local Rubex session alive during the entire rest of this process or you will need to grab a new auth token!**

Port to Use

1. With the Utopia project still running, go to your taskbar and right click IIS Express.
2. Under the header "View Sites", you should see Utopia. Hover over it to see the ports Rubex is using.

3. There should be two ports, one that matches the port in the URL (i.e. localhost:54298) and one that's different (i.e. localhost:44313).
4. The port you want to copy and save for later is the one that **doesn't** match the URL port when you look at the browser running your Utopia project. The port you want is probably 44313 or 44311 or the like.
5. Keep track of it in the same place you saved your authentication token.

Node ID to Use

1. Download one of the wopi test files found [here](#) . It doesn't matter which one you use, there are just a few in here to try out.
2. Upload the file you chose to your localhost environment.
3. Once the upload completes, go to the UtopiaDB again and view the data in the DbNodes table.
4. Find the wopi test file you created and grab its Node ID from the Id column of the table (i.e. 294 or 11020). Keep track of it in the same place you saved your port and authentication token.

Putting it All Together

1. Open PowerShell in the wopi-validator-core\src\WopiValidator\bin\Release\net6.0\ directory
2. The first time you try to run a test with a new user, you must run the following test:

```
dotnet Microsoft.Office.WopiValidator.dll -t TOKEN -l 0 -w https://localhost:PORT/wopi/files/NODE_ID -g PutUserInfo
```

This will populate a string value on the user so that all subsequent tests will pass. If this is missing, you'll get error messages about missing the "UserInfo" property.

Substitute the TOKEN, PORT, and NODE_ID for your retrieved values.

3. If you want to run all of the WOPI Validations tests, run:

```
dotnet Microsoft.Office.WopiValidator.dll -t TOKEN -l 0 -w https://localhost:PORT/wopi/files/NODE_ID --ignore-skipped
```

Substitute the TOKEN, PORT, and NODE_ID for your retrieved values.

4. If you want to run a specific Test Case of the WOPI Validation tests, run:

```
dotnet Microsoft.Office.WopiValidator.dll -n NAME -w https://localhost:PORT/<YOUR ROUTE HERE> -t TOKEN -l 0
```

Substitute the TOKEN and PORT with your retrieved values and the YOUR ROUTE HERE needs to match up to the endpoint you are testing. The NAME is where you place the name of the test case you want to run.

- **For example:** If you are testing name `containers.EnumerateAncestorsOnRoot` then you should match it to the endpoint `/wopi/containers/NODE_ID/ancestry` where node is `NODE_ID` is your retrieved value from above.
- Test case and test group names are all found inside of TestCases.xml found in the same folder as that .dll

5. See the **Additional Info** section for more commands

Troubleshooting

- If at any time, all your test cases are getting skipped and you want to see why, be sure to remove the `--ignore-skipped` flag from your command and run it again. If you see that the reason for skipping the test cases is a **306 Unused** error, that means your port is wrong. Make sure you're using one of the ports listed under Utopia in IIS Express, and make sure it's not the port your browser is using to connect to Rubex.
- If you are getting lots of Authorization exceptions while you're running your test cases, it simply means your Rubex Authentication token is no longer valid. Log out (if you're not already) and login again. Then follow the steps in the Rubex Authentication Token section of this document to get the new auth token. Replace the token in your command with this new one and run the test cases again.
- If you are seeing a message about missing the UserInfo property, see step 2 of "Putting it All Together"

Additional Info

[Wopi Documentation](#)

Microsoft.Office.WopiValidator 1.0.0

Copyright (C) 2018 Microsoft

`-w, --wopisrc` Required. WopiSrc URL for a wopitest file

`-t, --token` Required. WOPI access token

-l, --token_ttl	Required. WOPI access token ttl
-c, --config	(Default: runConfig.xml) Path to XML file with test definitions
-g, --testgroup	Run only the tests in the specified group (cannot be used with testname)
-n, --testname	Run only the test specified (cannot be used with testgroup)
-e, --testcategory	(Default: All) Run only the tests in the specified category
-s, --ignore-skipped	Don't output any info about skipped tests.
--help	Display this help screen.
--version	Display version information.

New work item

Revision #2

Created 18 July 2022 23:23:57 by Bryce Holloway

Updated 19 April 2024 05:48:49 by Royce Williams