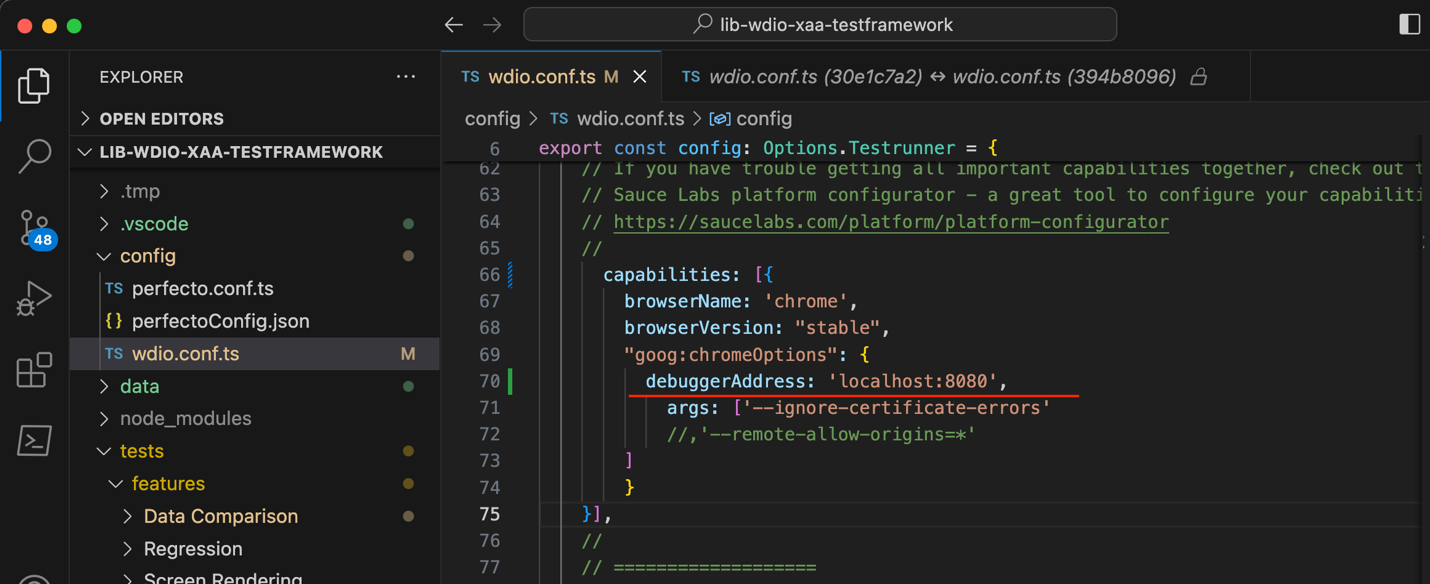
**Running a script in parallel environment**

Parallel environment must be accessed through Okta authentication using myApps. Because of the authentication, we must login in manually to myApps, and do some special setup to connect to the already opened browser.

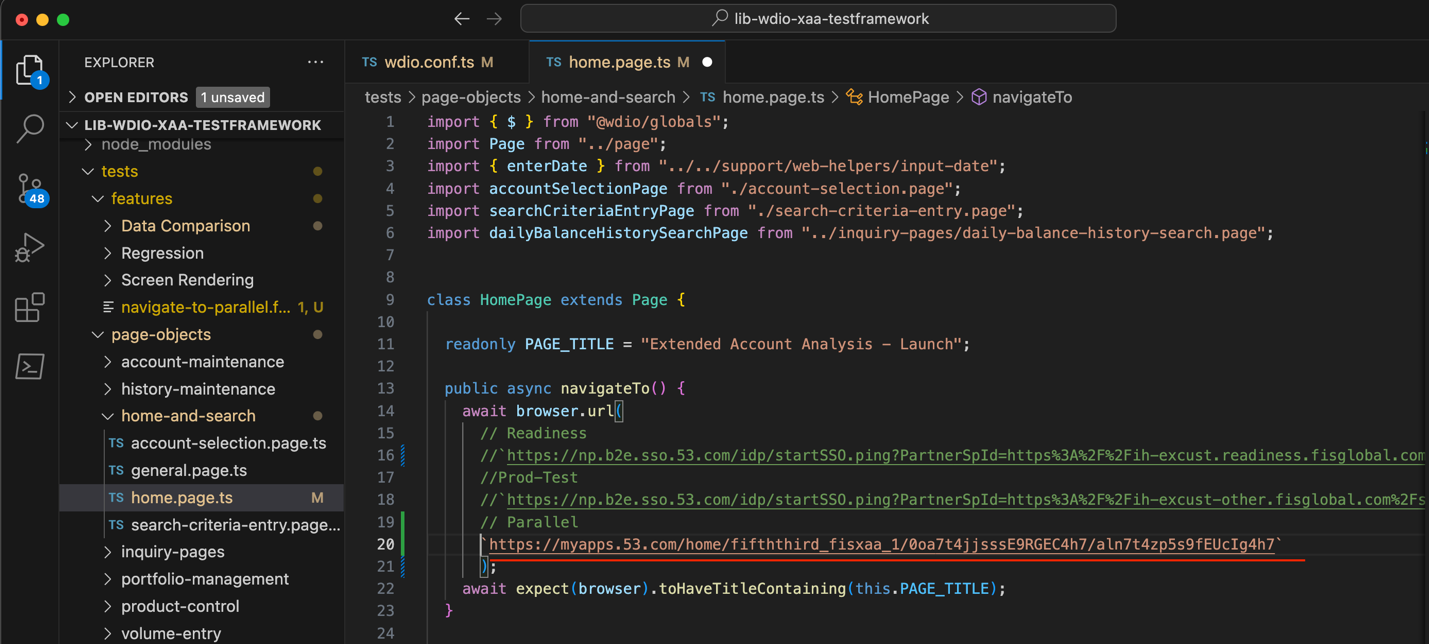
**Pre-requisite (one time setup):**

1. in the framework go to the file wdio.conf.ts. In the below section, add the line:

debuggerAddress: "localhost:8080",

1. Also in the framework go to the file [home.page.ts](http://home.page.ts). In the below section, comment out the current url and add the Parallel url

`https://myapps.53.com/home/fifththird\_fisxaa\_1/0oa7t4jjsssE9RGEC4h7/aln7t4zp5s9fEUcIg4h7`

****

**Steps which must be done each time to run in Parallel:**

1. Quit chrome browser if it’s open
2. Start chrome browser from command line with debugging enabled:
   1. On Windows:

"C:\Program Files\Google\Chrome\Application\chrome.exe" -remote-debugging-port=8080

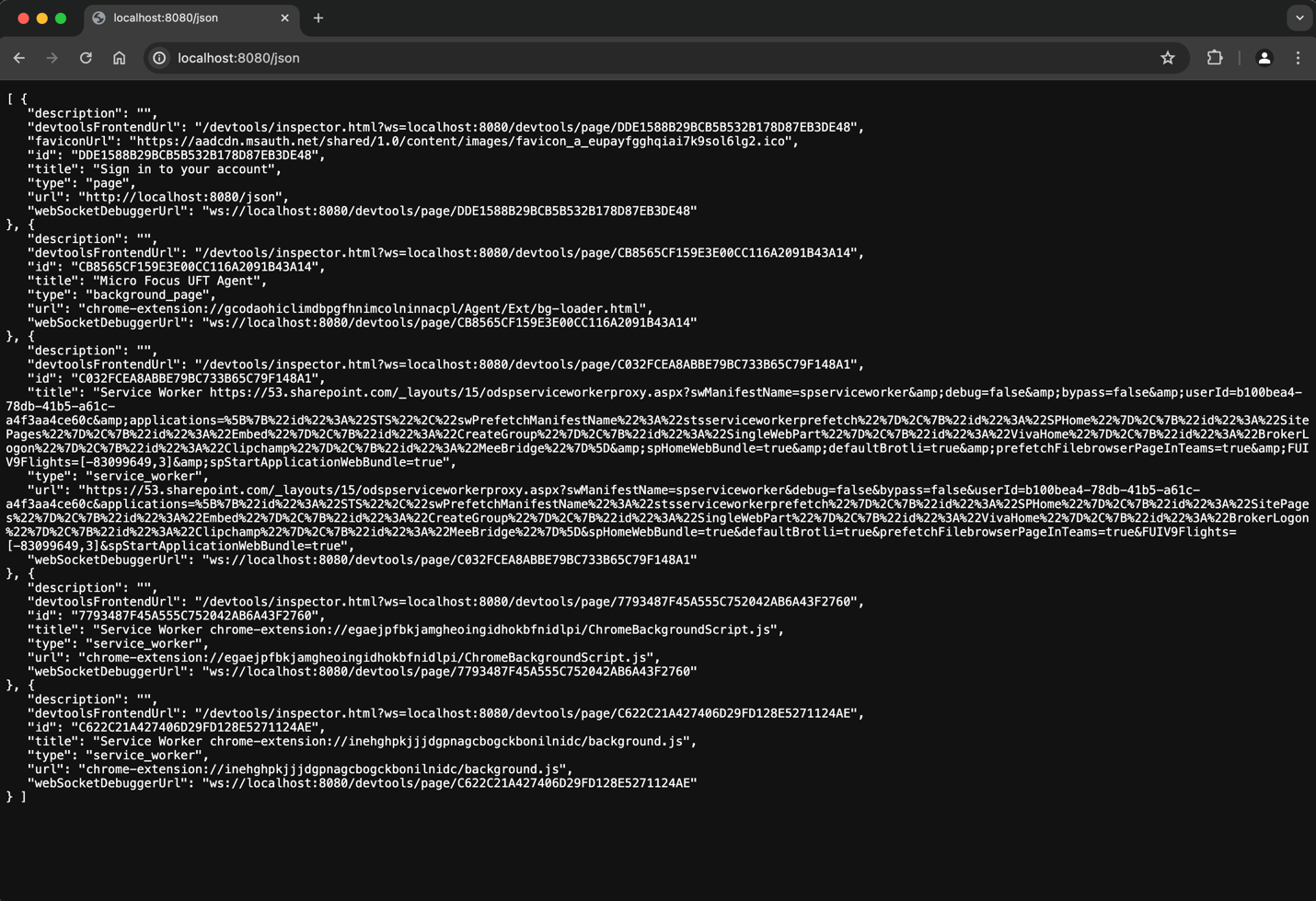
* 1. On Mac:

**/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome –remote-debugging-port=8080**

OR

**sudo /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome –remote-debugging-port=8080**

1. Verify the browser has launched by going to the url “http://localhost:8080/json”. If it launched correctly, you should see something like:



1. Login to <https://myapps.53.com/app/UserHome?session_hint=AUTHENTICATED>
2. Run the script from WDIO. It should navigate successfully to XAA in the already opened browser.

**Troubleshooting:**

* If you are unable to navigate to the expected screen at step 3:
  + Try steps 1-2 again, making sure chrome is completely closed
  + If that doesn’t work, the port “8080” may be occupied. Choose another set of 4 numbers to use as your port. Replace “8080” with your chosen number in the pre-requisite and steps 2 and 3.