* JSP and HTL(slightly template) will all be transformed to Java by AEM. So they can be debugged using <http://localhost:4502/system/console/fsclassloader>
* The way depfinder works is if you know there’s a certain path you have to include as a maven dependency, you can go into this tool, paste the path to the file (i.e. ‘com.example.File’) and it gives you the full markup needed to add the dependency to your POM file.

Note:

If you are trying to retrieve the source for a jsp or htl template in this way, make sure that you’ve actually run a page that requires that template on this server, otherwise it will not appear in the list.

# To Debug OOTB API

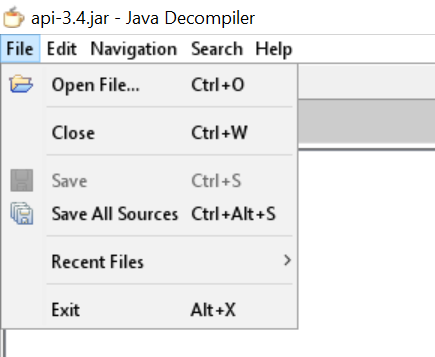
Try to find the file which is responsible for serving the request using <http://localhost:4502/system/console/components> or <http://localhost:4502/system/console/bundles>

Arrive at a bundle number for example like bundle427. This is exactly the bundle in the local file system.

Navigate to *<AEM installation folder>\crx-quickstart\launchpad\felix\bundle427\version<anything>*

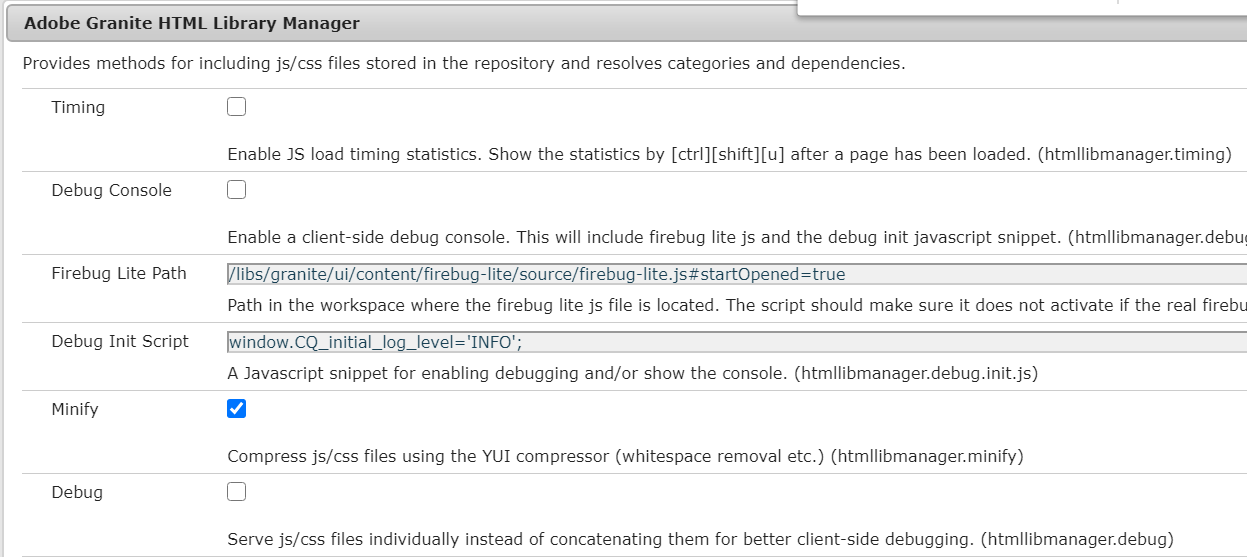
Use the latest version for best results

Use JD-GUI to decompile the above bundle and the decompiled class can be saved as zip using below

**

The above zip file can be imported into any IDE and breakpoints can be set for debugging.

* JS files are minified by default on prod because of Adobe Granite HTML Library Manager at <http://localhost:4502/system/console/configMgr>. For better debugging , check the debug checkbox which allows the js files to be served separately rather than in concatenated form.



Useful Resources : <https://engineering.icf.com/decompiling-and-debugging-aem/>