

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe. Adobe assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, the Creative Cloud logo, and the Adobe Marketing Cloud logo are either registered trademarks or trademarks of Adobe in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Adobe, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Module 10	Front-End Development using aemfed	3
	aemfed	4
	Exercise 1: Install aemfed (Local only)	5
	Exercise 2: Configure and run aemfed	7
	Exercise 3: Install content locally (Optional)	9
	Exercise 4: Add a component style using aemfed	10
	References	17

Front-End Development using aemfed

Introduction

aemfed is an open-source, command-line tool that can be used to speed-up front-end development.

Objectives

After completing this course, you will be able to:

- Install aemfed
- Explain development using aemfed

aemfed

aemfed is an open-source, Command line interface (CLI) that can be used to speed-up front-end development. It is powered by aemsync, Browsersync, and the Apache Sling Log Tracer.

Using aemsync, aemfed listens to file changes and automatically syncs the changes to a running AEM service. aemfed also determines which clientlibs are affected by the uploaded changes.

aemfed is also built to work with the Sling Log tracer to automatically display any server-side errors directly in the terminal window. It captures errors related to:

- Clientlibs
 - › Less compilation
 - › Javascript minification
- HTL templates
- JSP files
- .content.xml

If the error messages contain references to nodes in the JCR, it attempts to translate them back to the files on your local file system.

Exercise 1: Install aemfed (Local only)

In this exercise, you will install aemfed.

Prerequisite: You need to have NVM installed on your machine to complete this exercise. For instructions on how to install NVM, refer to **Module 6 - Creating project using Maven - Exercise 1 > Task 3**.



Note: You can skip this exercise if you use a ReadyTech environment because aemfed is already installed as part of the image.

1. Open a terminal window in your User directory.



Note: This global installation will require administrative privileges. On Windows, open command window using Run as admin option.

2. Install aemfed.

- Mac

```
$ npm install aemfed
```

- Windows

```
$ npm install aemfed --global
```



Note: The previous command will install aemfed globally. If you have a package.json for your AEM project, you can modify the package.json to add aemfed as a dev dependency:

```
$ npm install aemfed --save-dev
```

3. Show available options by entering the following command:

```
$ aemfed -h
```

```
C:\Program Files\nodejs>aemfed -h
Usage: aemfed [OPTIONS]
Options:
  -t targets           Default is http://admin:admin@localhost:4502
  -p proxy_port        Default is 3000
  -w path_to_watch     Default is current
  -e exclude_filter    Anymatch exclude filter; disabled by default
  -i sync_interval     Update interval in milliseconds; default is 100
  -o open_page         Browser page to be opened after successful launch; default is "false".
  -b browser           Browser where page should be opened in; this parameter is platform dependent; for example, Chrome is "google chrome" on OS X, "google-chrome" on Linux and "chrome" on Windows; default is "google chrome"
  -q load_qr           Enable QR code plugin for connected browsers; default is "true".
  -h                  Displays this screen
  -v                  Displays version of this package
```

Congratulations! You have installed aemfed.

Exercise 2: Configure and run aemfd

In this exercise you will enable the Apache Sling Log Tracer in AEM.

1. Enable the Apache Sling Log Tracer in AEM
2. Start aemfd.

Task 1: Enable the Apache Sling Log Tracer in AEM

1. Navigate to <http://localhost:4502/system/console/configMgr>.
2. Search for **Apache Sling Log Tracer** and open the configuration for the **Apache Sling Log Tracer**.
3. Update the configuration with the following values selected and **Save**.
 - › **Enabled**
 - › **Recording Servlet Enabled**

Apache Sling Log Tracer

Provides support for enabling log for specific loggers on per request basis. Refer to <http://sling.apache.org/documentation/bundles/log-tracers.html> for more details

Tracer Sets

oak-query : org.apache.jackrabbit.oak.query.QueryEngineImpl;level=debug

oak-writes : org.apache.jackrabbit.oak.jcr.operations.writes;level=trace

⚠ Default list of tracer sets configured. Tracer Set config confirms to following format. <set name> : <logger name>;level=<level name>, other loggers (tracerSets)

Enabled ☒

⚠ Enable the Tracer (enabled)

Recording Servlet Enabled ☒

⚠ Enable the Tracer Servlet. This servlet is required for the tracer recording feature to work and provides access to the json dump of the recording performed (servletEnabled)

Recording Cache Size 50

⚠ Recording cache size in MB which would be used to temporary cache the recording data (recordingCacheSizeInMB)

Recording Cache Duration 900

⚠ Time in seconds upto which the recording data would be held in memory before expiry (recordingCacheDurationInSecs)

Compress Recording ☒

⚠ Enable compression for recoding held in memory (recordingCompressionEnabled)

GZip Response ☒

⚠ If enabled the response sent would be compressed (gzipResponse)

Configuration Information

Persistent Identity (PID) org.apache.sling.tracer.internal.LogTracer

Configuration Binding Unbound or new configuration

Cancel Reset Delete Unbind **Save**

Task 2: Start aemfed

1. Open a new terminal window and navigate to your <AEM project directory>. To start aemfed and configure the AEM target service:

```
$ aemfed -t "http://admin:admin@localhost:4502" -w "ui.apps/src/main/content/jcr_root/"
```



Note: If you get a Windows Security Alert, click Allow Access.



Note: aemfed will return proxy URLs that you can use to see automated clientlib changes.

```
[Browsersync] Proxying: http://localhost:4502
[Browsersync] Access URLs:
-----
Local: http://localhost:3000
External: http://10.0.172.221:3000
-----
UI: http://localhost:3001
UI External: http://localhost:3001
-----
[QR Code] QR code of current url available in browser console using qr() or __browsersync__.qr()
```



Caution: Do not close this terminal window. It is controlling aemfed.

Exercise 3: Install content locally (Optional)

In this exercise, you will install the local Maven project into AEM.



Note: This exercise assumes you have a local Maven project. <AEM Project Directory> represents the location of this local Maven project.



Note: You can skip this exercise if you have already installed your AEM project.

1. Confirm that your AEM service is running.
2. Open a terminal window and navigate to your <AEM project directory>. Run the following command to deploy your AEM project

```
$mvn clean install -Padobe-public -PautoInstallSinglePackage
```

```
Administrator: Command Prompt
C:\Users\prpurush\training>mvn clean install -Padobe-public -PautoinstallPackage
```

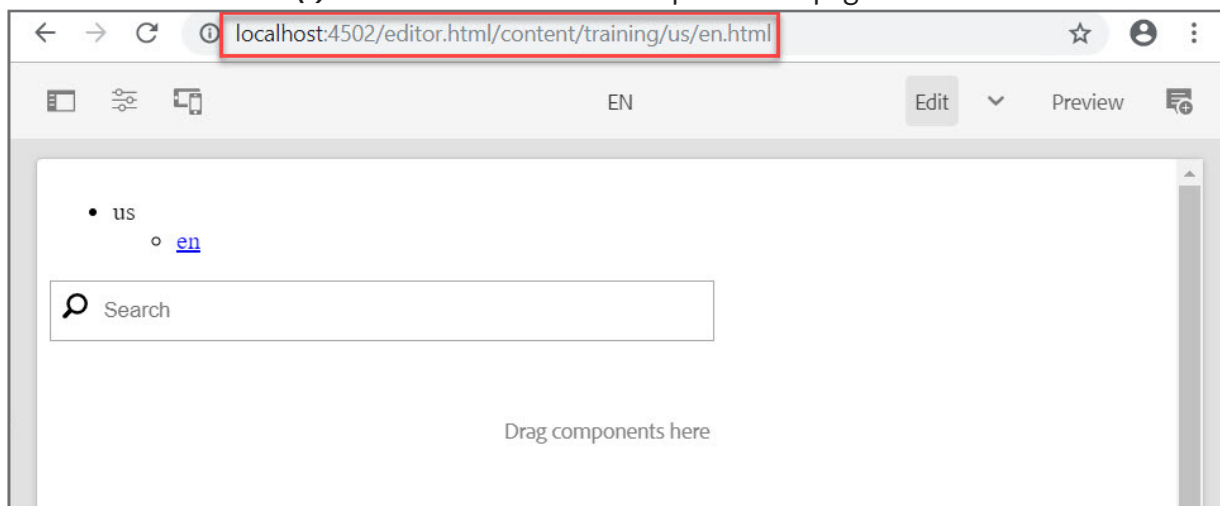
3. Verify the install is successful, as shown:

```
[INFO] training ..... SUCCESS [ 0.642 s]
[INFO] TrainingProject - Core ..... SUCCESS [ 13.140 s]
[INFO] TrainingProject - Repository Structure Package ..... SUCCESS [ 1.571 s]
[INFO] TrainingProject - UI apps ..... SUCCESS [ 25.962 s]
[INFO] TrainingProject - UI content ..... SUCCESS [ 22.935 s]
[INFO] TrainingProject - All ..... SUCCESS [ 3.877 s]
[INFO] TrainingProject - Integration Tests Bundles ..... SUCCESS [ 1.914 s]
[INFO] TrainingProject - Integration Tests Launcher ..... SUCCESS [ 6.151 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:18 min
[INFO] Finished at: 2019-12-03T11:11:27-08:00
[INFO] -----
```

Exercise 4: Add a component style using aemfed

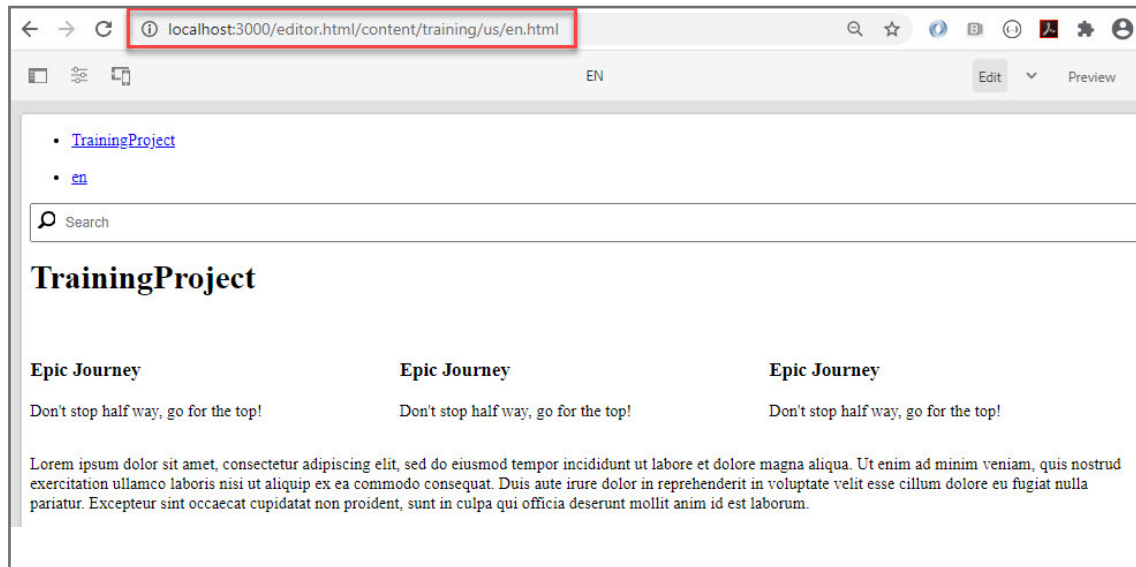
aemfed connects a local AEM server to a content package. In this exercise we will use a project created from the archetype, but you could connect with any unzipped content package.

1. Using your browser, navigate to **Sites > training > us** . Click the **en** page (thumbnail) to select it, and then click **Edit (e)** from the actions bar. This open the en page.



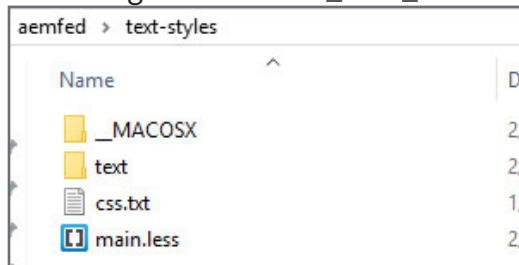
2. Drag a text component to the **en** Page and add some dummy text.

3. Modify the URL to point to the proxy server that is running. If you are using default ports, change 4502 (AEM) to port 3000 (aemfed) and refresh the browser.

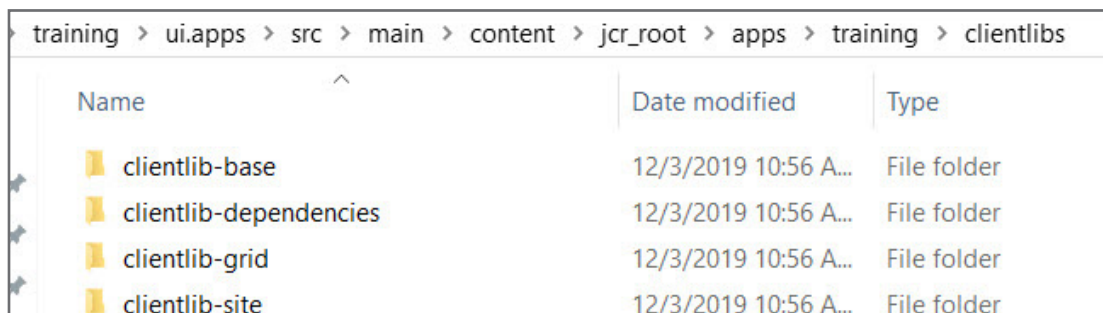


You should now see the same page content as your browser tab that displays content from **localhost:4502** but proxied via port **3000** (or whichever port the proxy is using).

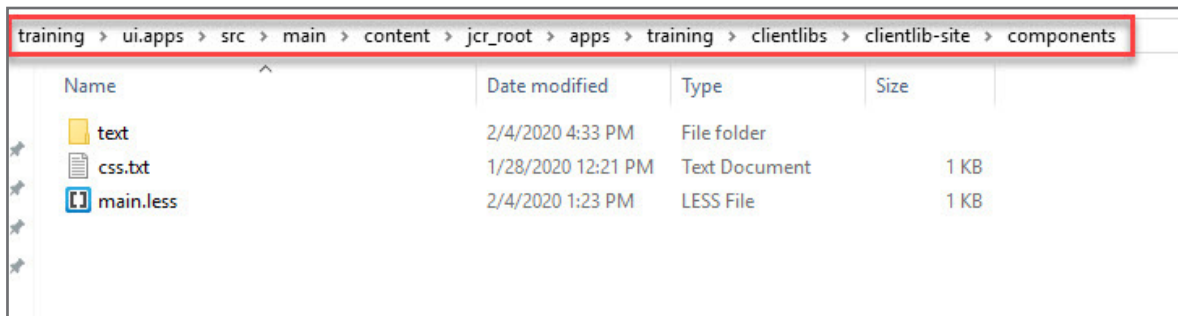
4. Navigate to **Exercise_Files_TB/aemfed/** and extract the contents of **text-styles.zip**, as shown:



5. Copy the folder **text**.
6. In a file window, navigate to **/<AEM Project Directory>/ui.apps/src/main/content/jcr_root/apps/training/clientlibs/clientlib-site**



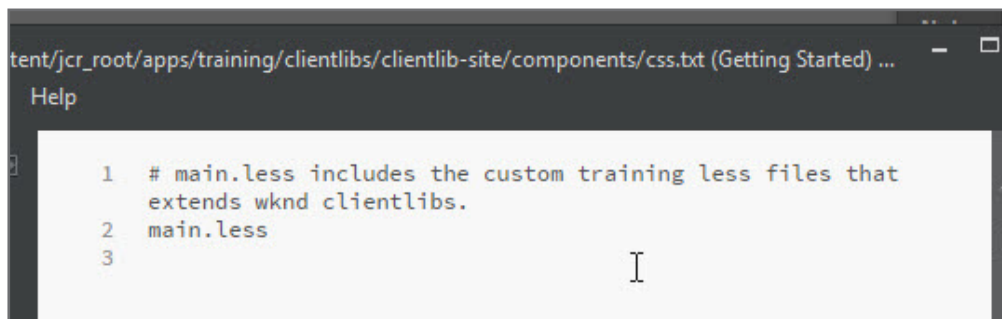
7. Paste the **text** folder from **Exercise_Files_TB > aemfed** into the **clientlib-site/components** folder (create **components** folder if needed):



8. Verify you see messages in the aemfed terminal window similar to the following screenshot:

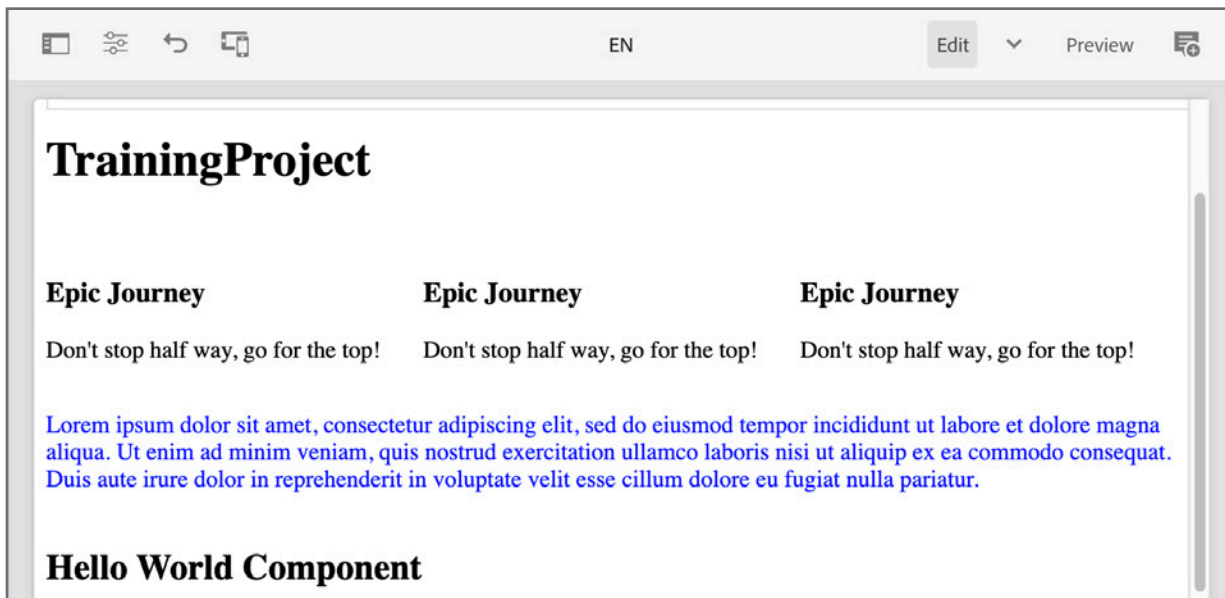
```
[localhost:4502] Special paths were changed, so rebuild clientlib tree
[localhost:4502] Get data from server: 252 ms
[localhost:4502] Process data: 18 ms
[localhost:4502] Clientlib tree: 271 ms
[localhost:4502] Rebuild clientlib tree: 271 ms
[Browsersync] Reloading Browsers...
ADD jcr_root/apps/trainingproject/clientlibs/.DS_Store
Deploying to [localhost:4502] in 43 ms at 2019-08-23T15:41:39.645Z: OK
[Browsersync] Reloading Browsers...
```

9. Even though initial text style has been loaded, it needs to be added to the design. If you already have a working design in your project, skip to step 11.
10. Using the IDE of your choice, navigate into the **/clientlib-site/** folder and open **css.txt** and add the line **main.less**.



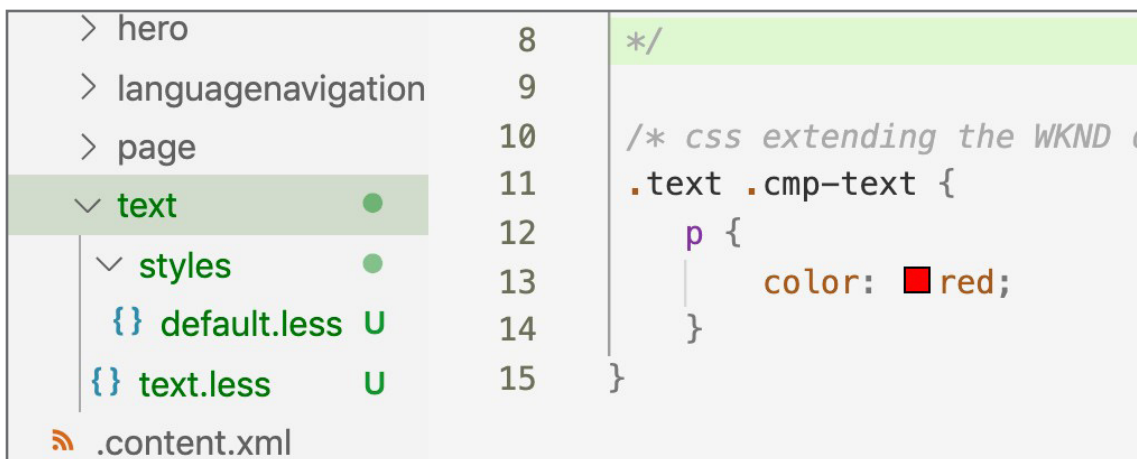
11. Save **css.txt** and close.
12. Locate the **main.less** file under **clientlibs/clientlib-site**. If the file does not exist, create it.
13. Open **main.less** and add the following line:


```
@import "components/text/text.less";
```
14. Save **main.less** and close.
15. Go back to the browser tab <http://localhost:3000/editor.html/content/training/us/en.html> and observe the text color has changed to blue ,as shown:



You should see the browser showing the proxy page change and messages regarding the styling change in the aemfd window.

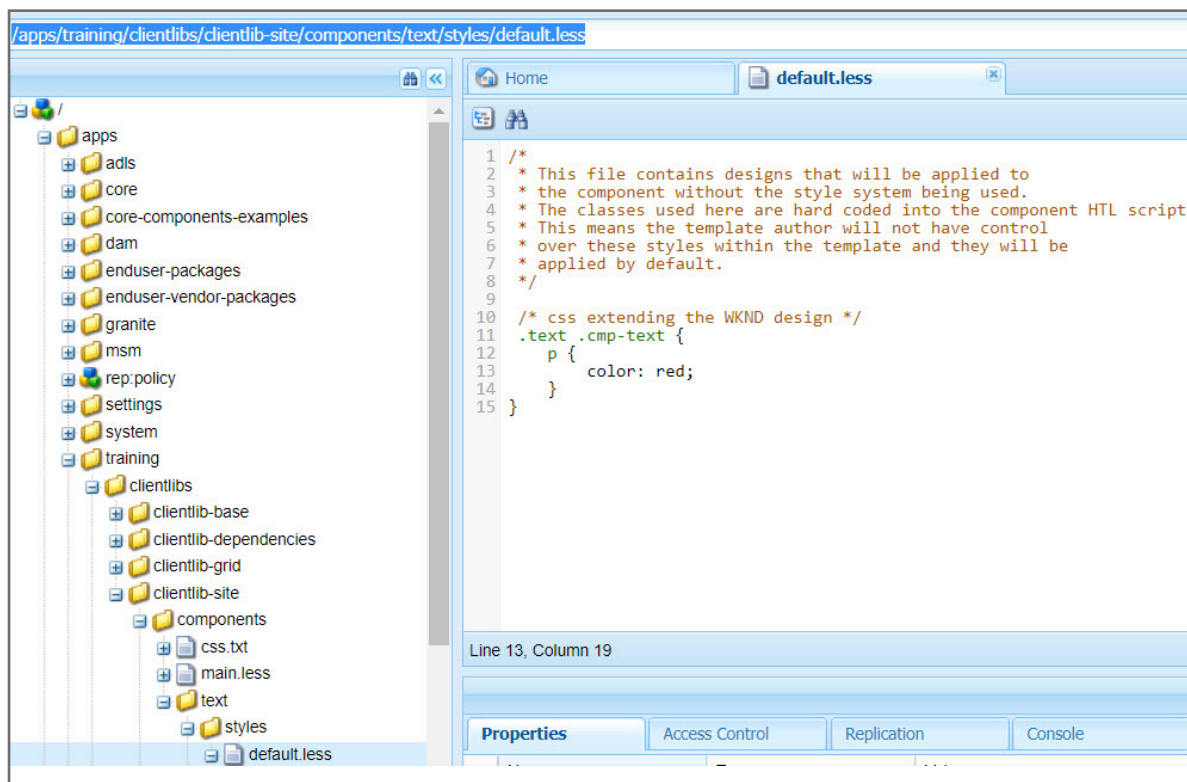
16. Using the IDE of your choice, navigate to **clientlib-site/components/text/styles** and open **default.less**. Update the style to a color of your choice, such as **red**.



You will notice messages in the aemfd window each time you open or modify a file.

```
[Browsersync] Reloading Browsers...
ADD jcr_root/apps/trainingproject/clientlibs/clientlib-site/components/text/text.less
  Deploying to [localhost:4502] in 82 ms at 2019-08-23T15:49:22.486Z: OK
[localhost:4502] Only styling was changed, try to inject
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-base.css
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-site.css
ADD jcr_root/apps/trainingproject/clientlibs/clientlib-site/components/text/styles/default.less
  Deploying to [localhost:4502] in 43 ms at 2019-08-23T15:49:24.228Z: OK
[localhost:4502] Only styling was changed, try to inject
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-base.css
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-site.css
```

17. Using CRXDE Lite, navigate to `/apps/training/clientlibs/clientlib-site/components/text/styles/default.less` and verify that the files were uploaded to AEM.



You should see the browser showing the proxy page change and messages regarding the styling change in the aemfd window.

TIP: If you did not see the expected behavior try the following:

Clear browser cache:

1. Open Settings in the browser. In Chrome, it is the three vertical dots at the end of the address bar and then choose Settings from the menu.
2. In the left rail, click **Advanced** to open the menu.
3. Click **Privacy and Security** from the Advanced menu.
4. Click **Clear browsing data** from the **Privacy and security** dialog.
5. From the list in the popup window, check the checkbox for **Cached images and files** and click **Clear data**.
6. Navigate to: <http://localhost:4502/content/training/en.html>. You should see the text rendered in the desired color.
7. You can also check by right-clicking on the page and selecting View source. Search for `<link rel="stylesheet" href="/etc.clientlibs/training/clientlibs/clientlib-base.css" type="text/css">` in the source output.
8. Click on the css link. The concatenated css file should open.
9. Search for the element that you set in **default.less** and verify the color value.

Clear Server cache:

1. Using **CRXDE Lite**, verify that the training site is included in the dependency property for **/apps/training/clientlibs/clientlib-base**.
2. In your browser, navigate to: <http://localhost:4502/libs/granite/ui/content/dumplib.html>
3. At the top of the page, select Click **here** for rebuilding all libraries.
4. Click **Invalidate Caches** to mark the client library caches invalid and force a recompile.



Caution: Do not click Rebuild Libraries. This is an unnecessary and extensive process and will take quite a while.

Tip: To force a single library to rebuild, rename the cq:clientLibraryFolder node, in this case the **/apps/training/clientlibs/clientlib-base** node.

5. Navigate to <http://localhost:4502/content/training/en.html>. You should see the text rendered in the desired color. You should also see the correct color in the aemfd proxy rendered page.

Now, test the Sling Log Tracer.

18. Using your IDE, change the font color to green and remove the semicolon at the end of the line to force an error. Save the file.

```
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-base.css
[Browsersync] File event [change] : /apps/trainingproject/clientlibs/clientlib-site.css
[localhost:4502] Tracer output for [/etc.clientlibs/trainingproject/clientlibs/clientlib-base.css?browsersync=1566576729776] (def7a936-ceca-43ac-ace6-8a7b02436c93)
[ERROR] LessCompilerImpl: failed to compile less /apps/trainingproject/clientlibs/clientlib-site/main.less: ParseError: Unrecognised input in /apps/trainingproject/clientlibs/clientlib-site/components/text/styles/default.less on line 6, column 5:
5
6   p {
7     color: green
Local source: /Adobe/Archetype19/training/ui.apps/src/main/content/jcr_root/apps/trainingproject/clientlibs/clientlib-site/components/text/styles/default.less:6:5
```



Note: The Sling Log Tracer outputs the less compiler error message and displays the offending line number in the file. There is no need to go digging through the error log to find the problem.

TIP: If you did not see the expected behavior, try the following:

Rebuild the client libraries:

If the recompile is not attempted, the error in default.less will not be caught. In this case, you can try two options:

1. Rename the cq:clientLibraryFolder node; (in this case: /apps/training/clientlibs/clientlib-base) to force a recompile.
 2. If that does not produce the desired results, navigate to <http://localhost:4502/libs/granite/ui/content/dumplib.html>.
 3. At the top of the page select Click here for rebuilding all libraries.
 4. Click Rebuild Libraries. Now, if you change the default.less file again and deliberately introduce an error, the system should attempt a recompile and the error will be reported by the Sling Log Tracer and logged to the aemfed command window.
-

19. Using your IDE, correct the error and notice the font color change.

Congratulations! You have used aemfed to assist with AEM front-end development.

References

You can use the following links for more information on:

- aemfed: <https://aemfed.io/>