New issue   Jump to bottom

Reduce memory consumption during content-build #15601

Merged **ncksllvn** merged 8 commits into `master` from `nps--fix-memory-issue` on Jan 13   Code Climate ✓

Conversation 24   Commits 8   Checks 4   **Files changed 24**

Jump to...

33 src/site/stages/build/index.js

```
  const registerLiquidFilters = require('../../filters/liquid');
  const { getDrupalContent } = require('./drupal/metalsmith-drupal');
  const addDrupalPrefix = require('./plugins/add-drupal-prefix');
- const addNonceToScripts = require('./plugins/add-nonce-to-scripts');
- const addSubheadingsIds = require('./plugins/add-id-to-subheadings');
- const checkBrokenLinks = require('./plugins/check-broken-links');
  const checkCollections = require('./plugins/check-collections');
  const checkForCMSUrls = require('./plugins/check-cms-urls');
  const downloadAssets = require('./plugins/download-assets');
  const readAssetsFromDisk = require('./plugins/read-assets-from-disk');
- const processEntryNames = require('./plugins/process-entry-names');
  const createDrupalDebugPage = require('./plugins/create-drupal-debug');
  const createEnvironmentFilter = require('./plugins/create-environment-filter');
  const createHeaderFooter = require('./plugins/create-header-footer');
  const createSitemaps = require('./plugins/create-sitemaps');
  const downloadDrupalAssets = require('./plugins/download-drupal-assets');
  const leftRailNavResetLevels = require('./plugins/left-rail-nav-reset-levels');
- const parseHtml = require('./plugins/parse-html');
- const replaceContentsWithDom = require('./plugins/replace-contents-with-dom');
- const injectAxeCore = require('./plugins/inject-axe-core');
+ const modifyDom = require('./plugins/modify-dom');
  const rewriteDrupalPages = require('./plugins/rewrite-drupal-pages');
  const rewriteVaDomains = require('./plugins/rewrite-va-domains');
- const updateExternalLinks = require('./plugins/update-external-links');
  const updateRobots = require('./plugins/update-robots');

  /**
    smith.use(createSitemaps(BUILD_OPTIONS), 'Create sitemap');
    smith.use(updateRobots(BUILD_OPTIONS), 'Update robots.txt');
    smith.use(checkForCMSUrls(BUILD_OPTIONS), 'Check for CMS URLs');
-
-   /**
-    * Parse the HTML into a JS data structure for use in later plugins.
-    * Important: Only plugins that use the parsedContent to modify the
-    * content can go between the parseHtml and outputHtml plugins. If
-    * the content is modified directly between those two plugins, any
-    * changes will be overwritten during the outputHtml step.
-    */
-   smith.use(parseHtml, 'Parse HTML files');
-
-   /**
-    * Add nonce attribute with substitution string to all inline script tags
-     * Convert onclick event handles into nonced script tags
-     */
-    smith.use(addNonceToScripts, 'Add nonce to script tags');
     smith.use(
-      processEntryNames(BUILD_OPTIONS),
-      'Process [data-entry-name] attributes into Webpack asset paths',
+      modifyDom(BUILD_OPTIONS),
+      'Parse a virtual DOM from every .html file and perform a variety of DOM sub-operations on each file',
     );
-    smith.use(updateExternalLinks(BUILD_OPTIONS), 'Update external links');
-    smith.use(addSubheadingsIds(BUILD_OPTIONS), 'Add IDs to subheadings');
-    smith.use(checkBrokenLinks(BUILD_OPTIONS), 'Check for broken links');
-    smith.use(injectAxeCore(BUILD_OPTIONS), 'Inject axe-core for accessibility');
-    smith.use(replaceContentsWithDom, 'Save the changes from the modified DOM');

     /* eslint-disable no-console */
     smith.build(err => {
```

113 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/add-id-to-subheadings.js

Load diff

This file was deleted.

65 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/check-broken-links/index.js

Load diff

This file was deleted.

122 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/check-broken-links/tests/index.unit.spec.js

Load diff

This file was deleted.

52 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/inject-axe-core.js

Load diff

This file was deleted.

106 🟩🟩🟩🟩🟩 src/site/stages/build/plugins/modify-dom/add-id-to-subheadings.js

```
@@ -0,0 +1,106 @@
+ /* eslint-disable no-param-reassign */
```

This conversation was marked as resolved by **nckslvn**

Show conversation

```
+
+ /*
+  * Add unique ID to H2s and H3s that aren't in WYSIWYG or accordion buttons
+  */
+ const { ENTITY_BUNDLES } = require('../../../../constants/content-modeling');
+
+ const entityBundlesForResourcesAndSupport = new Set([
+   ENTITY_BUNDLES.STEP_BY_STEP,
+   ENTITY_BUNDLES.FAQ_MULTIPLE_Q_A,
+   ENTITY_BUNDLES.Q_A,
+   ENTITY_BUNDLES.CHECKLIST,
+   ENTITY_BUNDLES.MEDIA_LIST_IMAGES,
+   ENTITY_BUNDLES.MEDIA_LIST_VIDEOS,
+   ENTITY_BUNDLES.SUPPORT_RESOURCES_DETAIL_PAGE,
+ ]);
+
+ function createUniqueId(headingEl, headingOptions) {
+   const headingString = headingEl.text();
+   const length = 30;
+   let anchor = headingString
+     .trim()
+     .toLowerCase()
+     .replace(/[^\w\- ]+/g, '')
+     .replace(/\s/g, '-')
+     .replace(/-+$/, '')
+     .substring(0, length);
+
+   if (headingOptions.previousHeadings.includes(anchor)) {
+     anchor += `-${headingOptions.getHeadingId()}`;
+   }
+
+   headingOptions.previousHeadings.push(anchor);
+   return anchor;
+ }
+
+ module.exports = {
+   modifyFile(fileName, file) {
+     let idAdded = false;
+
+     if (fileName.endsWith('html')) {
+       const { dom } = file;
+       const tableOfContents = dom('#table-of-contents ul');
+
+       if (!tableOfContents) {
+         return;
+       }
+
+       const headingOptions = {
+         previousHeadings: [],
+         previousId: 0,
+         getHeadingId() {
+           return ++this.previousId;
+         },
+       };
+
+       let nodes = null;
+
+       if (entityBundlesForResourcesAndSupport.has(file.entityBundle)) {
+         nodes = dom('article h2');
+         if (nodes.length < 2) {
```

```
+          return;
+        }
+      } else {
+        nodes = dom('h2, h3');
+      }
+
+      nodes.each((index, el) => {
+        const heading = dom(el);
+        const parent = heading.parents();
+        const isInAccordionButton = parent.hasClass('usa-accordion-button');
+        const isInAccordion = parent.hasClass('usa-accordion-content');
+
+        // skip heading if it already has an id and skip heading if it's in an accordion button
+        if (!heading.attr('id') && !isInAccordionButton) {
+          const headingID = createUniqueId(heading, headingOptions);
+          heading.attr('id', headingID);
+          idAdded = true;
+        }
+
+        // if it is an h2, add the h2 to the table of contents
+        if (
+          el.tagName.toLowerCase() === 'h2' &&
+          tableOfContents &&
+          heading.text().toLowerCase() !== 'on this page' &&
+          !isInAccordionButton &&
+          !isInAccordion
+        ) {
+          tableOfContents.append(
+            `<li class="vads-u-margin-bottom--2"><a href="#${heading.attr(
+              'id',
+            )}" onClick="recordEvent({ event: 'nav-jumplink-click' });"
+              class="vads-u-display--flex vads-u-text-decoration--none">
+              <i class="fas fa-arrow-down va-c-font-size--xs vads-u-margin-top--1 vads-u-margin-right--1">
```

This conversation was marked as resolved by **nckslIvn** Show conversation

```
+              </i>${heading.text()}</a></li>`,
+          );
+          idAdded = true;
+        }
+      });
+
+      if (idAdded) {
+        file.modified = true;
+      }
+    }
+  },
+ };
```

nckslIvn on Jan 12 Author Contributor

This file is basically the same as its predecessor (the file was moved into the `modify-dom` dir) but now wrapped in the `modifyFile` hook

11 ...ges/build/plugins/add-nonce-to-scripts.js → ...lugins/modify-dom/add-nonce-to-scripts.js

```
    };
  }

- module.exports = (files, metalsmith, done) => {
-   Object.keys(files).forEach(fileName => {
+ module.exports = {
```

```
+   modifyFile(fileName, file) {
      if (path.extname(fileName) !== '.html') return;

-     const { dom } = files[fileName];
+     const { dom } = file;
      dom('script').each((index, scriptEl) => {
        const s = dom(scriptEl);
        // Only add nonce to inline scripts
      newScript.attr('nonce', CSP_NONCE);

      dom('body').append(newScript);
-     files[fileName].modified = true;
-   });
-   done();
+     file.modified = true;
+   },
  };
```

0 ...roken-links/helpers/applyIgnoredRoutes.js → ...roken-links/helpers/applyIgnoredRoutes.js

File renamed without changes.

0 ...ck-broken-links/helpers/getBrokenLinks.js → ...ck-broken-links/helpers/getBrokenLinks.js

File renamed without changes.

0 ...ck-broken-links/helpers/getErrorOutput.js → ...ck-broken-links/helpers/getErrorOutput.js

File renamed without changes.

0 ...heck-broken-links/helpers/isBrokenLink.js → ...heck-broken-links/helpers/isBrokenLink.js

File renamed without changes.

51 src/site/stages/build/plugins/modify-dom/check-broken-links/index.js

```
@@ -0,0 +1,51 @@
+ /* eslint-disable no-param-reassign, no-console, no-continue */
```

This conversation was marked as resolved by **nckslivn**

Show conversation

```
+
+ const path = require('path');
+
+ const getBrokenLinks = require('./helpers/getBrokenLinks');
+ const applyIgnoredRoutes = require('./helpers/applyIgnoredRoutes');
+ const getErrorOutput = require('./helpers/getErrorOutput');
+
+ /**
+  * Metalsmith middleware for verifying HREF/SRC values in HTML files are valid file references.
+  */
+
+ module.exports = {
+   initialize(buildOptions, files) {
+     const fileNames = Object.keys(files);
+     this.allPaths = new Set(fileNames);
+     this.brokenPages = [];
+   },
+
+   modifyFile(fileName, file, files, buildOptions) {
+     if (buildOptions.watch) {
+       return;
+     }
+
+     const isHtml = path.extname(fileName) === '.html';
+     if (!isHtml) return;
+
+     const linkErrors = getBrokenLinks(file, this.allPaths);
+
+     if (linkErrors.length > 0) {
+       this.brokenPages.push({
+         path: file.path,
+         linkErrors,
+       });
+     }
+   },
+
+   conclude(buildOptions, files) {
+     this.brokenPages = applyIgnoredRoutes(this.brokenPages, files);
+
+     if (this.brokenPages.length > 0) {
+       const errorOutput = getErrorOutput(this.brokenPages);
+
+       if (buildOptions['drupal-fail-fast']) {
+         throw new Error(errorOutput);
+       }
+
+       console.log(errorOutput);
+     }
+   },
+ };
```

nckslivn on Jan 12 Author Contributor

Similar to the `add-id-to-subheadings` this file was just refactored to go into a `modifyFile` hook

`0 ...s/helpers/applyIgnoredRoutes.unit.spec.js → ...s/helpers/applyIgnoredRoutes.unit.spec.js`

File renamed without changes.

`0 ...tests/helpers/getBrokenLinks.unit.spec.js → ...tests/helpers/getBrokenLinks.unit.spec.js`

File renamed without changes.

`0 ...tests/helpers/getErrorOutput.unit.spec.js → ...tests/helpers/getErrorOutput.unit.spec.js`

File renamed without changes.

`0 ...s/tests/helpers/isBrokenLink.unit.spec.js → ...s/tests/helpers/isBrokenLink.unit.spec.js`

File renamed without changes.

`50 src/site/stages/build/plugins/modify-dom/index.js`

```
... ...  @@ -0,0 +1,50 @@
```

```
+ /* eslint-disable no-param-reassign */
```

This conversation was marked as resolved by **ncksllvn**

Show conversation

```
+ const path = require('path');
+ const cheerio = require('cheerio');
+ const addNonceToScripts = require('./add-nonce-to-scripts');
+ const processEntryNames = require('./process-entry-names');
+ const updateExternalLinks = require('./update-external-links');
+ const addSubheadingsIds = require('./add-id-to-subheadings');
+ const checkBrokenLinks = require('./check-broken-links');
+ const injectAxeCore = require('./inject-axe-core');
+
+ const domModifiers = [
+   addNonceToScripts,
+   processEntryNames,
+   updateExternalLinks,
+   addSubheadingsIds,
+   checkBrokenLinks,
+   injectAxeCore,
+ ];
+
+ const modifyDom = BUILD_OPTIONS => files => {
+   for (const modifier of domModifiers) {
+     if (modifier.initialize) {
+       modifier.initialize(BUILD_OPTIONS, files);
+     }
+   }
+
+   // Store only one `file.dom` in memory at a time
+   // because storing the virtual DOM of every .html file in memory
+   // at once would cause a massive amount of memory to be consumed.
+   for (const [fileName, file] of Object.entries(files)) {
+     if (path.extname(fileName) === '.html') {
+       file.dom = cheerio.load(file.contents);
+       for (const modifier of domModifiers) {
+         modifier.modifyFile(fileName, file, files, BUILD_OPTIONS);
+       }
+       if (file.modified) {
+         file.contents = Buffer.from(file.dom.html());
+       }
+       delete file.dom;
+     }
+   }
+
+   for (const modifier of domModifiers) {
+     if (modifier.conclude) {
+       modifier.conclude(BUILD_OPTIONS, files);
+     }
+   }
+ };
+
+ module.exports = modifyDom;
```

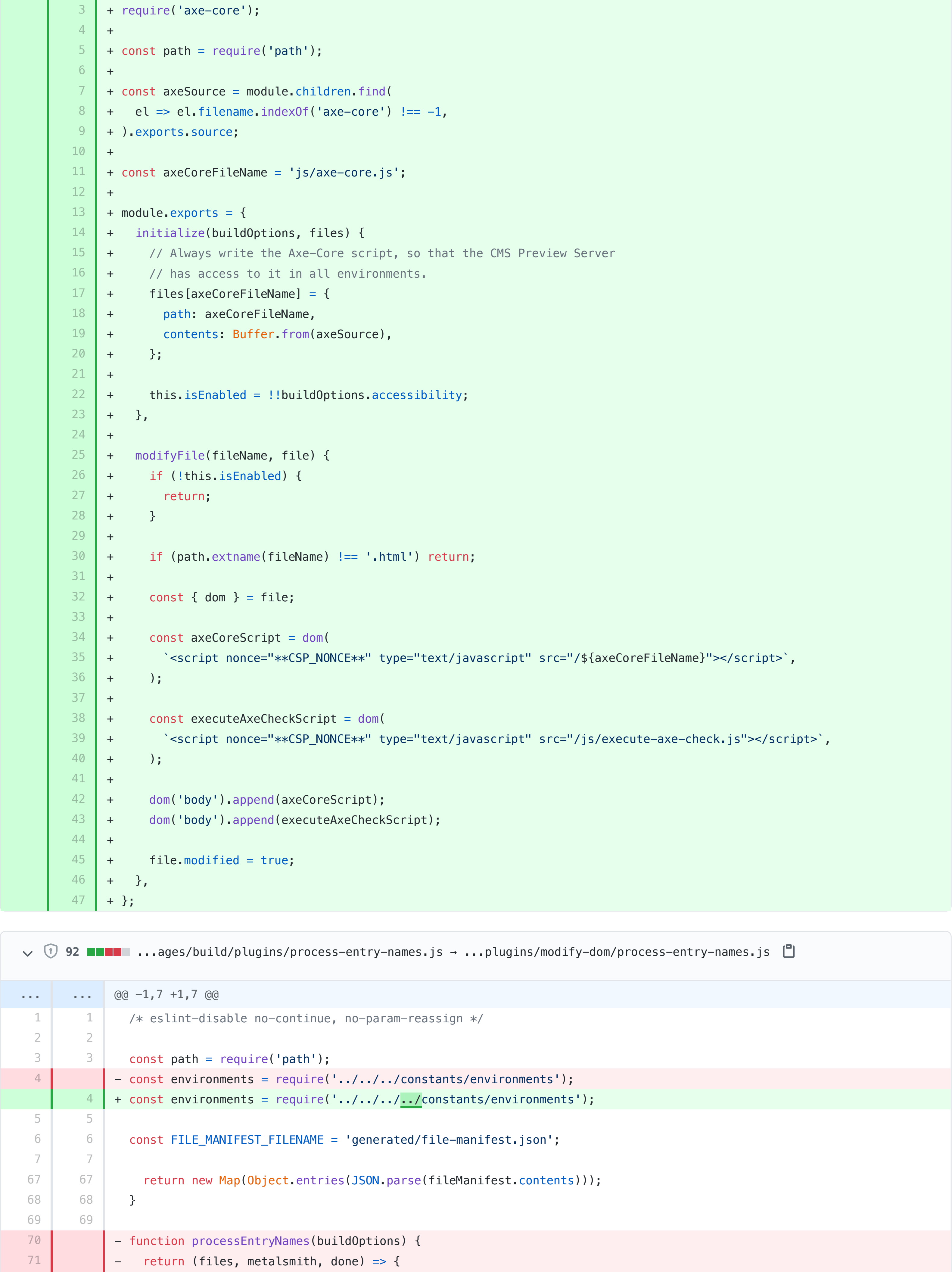
47 ■■■■■ src/site/stages/build/plugins/modify-dom/inject-axe-core.js

```
...    ...    @@ -0,0 +1,47 @@
+ /* eslint-disable no-continue, no-param-reassign */
```

This conversation was marked as resolved by **ncksllvn**

Show conversation

```
+
```



```
+ require('axe-core');
+
+ const path = require('path');
+
+ const axeSource = module.children.find(
+   el => el.filename.indexOf('axe-core') !== -1,
+ ).exports.source;
+
+ const axeCoreFileName = 'js/axe-core.js';
+
+ module.exports = {
+   initialize(buildOptions, files) {
+     // Always write the Axe-Core script, so that the CMS Preview Server
+     // has access to it in all environments.
+     files[axeCoreFileName] = {
+       path: axeCoreFileName,
+       contents: Buffer.from(axeSource),
+     };
+
+     this.isEnabled = !!buildOptions.accessibility;
+   },
+
+   modifyFile(fileName, file) {
+     if (!this.isEnabled) {
+       return;
+     }
+
+     if (path.extname(fileName) !== '.html') return;
+
+     const { dom } = file;
+
+     const axeCoreScript = dom(
+       `<script nonce="**CSP_NONCE**" type="text/javascript" src="/${axeCoreFileName}"></script>`,
+     );
+
+     const executeAxeCheckScript = dom(
+       `<script nonce="**CSP_NONCE**" type="text/javascript" src="/js/execute-axe-check.js"></script>`,
+     );
+
+     dom('body').append(axeCoreScript);
+     dom('body').append(executeAxeCheckScript);
+
+     file.modified = true;
+   },
+ };
```

92 🟩🟩🟥🟥 ...ages/build/plugins/process-entry-names.js → ...plugins/modify-dom/process-entry-names.js

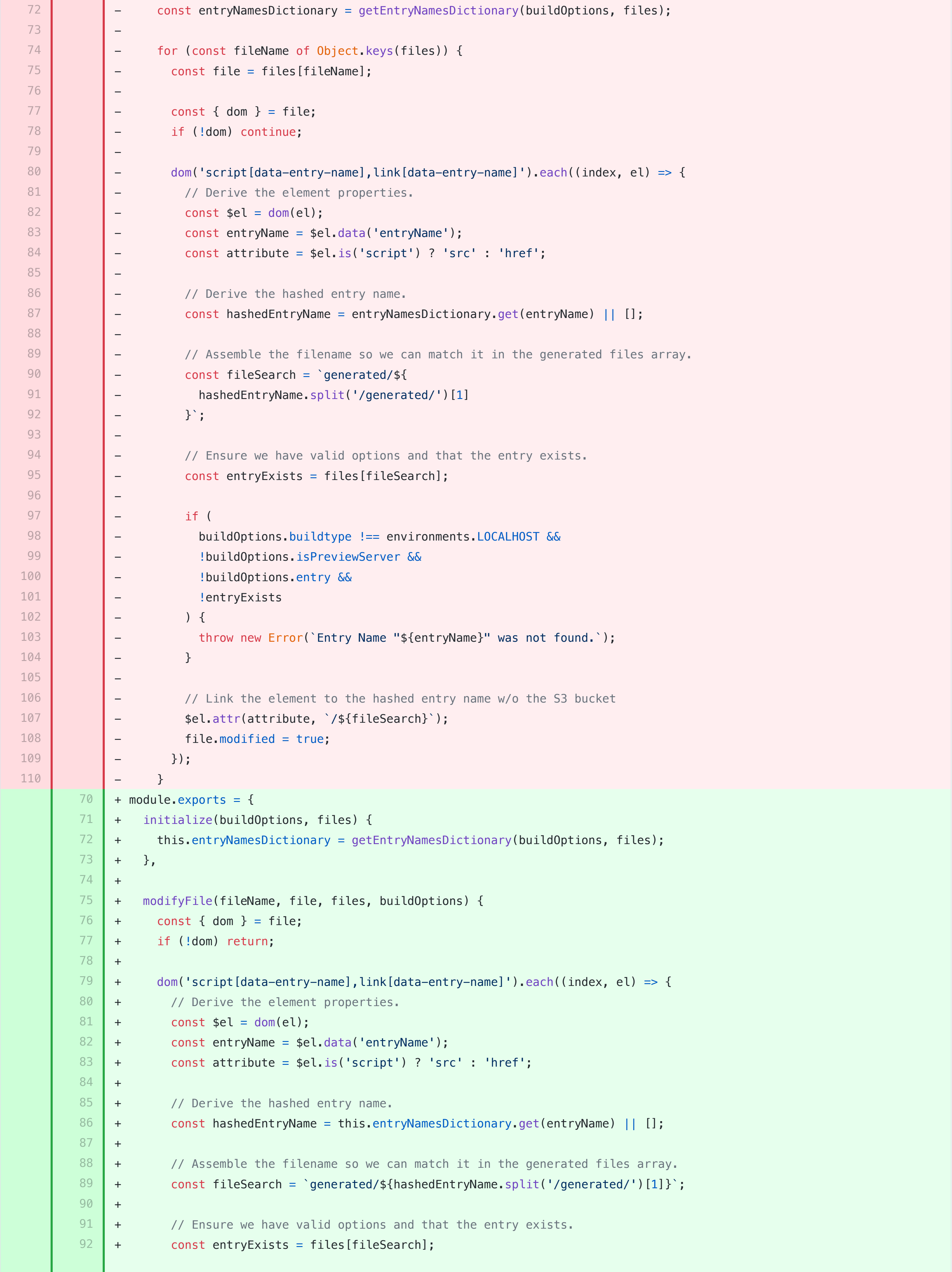
```
@@ -1,7 +1,7 @@
  /* eslint-disable no-continue, no-param-reassign */

  const path = require('path');
- const environments = require('../../../constants/environments');
+ const environments = require('../../../../constants/environments');

  const FILE_MANIFEST_FILENAME = 'generated/file-manifest.json';

    return new Map(Object.entries(JSON.parse(fileManifest.contents)));
  }

- function processEntryNames(buildOptions) {
-   return (files, metalsmith, done) => {
```



```diff
-     const entryNamesDictionary = getEntryNamesDictionary(buildOptions, files);
-
-     for (const fileName of Object.keys(files)) {
-       const file = files[fileName];
-
-       const { dom } = file;
-       if (!dom) continue;
-
-       dom('script[data-entry-name],link[data-entry-name]').each((index, el) => {
-         // Derive the element properties.
-         const $el = dom(el);
-         const entryName = $el.data('entryName');
-         const attribute = $el.is('script') ? 'src' : 'href';
-
-         // Derive the hashed entry name.
-         const hashedEntryName = entryNamesDictionary.get(entryName) || [];
-
-         // Assemble the filename so we can match it in the generated files array.
-         const fileSearch = `generated/${
-           hashedEntryName.split('/generated/')[1]
-         }`;
-
-         // Ensure we have valid options and that the entry exists.
-         const entryExists = files[fileSearch];
-
-         if (
-           buildOptions.buildtype !== environments.LOCALHOST &&
-           !buildOptions.isPreviewServer &&
-           !buildOptions.entry &&
-           !entryExists
-         ) {
-           throw new Error(`Entry Name "${entryName}" was not found.`);
-         }
-
-         // Link the element to the hashed entry name w/o the S3 bucket
-         $el.attr(attribute, `/${fileSearch}`);
-         file.modified = true;
-       });
-     }
+ module.exports = {
+   initialize(buildOptions, files) {
+     this.entryNamesDictionary = getEntryNamesDictionary(buildOptions, files);
+   },
+
+   modifyFile(fileName, file, files, buildOptions) {
+     const { dom } = file;
+     if (!dom) return;
+
+     dom('script[data-entry-name],link[data-entry-name]').each((index, el) => {
+       // Derive the element properties.
+       const $el = dom(el);
+       const entryName = $el.data('entryName');
+       const attribute = $el.is('script') ? 'src' : 'href';
+
+       // Derive the hashed entry name.
+       const hashedEntryName = this.entryNamesDictionary.get(entryName) || [];
+
+       // Assemble the filename so we can match it in the generated files array.
+       const fileSearch = `generated/${hashedEntryName.split('/generated/')[1]}`;
+
+       // Ensure we have valid options and that the entry exists.
+       const entryExists = files[fileSearch];
+
+       if (
+         buildOptions.buildtype !== environments.LOCALHOST &&
+         !buildOptions.isPreviewServer &&
+         !buildOptions.entry &&
+         !entryExists
+       ) {
+         throw new Error(`Entry Name "${entryName}" was not found.`);
+       }

+       // Link the element to the hashed entry name w/o the S3 bucket
+       $el.attr(attribute, `/${fileSearch}`);
+       file.modified = true;
+     });
+   },
+
+   conclude(buildOptions, files) {
      if (!buildOptions.isPreviewServer) {
-       copyAssetsToTeamSitePaths(buildOptions, files, entryNamesDictionary);
+       copyAssetsToTeamSitePaths(buildOptions, files, this.entryNamesDictionary);
```

ncksllvn on Jan 12 Author Contributor

I'm not certain this `copyAssetsToTeamSitePaths` function really does anything anymore. It seems like whatever is going on here has replaced it,

vets-website/config/webpack.config.js
Line 25 in b46b14f

```
25    // TODO: refactor the other approach for creating files without the hash so that we're or
      doing that in the webpack config: https://github.com/department-of-veterans-affairs/vets-
      website/blob/a012bad17e5bf024b0ea7326a72ae6a737e349ec/src/site/stages/build/plugins/proce
      entry-names.js#L35
```

.

ncksllvn on Jan 12 Author Contributor

cc **@timwright12** **@cvalarida**

```
      }
-
-     done();
-   };
- }
-
- module.exports = processEntryNames;
+   },
+ };
```

70 src/site/stages/build/plugins/modify-dom/update-external-links.js

```
@@ -0,0 +1,70 @@
+ /* eslint-disable no-param-reassign */
```

This conversation was marked as resolved by **ncksllvn** Show conversation

```
+ const newTabDomains = [
+   'myhealth.va.gov',
+   'ebenefits.va.gov',
+   'www.accesstocare.va.gov',
+   'www.gibill.va.gov/wave',
+   'www.blogs.va.gov',
+   'www.data.va.gov',
+   'mobile.va.gov',
10  +   'www.accesstocare.va.gov',
11  +   'www.oit.va.gov',
12  + ];
13  +
14  + function isVADomainThatOpensInNewTab(href) {
15  +   return newTabDomains.some(domain => href.toLowerCase().includes(domain));
16  + }
17  +
18  + function isNonVADomainThatOpensInSameTab(href) {
19  +   return href.toLowerCase().includes('veteranscrisisline.net');
```

⚠ Check warning on line 19 in src/site/stages/build/plugins/modify-dom/update-external-links.js

Code scanning

Incomplete URL substring sanitization

'veteranscrisisline.net' can be anywhere in the URL, and arbitrary hosts may come before or after it.

```
20  + }
21  +
22  + module.exports = {
23  +   modifyFile(fileName, file) {
24  +     let linkUpdated = false;
25  +
26  +     if (fileName.endsWith('html')) {
27  +       const { dom } = file;
28  +       dom('a[href^="http"]').each((i, el) => {
29  +         const link = dom(el);
30  +         const relAttr = link.attr('rel');
31  +         const targetAttr = link.attr('target');
32  +         const hrefAttr = link.attr('href') || '';
33  +
34  +         // We want to make sure links that open in a new tab
35  +         // always have noopener, but data-allow-opener is an
36  +         // escape hatch if we need it
37  +         if (
38  +           targetAttr === '_blank' &&
39  +           typeof link.attr('data-allow-opener') === 'undefined' &&
40  +           (!relAttr || !relAttr.includes('noopener'))
41  +         ) {
42  +           linkUpdated = true;
43  +           link.attr('rel', `noopener ${relAttr || ''}`);
44  +         }
45  +
46  +         // If a link makes it through this logic, it's an
47  +         // external link that should always open in a new tab
48  +         // There is an escape hatch here, too
49  +         if (
50  +           typeof link.attr('data-same-tab') === 'undefined' &&
51  +           !isNonVADomainThatOpensInSameTab(hrefAttr) &&
52  +           ((!hrefAttr.includes('va.gov') && !hrefAttr.includes('vets.gov')) ||
```

⚠ Check warning on line 52 in src/site/stages/build/plugins/modify-dom/update-external-links.js

Code scanning

Incomplete URL substring sanitization

'va.gov' can be anywhere in the URL, and arbitrary hosts may come before or after it.

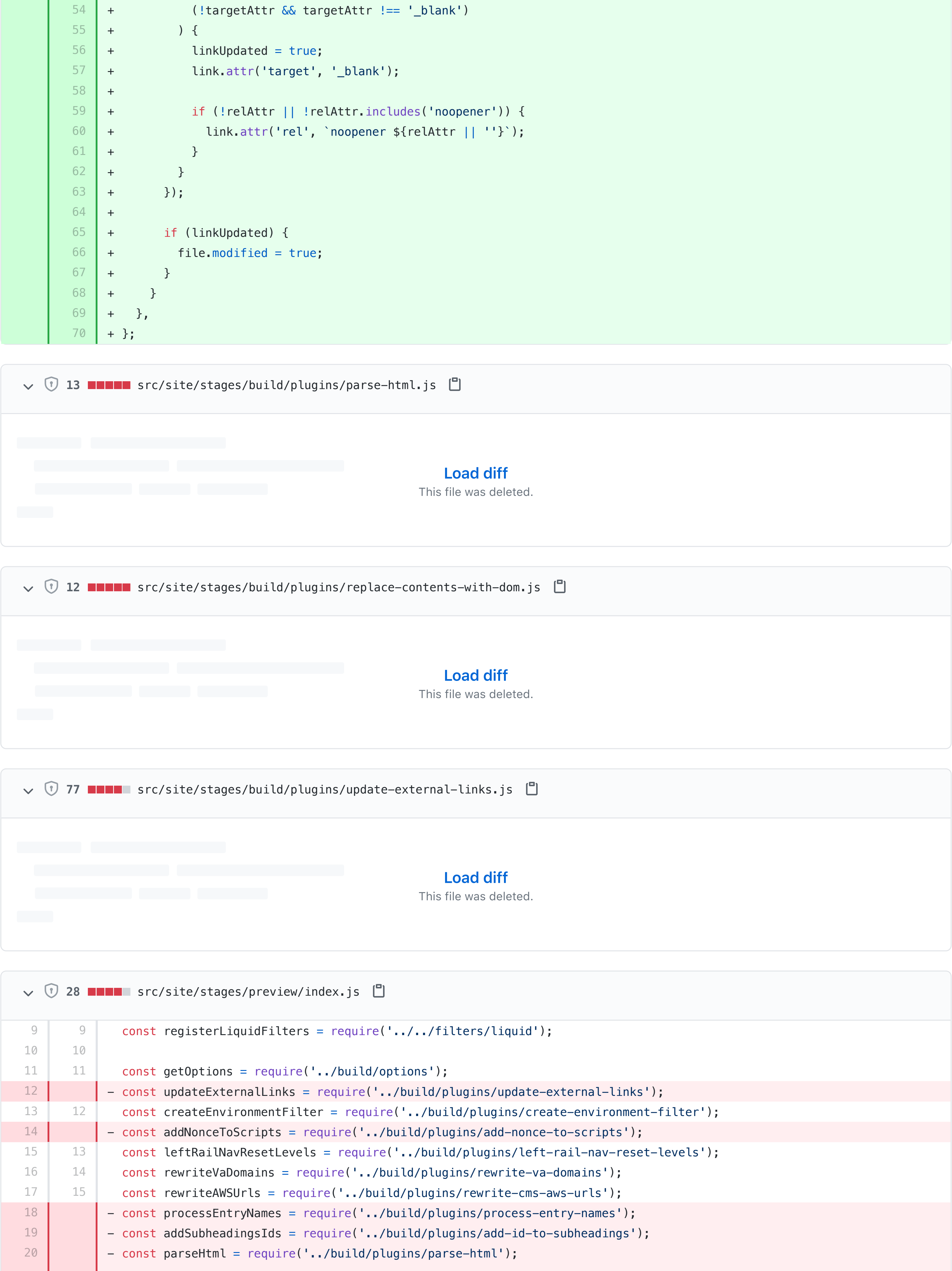
⚠ Check warning on line 52 in src/site/stages/build/plugins/modify-dom/update-external-links.js

Code scanning

Incomplete URL substring sanitization

'vets.gov' can be anywhere in the URL, and arbitrary hosts may come before or after it.

```
53  +             isVADomainThatOpensInNewTab(hrefAttr)) &&
```



```
+           (!targetAttr && targetAttr !== '_blank')
+         ) {
+           linkUpdated = true;
+           link.attr('target', '_blank');
+
+           if (!relAttr || !relAttr.includes('noopener')) {
+             link.attr('rel', `noopener ${relAttr || ''}`);
+           }
+         }
+       });
+
+       if (linkUpdated) {
+         file.modified = true;
+       }
+     }
+   },
+ };
```

13 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/parse-html.js

Load diff

This file was deleted.

12 🟥🟥🟥🟥🟥 src/site/stages/build/plugins/replace-contents-with-dom.js

Load diff

This file was deleted.

77 🟥🟥🟥🟥⬜ src/site/stages/build/plugins/update-external-links.js

Load diff

This file was deleted.

28 🟥🟥🟥🟥⬜ src/site/stages/preview/index.js

```
  const registerLiquidFilters = require('../../filters/liquid');

  const getOptions = require('../build/options');
- const updateExternalLinks = require('../build/plugins/update-external-links');
  const createEnvironmentFilter = require('../build/plugins/create-environment-filter');
- const addNonceToScripts = require('../build/plugins/add-nonce-to-scripts');
  const leftRailNavResetLevels = require('../build/plugins/left-rail-nav-reset-levels');
  const rewriteVaDomains = require('../build/plugins/rewrite-va-domains');
  const rewriteAWSUrls = require('../build/plugins/rewrite-cms-aws-urls');
- const processEntryNames = require('../build/plugins/process-entry-names');
- const addSubheadingsIds = require('../build/plugins/add-id-to-subheadings');
- const parseHtml = require('../build/plugins/parse-html');
```

```diff
- const replaceContentsWithDom = require('../build/plugins/replace-contents-with-dom');
- const injectAxeCore = require('../build/plugins/inject-axe-core');
+ const modifyDom = require('../build/plugins/modify-dom');

  async function createPipeline(options) {
    const BUILD_OPTIONS = await getOptions(options);
     */
    smith.use(rewriteVaDomains(BUILD_OPTIONS));
    smith.use(rewriteAWSUrls(BUILD_OPTIONS));
-
-   /**
-    * Parse the HTML into a JS data structure for use in later plugins.
-    * Important: Only plugins that use the parsedContent to modify the
-    * content can go between the parseHtml and outputHtml plugins. If
-    * the content is modified directly between those two plugins, any
-    * changes will be overwritten during the outputHtml step.
-    */
-   smith.use(parseHtml);
-   /*
-   Add nonce attribute with substition string to all inline script tags
-   Convert onclick event handles into nonced script tags
-   */
-   smith.use(addNonceToScripts);
-   smith.use(processEntryNames(BUILD_OPTIONS));
-   smith.use(updateExternalLinks(BUILD_OPTIONS));
-   smith.use(addSubheadingsIds(BUILD_OPTIONS));
-   smith.use(injectAxeCore(BUILD_OPTIONS));
-   smith.use(replaceContentsWithDom);
+   smith.use(modifyDom(BUILD_OPTIONS));

    return smith;
  }
```

ProTip! Use n and p to navigate between commits in a pull request.