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

👤 **ncksllvn** commented on Jan 11 • edited ▾                                    Contributor

## Description

[department-of-veterans-affairs/va.gov-team#13821](department-of-veterans-affairs/va.gov-team#13821)

The big memory hog during our build is when we parse every single HTML file into a DOM using Cheerio and store the entire result in memory at once. This PR modifies that by parsing each DOM one at a time, doing the next build steps that require the DOM, then immediately tearing down the DOM so that we only ever have one DOM tree in memory at once. I had to refactor the build steps (those dependent on `file.dom` ) to accept just a single `file` at a time to make this work.

It appears we can save several gigs of memory and shave off almost 30 seconds doing this.

▼ Build output w/these changes

```
                                          Step summary
    Step              Description                  Time Elapsed   Memory Used This Step   Total Memory Used After
    Step

       0    Preserving Webpack build output         1767ms         3.7mB                   33.29mB
       1    Create React pages                      29ms           0.17mB                  33.48mB
       2    Get Drupal content                      11518ms        333.17mB                366.67mB
       3    Add Drupal Prefix                       19ms           0.82mB                  367.52mB
       4    Create Outreach Assets Data             8ms            0.55mB                  368.08mB
       5    Create "Resources and support" section of the website   104ms   29.86mB       397.97mB
       6    Create environment filter               1ms            0.13mB                  398.12mB
       7    Add filenames for debugging             1ms            0.39mB                  398.53mB
       8    Check collections                       5ms            0.23mB                  398.77mB
       9    Group collections                       860ms          4.36mB                  403.15mB
      10    Reset left rail navigation menu levels  3ms            0.54mB                  403.7mB
      11    Add the date to filenames               29ms           7.28mB                  411mB
      12    Add app assets                          95ms           −3.82mB                 407.2mB
      13    Add content assets                      53ms           −34.55mB                372.66mB
      14    Plug the content into the templates     116ms          5.15mB                  377.82mB
      15    Translate the markdown to html          18ms           1.39mB                  379.23mB
      16    Add permalinks and change foo.md to foo/index.html   734ms   −0.5mB            378.74mB
      17    Create header and footer                11ms           1.54mB                  380.29mB
      18    Generate navigation                     264ms          6.59mB                  386.89mB
      19    Apply layouts                           24142ms        198.07mB                584.97mB
      20    Rewrite VA domains for the buildtype    2455ms         −179.65mB               405.37mB
      21    Rewrite Drupal pages                    0ms            0.01mB                  405.43mB
      22    Create Drupal debug page                2ms            0.69mB                  406.13mB
      23    Download Drupal assets                  2ms            0.4mB                   406.55mB
      24    Read application assets from disk        558ms         −28.38mB                378.19mB
      25    Create sitemap                          130ms          1.77mB                  379.98mB
      26    Update robots.txt                       0ms            0.01mB                  380.01mB
      27    Check for CMS URLs                      1318ms         40.47mB                 420.49mB
      28    Parse HTML files                        42223ms        399.25mB                819.76mB
   Build finished!
   ✨  Done in 91.32s.
```

▼ Without these changes

```
                                      Step summary
      Step             Description                     Time Elapsed   Memory Used This Step   Total Memory
   Used After Step

        0    Preserving Webpack build output           1962ms         3.22mB                  39.66mB
        1    Create React pages                        10ms           0.16mB                  39.84mB
        2    Get Drupal content                        11080ms        327.19mB                367.04mB
        3    Add Drupal Prefix                         9ms            2.77mB                  369.84mB
        4    Create Outreach Assets Data               7ms            0.55mB                  370.4mB
        5    Create "Resources and support" section of the website  110ms  4.67mB            375.09mB
        6    Create environment filter                 1ms            0.13mB                  375.24mB
        7    Add filenames for debugging               1ms            0.39mB                  375.65mB
        8    Check collections                         5ms            0.23mB                  375.9mB
        9    Group collections                         816ms          9.08mB                  384.99mB
       10    Reset left rail navigation menu levels    3ms            0.73mB                  385.73mB
       11    Add the date to filenames                 27ms           -5.97mB                 379.78mB
       12    Add app assets                            101ms          -1.69mB                 378.11mB
       13    Add content assets                        102ms          -7.48mB                 370.64mB
       14    Plug the content into the templates       131ms          5.21mB                  375.88mB
       15    Translate the markdown to html            20ms           1.39mB                  377.29mB
       16    Add permalinks and change foo.md to foo/index.html  733ms  2.64mB               379.94mB
       17    Create header and footer                  13ms           1.54mB                  381.49mB
       18    Generate navigation                       281ms          6.03mB                  387.54mB
       19    Apply layouts                             22272ms        154.41mB                541.96mB
       20    Rewrite VA domains for the buildtype      2540ms         -130.1mB                411.92mB
       21    Rewrite Drupal pages                      0ms            0.01mB                  411.98mB
       22    Create Drupal debug page                  2ms            0.69mB                  412.69mB
       23    Download Drupal assets                    2ms            0.41mB                  413.11mB
       24    Read application assets from disk         584ms          -34.69mB                378.43mB
       25    Create sitemap                            101ms          2.6mB                   381.05mB
       26    Update robots.txt                         0ms            0.01mB                  381.07mB
       27    Check for CMS URLs                        1324ms         40.99mB                 422.08mB
       28    Parse HTML files                          51322ms        5193.87mB               5615.95mB
       29    Add nonce to script tags                  1896ms         101.1mB                 5717.08mB
       30    Process [data-entry-name] attributes into Webpack asset paths  546ms  9.83mB    5726.92mB
       31    Update external links                     449ms          17.56mB                 5744.49mB
       32    Add IDs to subheadings                    666ms          6.24mB                  5750.75mB
       33    Check for broken links                    5676ms         504.01mB                6254.77mB
       34    Inject axe-core for accessibility         17ms           0.88mB                  6255.67mB
       35    Save the changes from the modified DOM    5612ms         -4871.23mB              1384.46mB
  Build finished!
  ✨  Done in 118.86s.
```

## Testing done

⚠️ Needs a lot more testing! Todos -

☑ Make preview server's build align with these changes

☑ The preview server (make sure the Axe check still works)

☑ Team site Webpack assets (make sure the hash-free bundles are still being copied over correctly)
   - I confirmed this by checking the outputted `generated/` directory of a vagovprod build to confirm `proxy-rewrite.js` and other TeamSite-required files are there

☑ Error handling - make sure broken links still fail the build
   - Would like to double confirm this

☑ Confirm each build step still works

## Screenshots

I don't think there is any screenshot to post. I pasted the terminal output above.

## Acceptance criteria

☑ No more memory issues during build

## Definition of done

☑ Events are logged appropriately

☑ Documentation has been updated, if applicable

☐ A link has been provided to the originating GitHub issue (or connected to it via ZenHub)

☑ No sensitive information (i.e. PII/credentials/internal URLs/etc.) is captured in logging, hardcoded, or specs

❤️ 3

-○- Attempt to fix memory issue     ✕ 9f75e29

👁 **va-vfs-bot** requested a review from **department-of-veterans-affairs/frontend-review-group** on Jan 11

👁 **va-vfs-bot** reviewed on Jan 11

View changes

**va-vfs-bot** left a comment

# ESLint is disabled

`vets-website` uses ESLint to help enforce code quality. In most situations we would like ESLint to remain enabled.

## What you can do

See if the code can be refactored to avoid disabling ESLint, or wait for a VSP review.

| `src/site/stages/build/plugins/modify-dom/check-broken-links/index.js` | ⬍ Show resolved |
|---|---|

| `src/site/stages/build/plugins/modify-dom/parse-html.js` Outdated | ⬍ Show resolved |
|---|---|

✎ **ncksllvn** changed the title ~~Attempt to fix memory issue~~ Attempt to fix memory issue during content-build on Jan 11

-○- lint     ✕ b08e839

👁 **va-vfs-bot** reviewed on Jan 11

View changes

**va-vfs-bot** left a comment

# Icon found

Icons can be decorative, but sometimes they are used to convey meaning. If there are any semantics associated with an icon, those semantics should also be conveyed to a screen reader.

## What you can do

Review the markup and see if the icon provides information that isn't represented textually, or wait for a VSP review.

| `src/site/stages/build/plugins/modify-dom/add-id-to-subheadings.js` | ⬍ Show resolved |
|---|---|

**va-vfs-bot** reviewed on Jan 11

View changes

**va-vfs-bot** left a comment

# ESLint is disabled

`vets-website` uses ESLint to help enforce code quality. In most situations we would like ESLint to remain enabled.

## What you can do

See if the code can be refactored to avoid disabling ESLint, or wait for a VSP review.

| | |
|---|---|
| `src/site/stages/build/plugins/modify-dom/add-id-to-subheadings.js` | ↕ Show resolved |
| `src/site/stages/build/plugins/modify-dom/add-id-to-subheadings.js` `Outdated` | ↕ Show resolved |
| `src/site/stages/build/plugins/modify-dom/add-id-to-subheadings.js` `Outdated` | ↕ Show resolved |
| `src/site/stages/build/plugins/modify-dom/inject-axe-core.js` | ↕ Show resolved |
| `src/site/stages/build/plugins/modify-dom/update-external-links.js` | ↕ Show resolved |

✓ **mpelzsherman** approved these changes on Jan 11

View changes

**mpelzsherman** left a comment                                    `Contributor`

I tested this on my machine and it worked great! Thank you **@ncksllvn**!! I hope we are able to merge this soon!

❤️ 1

**mpelzsherman** mentioned this pull request on Jan 11

**Expand VAMC tests** #15543

⑂ Merged

**ncksllvn** added 4 commits on Jan 11

○ 🖼 `update the preview script to align with changes`                        ✕ 50a659c

○ 🖼 `remove unused file`                                                        24e3120

○ 🖼 `rm unused lint disables`                                                   519599f

○ 🖼 `rename plugin from parseHtml`                                           ✓ 0a32515

👁 **va-vfs-bot** reviewed on Jan 11

View changes

**va-vfs-bot** left a comment

# ESLint is disabled

`vets-website` uses ESLint to help enforce code quality. In most situations we would like ESLint to remain enabled.

## What you can do

See if the code can be refactored to avoid disabling ESLint, or wait for a VSP review.

| `src/site/stages/build/plugins/modify-dom/index.js` | ⇕ Show resolved |
|---|---|

👁 **ncksllvn** reviewed on Jan 11

[View changes]

`src/site/stages/build/plugins/modify-dom/index.js`  ( Outdated )

```
44  +    }
45  + };
46  +
47  + module.exports = parseHtml;
```

**ncksllvn** on Jan 11  ( Author )  ( Contributor )

This whole plugin is the key change. Everything listed in the `domModifiers` list use to be a separate Metalsmith plugin. This PR converts them into sub-steps within this plugin where each sub-step instead operates on a single file at a time, so that only one `file.dom` exists in memory at once.

👁 **ncksllvn** marked this pull request as ready for review on Jan 11

✏️ **ncksllvn** changed the title ~~Attempt to fix memory issue during content-build~~ Reduce memory consumption during content-build on Jan 11

🚀 **va-vfs-bot** had a problem deploying to **master/nps--fix-memory-issue** on Jan 11 ( Error )

🚀 **va-vfs-bot** temporarily deployed to master/nps--fix-memory-issue on Jan 11 ( Inactive )

**meganhkelley** commented on Jan 12

Hey @ncksllvn I notice this box still unchecked above: " Team site Webpack assets (make sure the hash-free bundles are still being copied over correctly)" — is that something you are planning to work on, or are you dependent on another team to complete that check?

**ncksllvn** commented on Jan 12    ( Contributor )  ( Author )

Good question @meganhkelley - I'm going to confirm that this morning then I'll check the box. It isn't much work.

👍 1

⊶ `Move copyAssetsToTeamSitePaths into the conclude hook`                        ✕ f56d50b

👁 **ncksllvn** reviewed on Jan 12

**src/site/stages/build/plugins/modify-dom/process-entry-names.js**

| 112 | 110 | | `if (!buildOptions.isPreviewServer) {` |
| 113 | | - | `    copyAssetsToTeamSitePaths(buildOptions, files, entryNamesDictionary);` |
| | 111 | + | `    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 only doing that in the webpack config: https://github.com/department-of-veterans-affairs/vets-website/blob/a012bad17e5bf024b0ea7326a72ae6a737e349ec/src/site/stages/build/plugins/process-entry-names.js#L35` |

.

**ncksllvn** on Jan 12    Author    Contributor

cc @timwright12 @cvalarida

**ncksllvn** commented on Jan 12    Contributor    Author

@meganhkelley This is ready for review 😎

Update/add some comments    ✓ a7d0081

**ncksllvn** reviewed on Jan 12

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

| 103 | + | `      }` |
| 104 | + | `    }` |
| 105 | + | `  },` |
| 106 | + | `};` |

**ncksllvn** 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

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

| 48 | + | `      console.log(errorOutput);` |
| 49 | + | `    }` |
| 50 | + | `  },` |
| 51 | + | `};` |

**ncksllvn** on Jan 12    Author    Contributor

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

**ncksllvn** mentioned this pull request on Jan 12

**Improve memory usage during the content build** department-of-veterans-affairs/va.gov-team#13821

⊙ Open

📋 0 of 3 tasks complete ▱▱▱▱▱

---

🖼 **ncksllvn** commented on Jan 12                                    `Contributor`  `Author`

@meganhkelley FYSA that the GH issue this PR addresses is here, department-of-veterans-affairs/va.gov-team#13821.

---

🚀 🖼 **va-vfs-bot** deployed to **master/nps--fix-memory-issue** on Jan 12

[ View deployment ]

---

✅ **timwright12** approved these changes on Jan 13

[ View changes ]

---

🖼 **timwright12** left a comment                                              `Contributor`

Tested locally, all sets appear to be running as before and the site seems to render correctly.

---

🖼 **ncksllvn** merged commit **07edee3** into `master` on Jan 13              [ View details ]

5 checks passed

---

⎇ 🖼 **ncksllvn** deleted the `nps--fix-memory-issue` branch on Jan 13

🔗 🖼 **acrollet** mentioned this pull request on Jan 13

**Update va-gov/web (v0.1.1217 => v0.1.1227)** department-of-veterans-affairs/va.gov-cms#4004

⎇ Merged

🔗 🖼 **ncksllvn** mentioned this pull request 9 days ago

**[Jenkinsfile] Decrement Drupal parallelization to 10 if not a content deployment** #16275

⎇ Merged

📋 0 of 5 tasks complete ▱▱▱▱▱