



The current FE does not have a diff viewer, which leads to a degraded experience as users lose features which they've had on github. This PR adds this back.

Closes IS-54

#### Solution

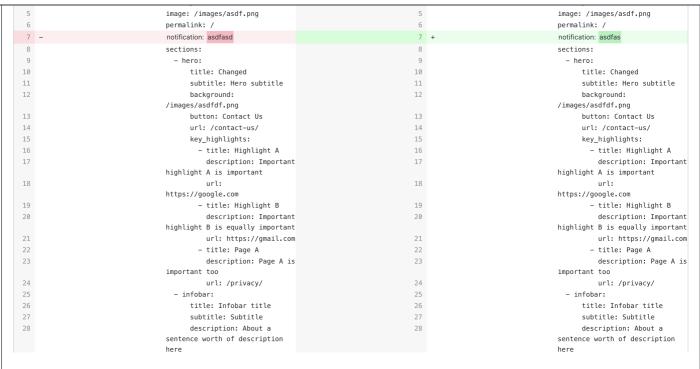
- 1. use ReactDiffViewer package to render the diff view
  - o i compared between a few packages and i chose this because this was the most feature complete + ts
  - this package uses classnames under the hood to style and it renders the whole view as a table. This causes difficulty with styling because tables have different css rules compared to normal. Some implementation details will be outlined below
  - o whole view is a table; lines (a tr) consists of a gutter (spacing + line no), marker (type of change), together with content, each of which is a td (this is x2 since we render a split view)
  - code folds also have the same structure except that content is only rendered in a single cell. This leads to wonky formatting as tables depend on cells to render. No good solution for now see below for pics.
- 2. use hook to query the backend
  - the structure of the underlying api call isn't ideal because the change is tied to a rr. this isn't ideal because the changes should be tied to a tuple of (commit hash, commit hash) and it might cause trouble in the future. probably will change this in a small PR next to keep changes focused.
  - passing a query param rather than being able to do /review/<fileName>. this is a workaround becaues filenames might be nested within directories (eg: src/api.js) and express matches on /review/src .\_. not sure if this is possible in express but the github API uses hte /review/<filename> form

## Screenshots

2 way code fold



note the expanded right hand gutter, this happens iff the code fold is at the very top of the table because then the td will use the code fold



#### lower code fold only



note how the gutters are normal sized now

full expansion renders fine

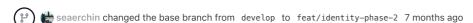
#### Tests

- navigate to a email login based repo
- create a new image
- create a new file
- create a new unlinked page
- create a new page in folder
- create a new page in a subfolder
- create a new resource room page
- repeat the above steps but edit existing items rather than creating
- request a review
- pages should be able to be compared, files/images should not
- resource room page should have human readable name

## notes

1. this is changed from the one in figma - the header is not the compare changes for <filename> . instead, this is displayed below. this is because to achieve the former, we'd need Context for this use case and trigger a state change, which seems tedious. will double check with @NatMaeTan if this is acceptable; otherwise, will do the header









b seaerchin had a problem deploying to staging 7 months ago 🕞 Failure



🚼 seaerchin requested review from dcshzj, alexanderleegs, kishore03109 and prestonlimlianjie 7 months ago



🖢 seaerchin had a problem deploying to staging 7 months ago (Failure)



alexanderleegs approved these changes on Nov 21, 2022

View reviewed changes



alexanderleegs left a comment

Minor nit in the comments!

Apart from that, the diff viewer currently doesn't render line numbers properly if the screen width is too small. As discussed offline, this is due to the library being used making use of a table under the hood, which is harder to modify - we're opting to leave it as is for now.

Example Title 22.md		
1	1	
2 title: Example Title 22	2 title: Example Title 22	
3 – permalink: / permalink /	3 + permalink: / testing /	
4 description: ""	4 description: ""	
5	5	
	6 + <style></th></tr><tr><th></th><td>7 + td, th {</td></tr><tr><th></th><td><pre>8 + border: none!important;</pre></td></tr><tr><th></th><th>9 + }</th></tr><tr><th></th><th>1 + </style>	
	0	
	1 +	
	1	
	1 +   Column 1   Column 2   Column 3	
	2	
	1 +	
	3	
	1 +         4	
	1 +	
	5	
	1 +	
	6	

Also, could we also track the portions which are WIP in an issue? I'm assuming that the edit and PR buttons in the overview, and the link to the direct staging site will be handled in a separate PR



```
src/layouts/ReviewRequest/components/RequestOverview/RequestOverview.tsx Outdated
```

```
197
     + }: {
198
         onClick: (name: string, filePath: string[]) => void
      + } & RequestOverviewProps) => {
```



alexanderleegs on Nov 21, 2022

Nit: could we rename this to be more informative, since we'll be adding functionality for the other 2 buttons as well in future?





seaerchin on Dec 12, 2022

wrt this - actually the other 2 are links, so they won't take an onClick





Reply...

Resolve conversation

#### view reviewed changes

src/layouts/ReviewRequest/components/DiffView.tsx

28 // make it look as though there's an extra space inserted 29 // in (see: `b` vs `boo`) 30 compareMethod={DiffMethod.WORDS} 31 styles={{



# dcshzj on Nov 29, 2022

Is it possible to specify that the diff renders using a monospaced font?





#### seaerchin on Dec 12, 2022

can i double check wym by this





### 🌉 dcshzj on Dec 15, 2022

Currently, the diff is rendered in a normal font if it is the same for the old and new value. I was thinking we should render the whole view in a monospace font similar to how GitHub renders the diff view.





#### seaerchin on Dec 15, 2022

oh ok gotcha - will have to look into this; thanks for raising!





Reply..

#### Resolve conversation

src/layouts/ReviewRequest/components/DiffView.tsx

```
oldValue={oldValue}
24
            newValue={newValue}
```



# dcshzj on Nov 29, 2022

Sanity check: How does this diff viewer handle media files (i.e. images)?





## seaerchin on Dec 12, 2022

we shouldn't handle it here - i was thinking in the DTO which we return from the backend, we specify the filetype (page/file/extlink/image), and we allow comparisons only for pages.





#### dcshzj on Dec 15, 2022

Yup I think that works too, or we can bypass the diff viewer and render the image as-is, so at least the user can see how the image looks like previously and now





### seaerchin on Dec 15, 2022

actually rendering the image as is might not be that simple just because the image might be extremely huge (eg for their homepage); i'm not sure if there's a good solution for this beyond just showing a link to the raw image maybe? wdyt about this





### dcshzj on Dec 15, 2022

I was thinking if maybe we can set a max height/width for the image but showing a link should be ok also hmm





# seaerchin on Dec 15, 2022

actually that wouldn't work also - we would need to determine our resizing strategy; whether we want to crop, preserve aspect ratio but then maybe have whitespace or like ignore the aspect ratio but then resize (and it might look weird in that case)





dcshzj on Dec 15, 2022



Reply..

#### Resolve conversation

src/layouts/ReviewRequest/components/DiffView.tsx

```
<Heading
113
                  as="h2"
114
                  textStyle="h2"
                >{`Compare changes for '${fileName}'`}</Heading>
115
```



dcshzj on Nov 29, 2022 • edited ▼

3 things:

- 1. Should truncation be applied here?
- 2. Is this filename or should it be the review request title? Not sure if the Figma is wrong here Clarified offline, this should be the page title, not the file name
- 3. I think the code indentation might need to be fixed ( </Heading> should be on its own line, etc)





#### seaerchin on Dec 12, 2022

ok so i just realised why i skipped point 2 - this is because the title exists only in the frontMatter. For normal pages, this isn't a big deal but  $for \ resource \ pages, \ they \ are \ prefixed \ by \ \ yyyy-mm-dd \ and \ \ type \ , \ which \ we \ need \ additional \ metadata \ to \ determine...$ 

i can do this, but i feel it's out of scope of this PR and i'd be relying on the new PageService from our BE; wdyt about me filing this as a BE issue?





seaerchin on Dec 12, 2022

re: truncation, yes - will add this in!





# dcshzj on Dec 15, 2022

Yup I think this is a BE issue, but I think it should still be the page title as per Figma? So even for resource room pages, it still renders the page title but excludes the date.





# seaerchin on Dec 15, 2022

yea, it's backend so what's happening now is that i'm going to add both the item type together with the title (not going to happen on the frontend) but it does take alot of github calls (1 per file so we have to hope that the users don't have much changes)





Reply...

### Resolve conversation

src/layouts/ReviewRequest/components/RequestOverview.tsx

```
@@ -165,6 +167,36 @@ export const RequestOverview = ({
165
       167
                 items,
166
       168
                 allowEditing,
       169
               }: RequestOverviewProps): JSX.Element => {
                 const [fileName, setFileName] = useState(""]
```



# dcshzj on Nov 29, 2022 • edited 🕶

Curious: Any reason for using state for the file name rather than relying on query params? I was thinking if it might be better if we allow users to use the URL to directly enter the diff view of a specific file.

Edit: Clarified offline, would be ideal to use query params!

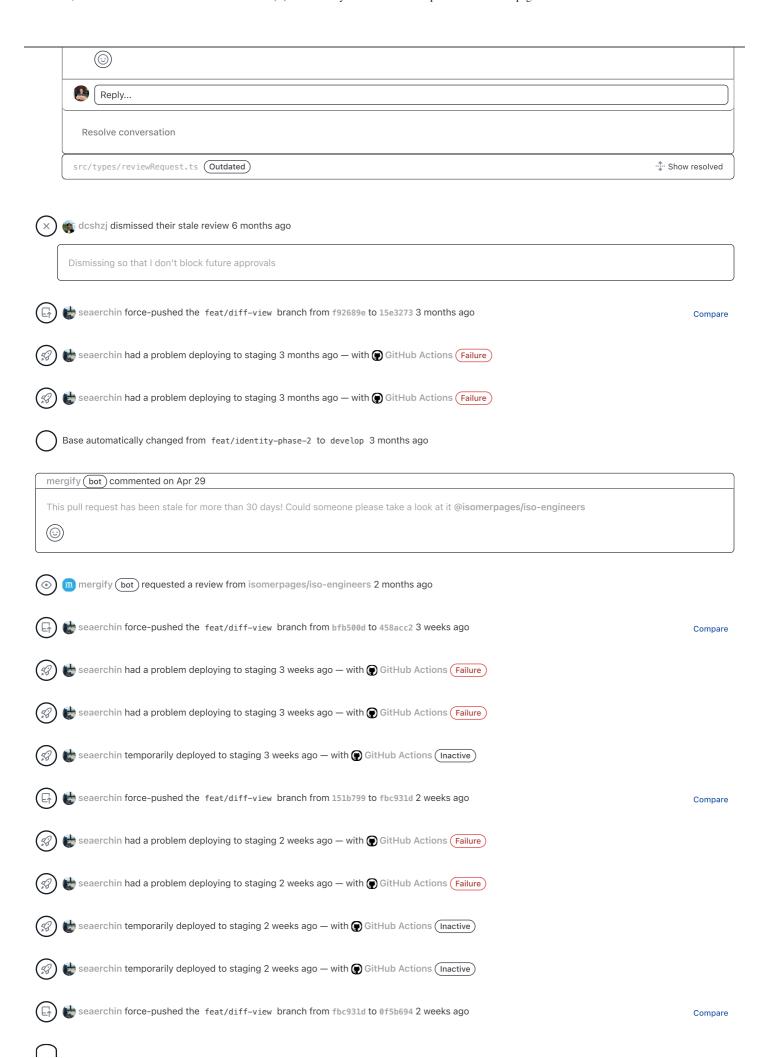


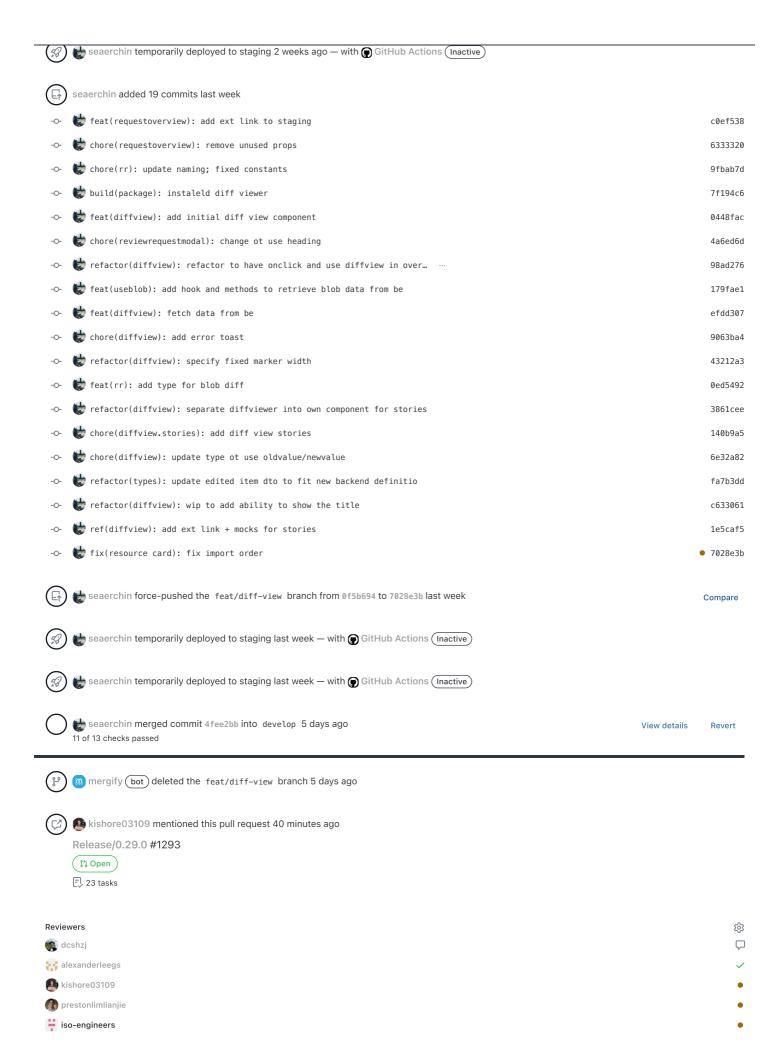


# seaerchin on Dec 12, 2022

i might do the query params in a follow up PR to avoid bloating this PR - is that ok w/ you @dcshzj







Labels None yet	· · · · · · · · · · · · · · · · · · ·
None yet	
Projects	\$
None yet	
Milestone	· · · · · · · · · · · · · · · · · · ·
No milestone	
Development	\$
Successfully marging this pull request may close those issues	

Successfully merging this pull request may close these issues

None yet

