Code Inspection Checklist

1.Java & Typescript

General Review

Checklist

Reviewer: Cameron Beattie
Reviewed File: FileDropzone.tsx, FileDropzone.test.tsx

1. Variable Declaration

- ☑ Are variable names informative
  - I am not positive what pstyle is for but the developer added a comment to explain it
  - Otherwise everything is clear
- ☑ Are variable names unique (not confusing or similar)?
  - variable names are unique
- ☑ Are variable names following chosen capitalization conventions (camel case)?
  - Variable names are camel case
- ☑ Are variables properly initialized?
  - Constant variables are declared with const
- ☑ Are variables labelled as private or public based on their use?
  - Not necessary
- ☑ Is every declared variable used?
  - There are no unused variables
- ☑ Is there excessive use of unnecessary temporary variables?
  - there are no temporary variables

2. Methods and method signatures including return and input types

- ☑ Do method names reflect method functionalities?
  - Method names are clear as to what they do
- ☑ Do method expected return values match the intended use of the return value?

    - There are no methods with expected return values
- ☑ Do methods have safeguards for problematic/unexpected input?
    - The useDropzone method anly accepts json and xml type files
- ☑ Is there a high cohesion between the methods within the same class?
    - NA

3. Control flow Defects

- ☑ Are Switch cases used instead of if/else blocks when appropriate?
    - NA
- ☑ Are While loops successfully terminated to avoid infinite loops?
    - NA
- ☑ Are control flows used efficiently in the handling of erroneous input?
    - NA

- ☑ Are loop variables declared properly so that their scopes are only as big as necessary?
    - There are no loops
- ☑ Are there checks for edge cases (out of bounds) for For loops?
    - NA
- ☑ Are there else blocks used for every if condition to ensure no case goes unhandled?
    - There is a case in onDrop that does not have an else clause, but it is not necessary because the expected behavior is to do nothing

4. Code style & practices

- ☑ Is code consistently indented, spaced, and formatted?
    - It is
- ☑ Code is well documented using inline comments and docstrings.
    - There is one comment, Docstrings are lacking significantly
- ☑ Are Expensive operations minimized (shallow object copies replacing deep ones if possible)
    - NA
- ☑ Are generics used where possible to improve code readability & reduce complexity?
    - NA

Typescript Review Checklist

- ☑ Are type annotations and inference used?
    - No variables are typed
- ☑ Are strict Null checks in place?
    - NA

Use type inference, type annotation, and generics.

Rollbar Review Checklist

- ☑ Is Rollbar being used consistently?
    - No, I will add the necessary logging

2. Front End (TS & ReactJS)

- ☑ Is the single responsibility principle applied to react components?
    - this component is only for the Dropzone
- ☑ Are container components used strictly for managing state and business logic?
    - NA
- ☑ Are presentational components used for UI rendering and logic strictly?
    - Yes
- ☑ Are related components, styles, and assets grouped within same directory?
    - Yes
- ☑ Are functional components used instead of class components if possible?
    - NA
- ☑ Are React hooks used to manage & control state & effects in functional components?
    - Yes
- ☑ Are hooks called at the top level of functional components?
    - Yes
- ☑ Are unnecessary re-renders avoided?
    - Yes
- ☑ Are local component states prioritized for UI-specific state matters?
    - NA

General Additional Notes:

Overall, this page is well implemented but somewhat difficult to understand in various portions. I believe some docstrings for methods would greatly improve it's readability.

Test File

Coverage

NA

General Notes

- I think that this component was well unit-tested. It checks UI renders and provides mock data to ensure the component behaves correctly. There aren't many independent functions to unit test (it is mostly setting component parameters) so it's difficult to say the test coverage