

Safari Print Button Issue in EDocman PDF Modal – Suggested Fix

Issue Summary

In Safari (tested on macOS), when the user clicks the "Print" button in the EDocman PDF modal viewer, the expected print dialog does not appear. Instead, a popup alert is shown, and the PDF file is downloaded. This issue does not occur in Chrome or Firefox.

Cause

Safari only allows popups or new tabs (`window.open`) when triggered directly by a user action.

In the current implementation, `window.open` is executed after the PDF file is loaded via `fetch().then(...)`, which Safari interprets as non-user-initiated and thus blocks it.

Proposed Solution

To ensure compatibility with Safari, modify the logic as follows:

1. Open a new tab immediately when the user clicks the Print button.
2. Load the PDF file asynchronously via `fetch()`.
3. Once loaded, assign the PDF to the opened tab using a blob URL.
4. Then call the `print()` method inside that tab.

Example Code

```
function printPDF() {
  const printWindow = window.open("", '_blank');
  if (!printWindow) {
    alert('Popup blocked. Please allow popups for this site.');
```

```
    return;
  }

  printWindow.document.write('<p>Loading PDF...</p>');

  fetch(pdfUrl)
    .then(response => response.blob())
    .then(blob => {
      const blobUrl = URL.createObjectURL(blob);
      printWindow.location.href = blobUrl;

      printWindow.onload = function () {
        setTimeout(() => {
          printWindow.print();
          setTimeout(() => URL.revokeObjectURL(blobUrl), 1000);
        }, 1000);
      };
    })
    .catch(error => {
      console.error('Print failed:', error);
      printWindow.document.write('<p>Failed to load PDF.</p>');
    });
}
```

Remarks

- This approach improves Safari compatibility without affecting Chrome or Firefox.
- Since the change is isolated to the print action, the risk to other features is minimal.
- It may be considered a fallback specific to Safari behavior, and can be applied conditionally if needed.