

HTML Cleanup Fix Summary

HTML Cleanup Functionality Restored

Issue Found:

The `cleanup_html: true` setting in JSON wasn't working because:

- Missing `cleanupIntermediateHtmlFiles()` method
- No call to cleanup function in main generation loop
- Cleanup logic was completely absent

Solution Applied:

1. Added Cleanup Method

```
private function cleanupIntermediateHtmlFiles(): void
{
    // Scans output directory for HTML files
    // Removes all HTML files except index.html and README.md
    // Provides detailed logging of what was removed/kept
}
```

2. Integrated Cleanup Call

```
// Clean up intermediate HTML files if requested and PDFs were generated
if ($this->config['conversion']['cleanup_html'] && $this->config['output']['generate_pdf']) {
    $this->cleanupIntermediateHtmlFiles();
}
```

3. Enhanced Logging

- Shows count of files cleaned up
- Lists first 5 removed files by name
- Shows which files were kept (index.html, README.md)
- Only runs when both cleanup is enabled AND PDFs are generated

⚡ Configuration:

```
{  
  "conversion": {  
    "cleanup_html": true // Now works correctly  
  }  
}
```

Test Results:

Cleanup enabled: Removes intermediate HTML files

Selective cleanup: Keeps index.html and README.md

Detailed logging: Shows exactly what was removed

Conditional execution: Only runs when PDFs are generated

Safe operation: Won't delete important files

Usage:

1. Set "cleanup_html": true in config/docs-pdf.json
2. Run php scripts/generate-docs-pdf-simple.php --verbose
3. Watch for cleanup messages in output
4. Check that only essential HTML files remain

Expected Result:

Cleaning up intermediate HTML files...

- trash Removed: file1.html
- trash Removed: file2.html
- trash Removed: file3.html

Cleaned up 3 intermediate HTML files

- keep Kept: index.html, README.md

The HTML cleanup functionality now works exactly as intended!