

Productionization Walkthrough — ag-docs-to-pdf

Summary

Productionized the VS Code extension with improved architecture, type safety, performance, and a comprehensive test suite. **All 59 tests pass.**

Changes Made

1. Code Quality & Architecture

| File | Change |
|-------------------------------------|--|
| TS <code>markdownRenderer.ts</code> | Eliminated all <code>any</code> casts and <code>@ts-ignore</code> — now uses <code>MarkdownIt.Token</code> and <code>MarkdownIt.Renderer.RenderRule</code> from CJS-compatible namespace types |
| TS <code>PdfGenerator.ts</code> | Extracted <code>buildHeaderTemplate()</code> and <code>buildFooterTemplate()</code> functions; added optional <code>assetsDir</code> for testability |
| TS <code>browserFinder.ts</code> | Added <code>customPath</code> parameter, Brave browser support, and platform-specific error messages |
| { } <code>package.json</code> | Added <code>antigravity.chromePath</code> configuration and <code>pretest</code> / <code>test</code> scripts |

2. Performance

| File | Change |
|---------------------------------|---|
| TS <code>browserPool.ts</code> | New — singleton <code>BrowserPool</code> with lazy launch and promise-lock for concurrent safety |
| TS <code>PdfGenerator.ts</code> | Accepts <code>BrowserPool</code> — closes pages, not browsers. One Chromium instance shared across all exports |
| TS <code>extension.ts</code> | Pool lifecycle in <code>activate()</code> / <code>deactivate()</code> , removed noisy startup toast, fixed progress reporting |

3. Testing Infrastructure

| File | Purpose |
|--|--|
| <code>{ } tsconfig.test.json</code> | Separate TypeScript config for tests (outputs to <code>out/</code>) |
| <code>TS markdownRenderer.test.ts</code> | 30 unit tests: alerts (5 types), mermaid, checkboxes, diff blocks, carousel, file links, code blocks |
| <code>TS browserFinder.test.ts</code> | 6 tests: custom path, auto-detection, error messages |
| <code>TS pdfGenerator.test.ts</code> | 8 integration tests: PDF generation, nested dirs, empty input, browser reuse |

Test Results

`browserFinder`

- ✓ returns a custom path when it exists
- ✓ ignores custom path when it does not exist
- ✓ ignores undefined custom path
- ✓ ignores empty string custom path
- ✓ error message includes install instructions
- ✓ finds a browser on the current system

`markdownRenderer`

| | |
|----------------------------------|------------|
| Basic Rendering (11 tests) | ✓ all pass |
| Alerts Plugin (8 tests) | ✓ all pass |
| Mermaid Plugin (3 tests) | ✓ all pass |
| Checkbox Plugin (5 tests) | ✓ all pass |
| Diff Blocks (4 tests) | ✓ all pass |
| Carousel Plugin (2 tests) | ✓ all pass |
| File Link Icons Plugin (5 tests) | ✓ all pass |
| Code Blocks (3 tests) | ✓ all pass |
| createRenderer (3 tests) | ✓ all pass |

`PdfGenerator`

- ✓ generates a PDF from simple markdown
- ✓ generates a PDF with code blocks
- ✓ generates a PDF with alerts
- ✓ generates a PDF with checkboxes
- ✓ creates output directory if it does not exist
- ✓ uses title from H1 heading
- ✓ handles empty markdown gracefully
- ✓ reuses browser across multiple generate calls

59 passing (45s)

Build Verification

| Step | Command | Result |
|------------------|--|--|
| Type check | <code>tsc --noEmit</code> | <input checked="" type="checkbox"/> Zero errors |
| Test compile | <code>tsc -p tsconfig.test.json</code> | <input checked="" type="checkbox"/> Zero errors |
| Production build | <code>npm run build</code> | <input checked="" type="checkbox"/> dist/ bundle created |
| Test suite | <code>npm test</code> | <input checked="" type="checkbox"/> 59/59 passing |

Remaining Item

- **Large file handling** (`content-length guards`) was noted in the plan but not prioritized — consider adding in a follow-up if large markdown files cause memory issues.