

DiffService

September 4, 2026

DiffService

Kind: Service

Source: `atlوريا-monorepo/apps/api/src/change-request/diff.service.ts`

DiffService — the CR diff engine over the per-project docs git substrate (S2.2).

Two capabilities:

- `diffRefs`: a structured two-ref diff (per-file status, +/- counts, unified hunks with old/new line numbers, and heading-level `SectionDiffs` so the UI can say "\$Setup changed").
- `threeWaySections`: a pure, heading-aligned three-way section classifier for CR merge (base vs mine vs theirs → unchanged / take-theirs / keep-mine / conflict).

ALL git invocation is via promisified `execFile('git', [argv])` — never a shell string — mirroring `DocsRepoService`. Renames are reported as delete+add for v1 (`--no-renames`).

`DiffService` is the change-request diff engine for a project's documentation Git repository. It produces structured two-ref diffs with file statuses, line counts, hunks, and heading-level section changes, and it classifies three-way section merges as unchanged, take-theirs, keep-mine, or conflict. Git commands are executed safely through `execFile('git', argv)` rather than shell command strings.

Methods

Method	Signature	Returns	Description
<code>diffRefs</code>	<code>diffRefs(projectId: string, baseRef: string, headRef: string, paths: string[])</code>	<code>Promise<CrDiff></code>	Structured diff of <code>baseRef..headRef</code> in the project's docs repo, optionally scoped to <code>paths</code> .

Method	Signature	Returns	Description
threeWaySections	threeWaySections(base: string, mine: string, theirs: string)	ThreeWaySection[]	Pure three-way section classification (no I/O).

Dependencies

- DocsRepoService

Where it refuses work

- DiffService stops the work with Error when !ref || ref.startsWith('-') .
- DiffService stops the work with an early return when heading === undefined .
- DiffService stops the work with an early return when mineText === theirsText .
- DiffService stops the work with an early return when mineText === baseText .
- DiffService stops the work with an early return when theirsText === baseText .
- DiffService stops the work with an early return when raw === '/dev/null' .

When something fails

- DiffService handles failure in 1 place: it turns it into a return value in all 1.

Diagram

```
sequenceDiagram
    participant Caller as CR Controller / Merge Flow
    participant DiffService
    participant Git as Project Docs Git Repository

    Caller->>DiffService: diffRefs(projectPath, baseRef, headRef)
    DiffService->>Git: git diff --no-renames --numstat baseRef headRef
    Git-->>DiffService: per-file additions/deletions/status

    DiffService->>Git: git diff --no-renames --unified=... baseRef headRef
    Git-->>DiffService: unified diff hunks

    DiffService->>Git: git show baseRef:path / headRef:path
    Git-->>DiffService: document contents
    DiffService-->>Caller: CrDiff with files, hunks, and SectionDiffs

    Caller->>DiffService: threeWaySections(base, mine, theirs)
```

```
DiffService->>DiffService: Align sections by heading
DiffService-->>Caller: ThreeWaySection[] classifications
```

Usage

```
import { DiffService } from './diff.service';

async function reviewChangeRequest(
  diffService: DiffService,
  repoPath: string,
  baseRef: string,
  changeRequestRef: string,
) {
  const diff = await diffService.diffRefs(repoPath, baseRef, changeRequestRef);

  for (const file of diff.files) {
    console.log(
      `${file.status}: ${file.path} (+${file.additions} / -${file.deletions})`,
    );

    for (const section of file.sections) {
      console.log(`  Section "${section.heading}" changed`);
    }
  }

  return diff;
}

async function classifyMerge(
  diffService: DiffService,
  baseMarkdown: string,
  mineMarkdown: string,
  theirsMarkdown: string,
) {
  const sections = diffService.threeWaySections(
    baseMarkdown,
    mineMarkdown,
    theirsMarkdown,
  );

  return sections.filter((section) => section.status === 'conflict');
}
```

AI Coding Instructions

- Invoke Git only through promisified `execFile('git', argv)` ; never construct shell command strings or interpolate refs/paths into a shell command.

- Preserve `--no-renames` behavior for diffs: v1 intentionally represents renames as a deleted file plus an added file.
- Keep `diffRefs` output structured for UI consumers, including file-level status/counts, unified hunks with line numbers, and heading-level `SectionDiff` data.
- Ensure three-way merge logic aligns Markdown content by headings before classifying each section as `unchanged` , `take-theirs` , `keep-mine` , or `conflict` .
- Treat Git refs and repository paths as external inputs: validate assumptions and surface actionable errors when refs, files, or repository state cannot be resolved.

Relationships

- `DEPENDS_ON` → `DocsRepoService`

Referenced By

- `ChangeRequestModule` (`MODULE_PROVIDES`)
- `ChangeRequestModule` (`MODULE_EXPORTS`)