

PipelineMetrics

September 4, 2026

PipelineMetrics

Kind: Interface

Source: `atloria-monorepo/packages/doc-automation/src/v2/utils/pipeline-metrics.ts`

Pipeline Metrics Collector

Collects comprehensive metrics across all pipeline stages for post-run analysis. Persisted to `documentation_jobs.metrics` as JSONB.

Categories tracked:

- Token usage per stage and total
- Screenshot mapping/capture/failure breakdown
- Quality gate + verification scores per document
- Business context extraction stats
- InteractionMap discovery stats
- Stage timing with timestamps
- Document-level metrics (word count, section count)

Properties

Property	Type
<code>tokens</code>	<code>{</code>

```
total: { input: number; output: number; cost?: number };
byStage: Record<string, { input: number; output: number; calls: number }>;
```

```
} | | screenshots | {
totalPlaceholders: number;
mapped: number;
unmapped: number;
```

```

captured: number;
failed: number;
replaced: number;
/** Breakdown by mapping source /
mappingSources: {
interactionMap: number;
siteMap: number;
aiFallback: number;
};
/* Capture types breakdown /
captureTypes: Record<string, number>;
/* Failed capture reasons /
failureReasons: { url: string; reason: string }[];
/* State-guided captures attempted/succeeded /
stateGuided: { attempted: number; succeeded: number };
/* Workflow sequences detected /
workflowSequences: number;
} | / quality | {
/* Quality gate scores /
qualityGate: {
totalScored: number;
passed: number;
failed: number;
regenerated: number;
averageScore: number;
scores: { docId: string; title: string; score: number; passed: boolean }[];
};
/* Verification scores /
verification: {
totalVerified: number;
passed: number;
failed: number;
averageAccuracy: number;
scores: { docId: string; title: string; accuracy: number; claims: number; flagged:
number }[];
};
/* Grounding scores /
grounding: {

```

```

averageScore: number;
belowThreshold: number;
scores: { docId: string; score: number }[];
};
} | | extraction | {
/* InteractionMap stats /
interactionMap: {
totalRoutes: number;
routesWithModals: number;
routesWithTabs: number;
routesWithForms: number;
totalModals: number;
totalTabs: number;
totalFormFields: number;
totalPermissions: number;
} | null;
/* Business context extraction /
businessContext: {
sourcesProcessed: number;
personas: number;
features: number;
glossaryTerms: number;
workflows: number;
extracted: boolean;
} | null;
/* Route mappings from Stage 1 */
routeMappings: {
totalRoutes: number;
framework: string;
withComponents: number;
withTemplates: number;
} | null;
} | | timing | {
pipelineStartedAt: string;
pipelineEndedAt: string;
totalDurationMs: number;
stages: {
name: string;

```

```
startedAt: string;
endedAt: string;
durationMs: number;
status: 'success' | 'failed' | 'skipped';
};
} | | documents | {
total: number;
byType: Record<string, number>;
averageWordCount: number;
averageSections: number;
totalWordCount: number;
} |
```