

Technical diligence: bulletproof-nodejs

Remote repository assessment. Static and read-only.

Overall technical risk

MODERATE

1 high, 1 medium, 6 low, 2 info

Assessment coverage

Complete

14 of 14 checks completed.

Document status

DRAFT · not approved

Findings and narrative need reviewer decisions and verifier finalize before this is a deliverable.

Static checks completed: 14 of 14. Git history available. Build and test execution not performed.

Commit

7ce1b4690c4e on master

Generated

2026-10-04 21:36 UTC

Privacy mode

shareable

Evidence snapshot

f62ab4953c934f0a

01

Assessment

Assessment of <https://github.com/santiq/bulletproof-nodejs.git>. Strengths: Continuous integration is configured (1 pipeline(s)). Dependencies are locked for reproducible installs. Deployment is defined as code in the repository.

Material issues: test suite is small relative to the codebase (high). Evidence gaps remain (see the informational finding).

Assessment coverage: **complete**.

Material findings	Key strengths	Unresolved questions
Test suite is small relative to the codebase	Continuous integration is configured (1 pipeline(s)).	What is the measured line/branch coverage on the main services?
	Dependencies are locked for reproducible installs.	Which node version runs in production, and what is the upgrade plan?
	Deployment is defined as code in the repository.	builds and tests were not executed (execution not requested).

First 30 days	30 to 100 days
Measure coverage in CI and set a ratchet on the critical paths.	Confirm the deployed node version; schedule the upgrade in the first 100 days if it is unsupported.

02
Seller claims

12 claim(s) registered: 3 supported, 2 partially supported, 0 contradicted, 7 unverified. Each status is limited to the scope stated beneath it; supported means the repository evidence agrees within that scope, never more. 12 assessment(s) still await

reviewer confirmation.

All production code for the service lives in this repository

Unverified

C1 · seller email, 2026-09-08

No submodule references were found, which does not establish that no other repository is needed.

Scope:	this repository only; other repositories are not visible to a static analysis
Would settle it:	the seller's inventory of repositories and deployable artefacts; confirmation that the production artefact builds from this repository alone
Assessed by:	automatic, check single_repository
Review:	pending reviewer confirmation

Evidence (1)

repository_structure	50 files, 19 directories, 413844 bytes, max depth 3	9b65088586009cc8dde0632235083ee9
----------------------	---	----------------------------------

CI runs the full test suite on every push

Supported

C2 · CIM, p. 14

.circleci/config.yml runs `npm test`.

Scope:	the commands the CI configuration runs (.circleci/config.yml) and the test files in the repository; that the step runs on every commit and passes is not established
Would settle it:	a recent CI run log showing the test step and its result
Assessed by:	automatic, check tests_in_ci
Review:	pending reviewer confirmation

Evidence (2)

ci_pipeline	CircleCI .circleci/config.yml: 1 jobs, 4 steps	4d6bffdb881ef084eb3e85a80f4f8439
test_infrastructure	1 test files vs 28 source files (ratio 0.04); frameworks: Jest	03e7c582a6ffbcd17ea38cacc1cb6d8d

The codebase has an automated test suite

Supported

C3 · CIM, p. 14

A test suite exists but is small: 1 test file(s) against 28 source file(s) (ratio 0.04). Its adequacy is a finding, not part of this claim.

Scope:	existence of test files and frameworks in the repository; adequacy is assessed separately in the findings, and whether the tests pass requires execution
Would settle it:	what the tests cover, and whether more tests live elsewhere; a test run (execution mode or a CI log) to establish that the suite passes
Assessed by:	automatic, check tests_present
Review:	pending reviewer confirmation

Evidence (1)

test_infrastructure	1 test files vs 28 source files (ratio 0.04); frameworks: Jest	03e7c582a6ffbcd17ea38cacc1cb6d8d
---------------------	--	----------------------------------

No credentials or secrets are committed to the repository

Partially supported

C4 · seller call, 2026-09-08

Found 1 credential-shaped value(s), all with the shape of a development default; no other credential material. Whether any of these values is valid outside development is not established.

Scope:	scan of the indexed files at this snapshot for known token formats and for values assigned to credential names
--------	--

Would settle it: confirmation that deployed environments do not use these values; a git-history scan for previously committed credentials

Assessed by: automatic, check no_committed_secrets

Review: pending reviewer confirmation

Evidence (1)

secret_material	Hardcoded credential assignment at src/config/index.ts:51 (production file, medium)	e5d4d6a0d696d6bfda2eabae6bce5d45
-----------------	---	----------------------------------

The service runs on a supported Node.js LTS release

Unverified

C5 · CIM, p. 9

Repository evidence points against the claim: the repository's declared runtime is past end of life: node 10.15 (EOL 2021-04-30). The runtime in production is not established by the repository.

Scope: runtime declarations and image pins in the repository (build files, CI, container images); the runtime in production is not established

Would settle it: the deployed runtime versions, and the upgrade plan

Assessed by: automatic, check runtime_supported

Review: pending reviewer confirmation

Evidence (0) and counterevidence (1)

Counterevidence

runtime_lifecycle	node 10.15 (image circleci/node:10.15 in .circleci/config.yml): end_of_life, EOL 2021-04-30	89f985a036164d842c0332ad716e1cbf
-------------------	---	----------------------------------

Deployment is fully automated from the repository

Unverified

The claim states automation, which repository definitions alone do not establish.
Definitions found: infrastructure_config (Procfile).

Scope:	deployment definitions in the repository only; the check does not establish an automated release flow
Would settle it:	the release workflow (a CI deploy stage or the platform's deployment configuration) and a recent deployment log
Assessed by:	automatic, check deployment_as_code
Review:	pending reviewer confirmation

Evidence (1)

infrastructure_config	Heroku [heroku]: Procfile	de5c3a8a8d55df57e8ab780af0f361bd
-----------------------	---------------------------	----------------------------------

Dependencies are locked for reproducible installs

Supported

C7 · seller call, 2026-09-08

All 1 manifest(s) have a committed lockfile.

Scope:	production dependency manifests in the repository (package.json)
Assessed by:	automatic, check dependencies_locked
Review:	pending reviewer confirmation

Evidence (1)

dependency_manifest	npm manifest package.json: 45 dependencies, lockfile present	ee6bc23d25f273d254bce3e6a2dc9ad8
---------------------	--	----------------------------------

Two engineers actively maintain the codebase

Unverified

C8 · CIM, p. 4

Repository evidence points against the claim: 0 commit author(s) over 0 commit(s) in the last 365 days, fewer than the 2 claimed. Employment, work on other repositories and commits made under other identities are not visible here.

Scope:	authors of commits in the last 365 days (bot accounts excluded); employment and availability are not visible
Would settle it:	the team list and their commit identities, if commits are made under other names
Assessed by:	automatic, check <code>active_contributors:2</code>
Review:	pending reviewer confirmation

Evidence (0) and counterevidence (1)

Counterevidence

contributor_concentration	last_365_days: 0 contributors over 0 commits, top author n/a, None author(s) cover 50%	ebae307df02b8953b14d90b9b9ea1723
---------------------------	--	----------------------------------

The code is MIT licensed

Partially supported

C9 · CIM, p. 3

The licence file is MIT, but a manifest declares ISC.

Scope:	root licence file and licence fields in manifests
Would settle it:	the seller's confirmation of which declaration is authoritative
Assessed by:	automatic, check <code>license:MIT</code>
Review:	pending reviewer confirmation

Evidence (1) and counterevidence (1)

documentation_file	license: LICENSE, MIT	b4cde7c175a19c1ca72f722c60b11db1
--------------------	-----------------------	----------------------------------

Counterevidence

documentation_coverage	present: README, LICENSE; 0 doc files, 0 ADRs	2b188245d083fbbd46350285765e7631
------------------------	---	----------------------------------

Deployment and operations are documented

Unverified

C10 · seller call, 2026-09-08

Repository evidence points against the claim: the README has no section on deployment (covered: installation, usage, development, architecture, support, changelog). Documentation kept elsewhere in the repository or outside it was not classified.

Scope:	README section headings
Would settle it:	where deployment is documented, if anywhere
Assessed by:	automatic, check documentation:deployment
Review:	pending reviewer confirmation

Evidence (0) and counterevidence (1)

Counterevidence

documentation_file	readme: README.md, 435 words, topics: installation, usage, development, architecture, support, changelog	4b4b833b8f6315b63591da0e602de435
--------------------	--	----------------------------------

No customer data is stored in the repository

Unverified

C11 · seller call, 2026-09-08

This claim is not assessable from repository evidence automatically.

Scope:	no automatic check applies
Would settle it:	a reviewer decision citing evidence, or evidence from the seller
Assessed by:	none
Review:	pending reviewer confirmation

Test coverage is above 70 percent

Unverified

c12 · CIM, p. 14

The claim states a percentage, which repository files cannot measure. Repository evidence points against the claim: no coverage tool or coverage run is configured in the repository. Coverage measured by a CI service or a tool held elsewhere is not visible here.

Scope:	coverage tooling in the repository only; the check does not measure a percentage
Would settle it:	a dated coverage report with its denominator (files or lines measured) and the command that produced it
Assessed by:	automatic, check coverage_measured
Review:	pending reviewer confirmation

Evidence (0) and counterevidence (1)

Counterevidence

test_infrastructure	1 test files vs 28 source files (ratio 0.04); frameworks: Jest	03e7c582a6ffbcd17ea38cacc1cb6d8d
---------------------	--	----------------------------------

Evidence requests for the seller

a dated coverage report with its denominator (files or lines measured) and the command that produced it

a git-history scan for previously committed credentials

a reviewer decision citing evidence, or evidence from the seller

confirmation that deployed environments do not use these values

confirmation that the production artefact builds from this repository alone

the deployed runtime versions, and the upgrade plan

the release workflow (a CI deploy stage or the platform's deployment configuration) and a recent deployment log

the seller's confirmation of which declaration is authoritative

the seller's inventory of repositories and deployable artefacts

the team list and their commit identities, if commits are made under other names

where deployment is documented, if anywhere

03

Material findings

■ High

Test suite is small relative to the codebase

confidence 65% scope repository **review required**

1 test files for 28 source files (ratio 0.04); frameworks: Jest.

Large untested areas raise the cost and risk of every post-close change; file counts are a proxy, so confirm with a coverage measurement.

Ask management

What is the measured line/branch coverage on the main services?

Post-close

Measure coverage in CI and set a ratchet on the critical paths.

Relevance

valuation, effort weeks

Evidence (1)

test_infrastructure	1 test files vs 28 source files (ratio 0.04); frameworks: Jest	03e7c582a6ffbcd17ea38cacc1cb6d8d
---------------------	---	----------------------------------

04

All findings

■ Medium 1 finding(s)

■ Medium

CI runs on node 10.15 (circleci/node:10.15), which is past end of life

confidence 85% scope node

.circleci/config.yml declares node 10.15 via image circleci/node:10.15 (resolved 10.15); vendor end of life 2021-04-30. This is the CI environment, not necessarily the deployed runtime.

The declared runtime no longer receives vendor security fixes. The repository establishes the declaration, not what runs in production; the deployed version and the upgrade plan need confirming.

Ask management

Which node version runs in production, and what is the upgrade plan?

Post-close

Confirm the deployed node version; schedule the upgrade in the first 100 days if it is unsupported.

Relevance

post close risk, effort weeks

Evidence (1)

runtime_lifecycle	node 10.15 (image circleci/node:10.15 in .circleci/config.yml): end_of_life, EOL 2021-04-30	89f985a036164d842c0332ad716e1cbf
-------------------	---	----------------------------------

■ Low 6 finding(s)

■ Low

No recent development activity

confidence 95% scope repository

The most recent commit is 1857 days old; 0 commits in the last 365 days. The README describes the project as: example.

The project describes itself as an example, template or unmaintained; inactivity is consistent with that and carries no operating-company implication by itself.

Ask management

Is this the repository that is deployed, and where does current development happen?

Post-close

Confirm the deployed version against the repository before planning any work.

Relevance

informational

Evidence (1)

commit_activity

0 commits in last 90 days by 0 contributors; last commit 1857 days ago

a74486ed168bd7320d9fbb3aa4bc7df5

■ Low

No recognised dependency-update automation in the repository

confidence 85% scope repository

1 dependency manifest(s) with 26 runtime dependencies and no Dependabot or Renovate configuration.

No Dependabot or Renovate configuration is in the repository; automation held elsewhere (bots, scanners, platform settings) was not inspected. How dependencies are kept current and how vulnerability advisories are tracked is a process question for the team.

Ask management

How are dependency vulnerabilities and upgrades tracked today?

Post-close

Enable automated dependency update proposals with CI validation.

Relevance

post close risk, effort hours

Evidence (1)

dependency_manifest	npm manifest package.json: 45 dependencies, lockfile present	ee6bc23d25f273d254bce3e6a2dc9ad8
---------------------	--	----------------------------------

■ Low

Test coverage measurement not evidenced

confidence 80% scope repository

1 test files; no coverage run or coverage report is evidenced in the checked-in configuration.

Coverage may be measured elsewhere; the repository does not show it. Any coverage claim needs a current report as evidence.

Ask management

Can the team produce a current coverage report?

Post-close

Add coverage measurement to the test command in CI.

Relevance

informational, effort hours

Evidence (1)

test_infrastructure	1 test files vs 28 source files (ratio 0.04); frameworks: Jest	03e7c582a6ffbcd17ea38cacc1cb6d8d
---------------------	--	----------------------------------

■ Low

Default or local development credentials in configuration

confidence 70% scope repository

1 credential-shaped value(s): Hardcoded credential assignment x1; 1 in production or configuration files. Every production-context match has the shape of a development default: a well-known or self-describing password, one equal to the user name, a counted or repeated pattern, or a value that its name, its file or its configuration section marks as a default, seed, development or test value. The host a URL points at was not used to decide this.

Development defaults in committed configuration are common. They matter only if a deployed environment runs with them unchanged, which the repository does not establish.

Ask management

Do any deployed environments run with these default credentials?

Post-close

Confirm deployed environments override the defaults; rotate any that were real.

Relevance

post close risk, effort hours

Evidence (1)

secret_material	Hardcoded credential assignment at src/config/index.ts:51 (production file, medium)	e5d4d6a0d696d6bfda2eabae6bce5d45
-----------------	---	----------------------------------

■ Low

No tagged releases

confidence 70% scope repository review required

No git tags in a 864-day-old repository with 130 commits.

Releases are not identified by tags. Deployments may use commit or image identifiers instead; how a production version is identified and rolled back needs to be asked, not inferred from the absence of tags.

Ask management

How are releases identified and how is a production rollback performed?

Post-close

Introduce versioned, tagged releases tied to deployments.

Relevance

integration cost, effort days

Evidence (2)

release_cadence	No release tags	98d529bd4829306ccb5a1795d1ff7d33
git_repository	130 commits over 864 days, 0 tags, branch master	8610d1137c6593a3cee5f68571e63b36

■ Low

No monitoring integration evidenced in the repository

confidence 55% scope repository review required

Deployment configuration exists but no error tracking, metrics, tracing or incident-management integration was found in dependencies or configuration.

The repository shows no monitoring integration. Monitoring and alerting may exist outside the code; the on-call and incident arrangements need to be evidenced separately.

Ask management

What monitoring, alerting and on-call arrangements exist for production?

Post-close

Obtain monitoring exports and an incident register for an agreed period.

Relevance

post close risk

Evidence (1)

infrastructure_config	Heroku [heroku]: Procfile	de5c3a8a8d55df57e8ab780af0f361bd
-----------------------	---------------------------	----------------------------------

■ Info 2 finding(s)

■ Info

Evidence gaps in this analysis

confidence 100% scope analysis

builds and tests were not executed (execution not requested).

The affected areas were not assessed: evidence gathered later can add risks there or resolve apparent ones. Confirm them through management interviews or a follow-up run with execution enabled.

Evidence (1)

■ Info

No dedicated runbook; README covers architecture

confidence 70% scope repository review required

No dedicated runbook or architecture document beyond the README (0 documentation files, 0 ADRs); the README has sections on architecture.

Operational knowledge exists in the README; whether it is enough to operate the system without the current team is a question for the handover plan.

Ask management

Is there a runbook for deploy, rollback and the common incidents?

Post-close

Ask for or write runbooks for deploy, rollback and the top incident types.

Relevance

informational, effort days

Evidence (2)

documentation_coverage	present: README, LICENSE; 0 doc files, 0 ADRs	2b188245d083fbbd46350285765e7631
documentation_file	readme: README.md, 435 words, topics: installation, usage, development, architecture, support, changelog	4b4b833b8f6315b63591da0e602de435

Scope and limits

No code was executed: build and test results are not part of this assessment.

Not covered: production reliability and incident history, infrastructure cost, per-dependency licensing, vulnerability intelligence, and whether the tests exercise the business-critical workflows.

Remediation effort is a coarse category (hours, days, weeks, months), not an estimate.

Committed secrets and licence findings are pattern matches; liveness and legal interpretation require review.

Evidence appendix

54 observations. Every finding and claim above cites identifiers from this appendix; they are stable across runs of the same repository state on the same day, and facts measured against the calendar (activity windows, days since a commit, time to end of life) take a new identifier when the date changes. Up to three provenance items are shown per observation and excerpts are truncated; the JSON report holds all of them.

Languages & frameworks · 8 observation(s)

Mongoose ^5.10.6 (orm)

framework confidence high 2a98ec99ada796990cdde76aecdaf3

package.json:55

```
"mongoose": "^5.10.6",
```

Express ^4.17.1 (backend)

framework confidence high c1f95604c109223f08851ee2e4f0a423

package.json:29

"express"

JSON: 6 files

language confidence high 20a09afc4b5ef8e1dcfe86e8bd247808

.optic/api/specification.json

.prettierrc.json

nodemon.json

+2 more in the JSON report

JavaScript: 2 files, 21.9% of code

language confidence high 49676b12d9e762194dee047f213cbaab

.eslintrc.js

jest.config.js

TypeScript: 27 files, 77.3% of code

language confidence high 5eda7343db811a47c48c849cc3f1ef1e

src/api/index.ts

src/api/middlewares/attachCurrentUser.ts

src/api/middlewares/index.ts

+2 more in the JSON report

YAML: 3 files

language confidence high a55445d4116656f0e31f4bb840debda4

.circleci/config.yml

.gitpod.yml

optic.yml

Dockerfile: 1 files, 0.8% of code

language confidence high afc22085d2f0e394cc2c125ee78ef70a

.gitpod.Dockerfile

Markdown: 1 files

language confidence high d8e0bd4d39e884629eb9394700b7d26a

README.md

Repository structure · 2 observation(s)

npm component bulletproof-nodejs (package.json)

component confidence high 45e1d191980e896867769bd543a75122

package.json

50 files, 19 directories, 413844 bytes, max depth 3

repository_structure confidence high 9b65088586009cc8dde0632235083ee9

package-lock.json

jest.config.js

src/services/auth.ts

Dependencies · 1 observation(s)

npm manifest package.json: 45 dependencies, lockfile present

dependency_manifest confidence high ee6bc23d25f273d254bce3e6a2dc9ad8

package.json

package-lock.json

Git history & team · 11 observation(s)

.optic: 1 contributors, top owner Contributor 1 (100% of lines)

code_ownership confidence high 36f776efe917c0def34cd6b2dccc2b8d

git log --no-ext-diff --no-textconv...

.circleci: 2 contributors, top owner Contributor 2 (100% of lines)

code_ownership confidence high 540de6c15201f14f51a54baf36247c39

git log --no-ext-diff --no-textconv...

src: 20 contributors, top owner Contributor 3 (54% of lines)

code_ownership confidence high 6429afa1493602926c2d44a54dedc0a7

git log --no-ext-diff --no-textconv...

<root>: 23 contributors, top owner Contributor 3 (21% of lines)

code_ownership confidence high 64619f284f0d00f9d64d2015196cee30

git log --no-ext-diff --no-textconv...

tests: 3 contributors, top owner Contributor 4 (55% of lines)

code_ownership confidence high c8f7ffe4b74cd500021ee236a475c58d

git log --no-ext-diff --no-textconv...

0 commits in last 90 days by 0 contributors; last commit 1857 days ago

commit_activity confidence high a74486ed168bd7320d9fbb3aa4bc7df5

git log --no-ext-diff --no-textconv...

last_90_days: 0 contributors over 0 commits, top author n/a, None author(s) cover 50%

contributor_concentration confidence high 5a0cb599404d0b0fdddeafc412ca6e79

git log --no-ext-diff --no-textconv...

all_time: 30 contributors over 73 commits, top author 19%, 5 author(s) cover 50%

contributor_concentration confidence high e0035cf2d855567f69a2e26235c78145

git log --no-ext-diff --no-textconv...

last_365_days: 0 contributors over 0 commits, top author n/a, None author(s) cover 50%

contributor_concentration confidence high ebae307df02b8953b14d90b9b9ea1723

git log --no-ext-diff --no-textconv...

130 commits over 864 days, 0 tags, branch master

git_repository confidence high 8610d1137c6593a3cee5f68571e63b36

git log --no-ext-diff --no-textconv...

git rev-list --count HEAD

git for-each-ref --format=%(refname:short) refs/tags/

No release tags

release_cadence confidence high 98d529bd4829306ccb5a1795d1ff7d33

git for-each-ref...

Testing · 2 observation(s)

Jest (unit) detected via package, config, file_pattern

test_framework confidence high 6509dfae622ce01e577288491cb14ddf

package.json

package.json

jest.config.js

+1 more in the JSON report

1 test files vs 28 source files (ratio 0.04); frameworks: Jest

test_infrastructure confidence high 03e7c582a6ffbcd17ea38cacc1cb6d8d

tests/sample.test.ts

package.json

Build & runtime · 7 observation(s)

<root>: npm via npm, 9 declared commands

build_configuration confidence high f941797f2d2f778c692b6488ffb0e09b

package.json

package.json

ESLint (linter) via .eslintrc.js

build_tool confidence high 15ae114740967c5fa11bb9ca0cc1e8fc

.eslintrc.js

package.json

Docker build (container_build) via .gitpod.Dockerfile

build_tool confidence high 262dda256e0563a68d60204fd619f248

.gitpod.Dockerfile

npm (package_manager) via package-lock.json

build_tool confidence high 2a615ff0eff537ce32026ce44e4c47dc

package-lock.json

TypeScript (transpiler) via tsconfig.json

build_tool confidence high 524ea8d167d79e73df7d255f32210e00

tsconfig.json

package.json

Prettier (formatter) via .prettierrc.json

build_tool confidence high be7dcc8a06f0a572902b5af6c983d2c

.prettierrc.json

package.json

node 10.15 (image circleci/node:10.15 in .circleci/config.yml): end_of_life, EOL 2021-04-30

runtime_lifecycle confidence high 89f985a036164d842c0332ad716e1cbf

.circleci/config.yml

CI/CD · 1 observation(s)

CircleCI .circleci/config.yml: 1 jobs, 4 steps

ci_pipeline confidence high 4d6bffdb881ef084eb3e85a80f4f8439

.circleci/config.yml

Deployment & infrastructure · 2 observation(s)

.gitpod.Dockerfile: FROM gitpod/workspace-full:latest, gitpod/workspace-mongodb, 2 stage(s)

dockerfile confidence high c57ecc714d8f2142529ca14caa3c7d1c

.gitpod.Dockerfile

.gitpod.Dockerfile:1

FROM gitpod/workspace-full:latest

.gitpod.Dockerfile:2

FROM gitpod/workspace-mongodb

Heroku [heroku]: Procfile

infrastructure_config confidence high de5c3a8a8d55df57e8ab780af0f361bd

Procfile

External services & environment · 16 observation(s)

.env.example declares 8 variables (3 secret-like)

environment_declaration confidence high edeacd68c391aa3ca1fd78b79ea2a1c9

.env.example

variable PORT: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high 133f46dfc871bc557994bcd0675eb3ea

src/config/index.ts:17

.env.example

variable LOG_LEVEL: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high 4268649d24d005e6b696f21475728ee3

src/config/index.ts:34

.env.example

variable MAILGUN_DOMAIN: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high 54a701668ceed4756012bad90c731e6c

src/config/index.ts:65

.env.example

secret-like variable MONGODB_URI: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high 744fce92b8fc5d830741683d89492747

src/config/index.ts:22

.env.example

secret-like variable JWT_SECRET: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high c06e5098107cec3cffda877af7846733

src/config/index.ts:27

.env.example

variable AGENDA_CONCURRENCY: 1 production reference(s), declared in 0 file(s)

environment_variable confidence high c360d6517defd97ecd75425fd6628eca

src/config/index.ts:43

variable AGENDA_DB_COLLECTION: 1 production reference(s), declared in 0 file(s)

environment_variable confidence high c932870aa22091727798f10d8febfa4f

src/config/index.ts:41

variable NODE_ENV: 4 production reference(s), declared in 0 file(s)

environment_variable confidence high da222c15434ef480d09eb372bdc6bc6c

src/config/index.ts:4

src/config/index.ts:4

src/loaders/express.ts:39

+1 more in the JSON report

variable AGENDA_POOL_TIME: 1 production reference(s), declared in 0 file(s)

environment_variable confidence high efc5908f65c095794ce03163d2eb9fe7

src/config/index.ts:42

secret-like variable MAILGUN_API_KEY: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high f178281c51362d14cd86e663dfb9b755

src/config/index.ts:63

.env.example

variable JWT_ALGO: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high f6448fce2daf598110fa76d61bb83448

src/config/index.ts:28

.env.example

variable MAILGUN_USERNAME: 1 production reference(s), declared in 1 file(s)

environment_variable confidence high fee4bde549304b989a3b785be1238068

src/config/index.ts:64

.env.example

Mailgun (email) via env, package: production usage established

external_service confidence high 0d06f6b05723227f445bb56550a1e6d3

derived from 4 observation(s)

package.json

MongoDB (database) via env, package: production usage established

external_service confidence high 55c83a1ab986723c717b1b29a1b8c769

derived from 2 observation(s)

package.json

Hardcoded credential assignment at src/config/index.ts:51 (production file, medium)

secret_material confidence medium e5d4d6a0d696d6bfda2eabae6bce5d45

src/config/index.ts:51

Documentation · 4 observation(s)

present: README, LICENSE; 0 doc files, 0 ADRs

documentation_coverage confidence high 2b188245d083fbbd46350285765e7631

.editorconfig

LICENSE

README.md

readme: README.md, 435 words, topics: installation, usage, development, architecture, support, changelog

documentation_file confidence high 4b4b833b8f6315b63591da0e602de435

README.md

```
editor_config: .editorconfig
```

```
documentation_file    confidence high    9e9215bd9b2e04a67b69015b91ecc50d
```

```
.editorconfig
```

license: LICENSE, MIT

```
documentation_file    confidence high    b4cde7c175a19c1ca72f722c60b11db1
```

LICENSE

07

Run metadata

Analyzers

```
ci_cd · ok 1 observation(s), 58 ms, 1 of 1 files inspected.
```

```
dependencies · ok 1 observation(s), 15 ms, 1 of 1 files inspected.
```

documentation · ok 4 observation(s), 29 ms.

```
git_history · ok          11 observation(s), 1723 ms.
```

```
infrastructure · ok      2 observation(s), 13 ms, 1 of 1 files inspected.
```

languages · ok 6 observation(s), 2 ms.

secrets · ok 1 observation(s), 141 ms, 46 of 46 files inspected.

```
structure · ok                2 observation(s), 5 ms.
```

build · ok	6 observation(s), 14 ms.
environment · ok	13 observation(s), 98 ms, 33 of 33 files inspected.
frameworks · ok	2 observation(s), 5 ms.
lifecycle · ok	1 observation(s), 6 ms.
tests · ok	2 observation(s), 6 ms.
build_execution · skipped	0 observation(s), 0 ms. sandboxed execution disabled (pass --execute to enable)
services · ok	2 observation(s), 15 ms.
test_execution · skipped	0 observation(s), 0 ms. sandboxed execution disabled (pass --execute to enable)

Ingestion: git index, 50 files, 413844 bytes, cloned. Execution: disabled (execution not requested).
Rules run: 32.Privacy: 12 contributor identities pseudonymised; absolute local paths reduced to their last component in all fields; automatic redaction applied; review before external distribution.
Generated by verifier 0.1.0.

Reports minimise sensitive data and apply automatic redaction; review is still required before external sharing. Findings flagged for review need analyst confirmation.