API Design Quality Analysis Report

This report provides an assessment of your API design quality based on the provided OpenAPI specification, benchmarked against industry best practices. Scores range from 0 (very poor) to 10 (excellent). Benchmarks represent the expected level of maturity for well-designed APIs.

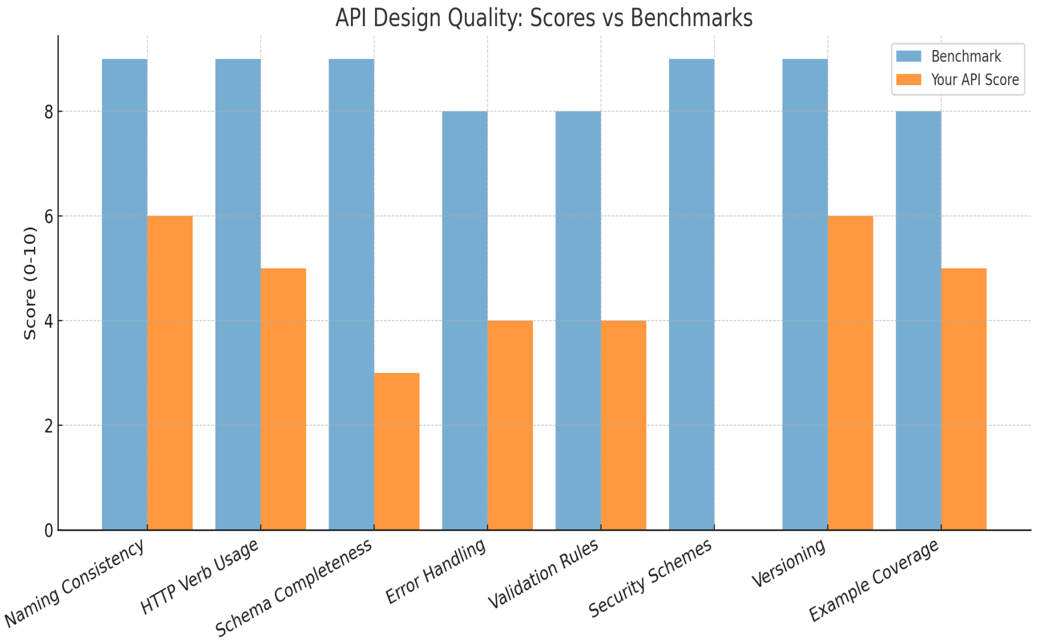
Observation Summary

Strengths: Good use of versioning, generally consistent naming, appropriate use of HTTP verbs in some areas. **Weaknesses:** Missing security schemes, incomplete schemas, poor error handling documentation, lack of validation rules.

Category Scores vs Benchmarks

Category	Score	Benchmark
Naming Consistency	6	9
HTTP Verb Usage	5	9
Schema Completeness	3	9
Error Handling	4	8
Validation Rules	4	8
Security Schemes	0	9
Versioning	6	9
Example Coverage	5	8

Visual Score Comparison



Detailed Observations & Recommendations

Naming Consistency — Score 6 vs Benchmark 9

- Path names mix snake_case and action-oriented styles.
- Adopt kebab-case for URLs.
- Prefer resource-oriented naming.

HTTP Verb Usage — Score 5 vs Benchmark 9

- Overuse of POST for update operations.
- Use PUT for full replacements and PATCH for partial updates.
- Ensure GET endpoints do not cause state changes.

Schema Completeness — Score 3 vs Benchmark 9

- Many schemas are empty or undefined.
- Define complete schemas with accurate field types and descriptions.
- Reuse common schemas via components.

Error Handling Documentation — Score 4 vs Benchmark 8

- Only partial error handling is defined.
- Add standard error responses (401, 403, 404, 500).
- Create reusable error response schema.

Validation Rules — Score 4 vs Benchmark 8

- Minimal validation rules present.
- Add min/max lengths, patterns, enums, and numeric ranges.

Security Schemes — Score 0 vs Benchmark 9

- No security schemes are defined.
- Add API Key, OAuth2, or Bearer authentication in components/securitySchemes.
- Apply globally or per operation.

Versioning — Score 6 vs Benchmark 9

- Versioning strategy exists but inconsistent.
- Consolidate endpoints under canonical versions.
- Mark deprecated APIs using deprecated: true.

Example Coverage — Score 5 vs Benchmark 8

- Request examples present but response examples missing.
- Add representative success and error response examples.