Here's an automated AI-generated review to support your development workflow.

📝 Summary

- Adds configurable rate limiting per API key using a token-bucket style limiter
- Introduces middleware that returns 429 with Retry-After when limits are exceeded
- Reads limits from config with sensible defaults for development vs production

⚠️ Potential Issues

- In-memory store means limits are per process; under multiple instances the effective limit is N × limit per key
- No distinction between authenticated and unauthenticated traffic; consider stricter limits for anonymous requests
- Retry-After is in seconds only; some clients may benefit from a more precise hint for backoff

🚀 Potential Optimizations

- Use a shared store (e.g. Redis) for rate state if you run more than one instance
- Consider separate limit configs for different endpoint groups (e.g. read vs write)
- Add a small header (e.g. X-RateLimit-Remaining) so clients can avoid hitting the limit

💡 Suggestions

- Security and throttling best practices applied; config validation is clear
- Config validation feedback: consider validating min/max limits at startup
- Docs and examples suggested: add a short README section on tuning limits per environment

---