



AI-Accelerated Node.js Development

Faster, Cleaner, More Maintainable Code with Copilot or Claude or Augment

Tyler Steck

Packsize International

Lucas Kamakura

Augment Code

International JavaScript Conference NYC 2025



About Us



Tyler Steck

Senior Engineer at Packsize International

Enthralled by AI and enjoy tinkering with Node.JS!



Lucas Kamakura

Solutions Architect at Augment Code

Provides AI-assisted development tools to help teams build better software faster.

The Node.js Initiatives

Manual, repetitive tasks that consume weeks of developer time



Large Feature Integration

Cross-cutting changes (API/Services/DB), docs and rollout plans — often days to weeks.



Untangling Async & Side Effects

Callback hell, hidden state, messy code — hours to days



Tests at Scale

Expanding integration/E2E, fixing flaky tests and mocks — hours to days



Dependency Upgrades & Security

Major version bumps, package dependencies, policy enforcement — hours to days

AI Agents + Process Enablement

Same tasks, less time — realistic reductions from disciplined AI usage



Implementing a New Feature

Scoping + scaffolding + happy-path code.

≈75% faster



Refactoring Legacy Code

Async/await, separation of concerns, test seams.

≈90% faster



Writing Tests

Unit → integration → E2E, with fixtures/mocks hydrated by AI.

≈90% faster



Security Recommendations

Mitigate vulnerabilities, implement major version upgrades and impact.


≈50% less time


AI as a Pair, Not a Replacement


Harmony: put the right tool in the engineer's hands - AI as a Pair




Developers

 Envision & Architect


 Quality & Security


 Review & Tune


 Orchestrate AI Agents




AI

 Code Generation

 Agents with MCP

 Codebase Awareness

 SDLC Integrations



AI-Powered Project Genesis

Prompt engineering is key for AI to generate a proper foundation

New Project Builders



lovable.dev



bolt.new



v0.dev



replit.com

Effective Prompt Strategy



Find similar projects using "like" comparisons



Specify frameworks, configs, and expectations



Prompt Examples



 Detailed prompts = better foundations. Iterate outside of these tools, then export and scale with your editor.



Enterprise AI Tools

Proper configuration is the foundation of successful AI-assisted development

AI Coding IDE & Agents



GitHub Copilot



Cursor



Claude



Augment

Essential Configuration Steps



User guidelines



AGENTS.md and `**/rules/*.md`



Codebase indexing and prompt w/ context



Connect Services + MCP (Model Context Protocol)





Demo 1: Refactor Legacy Node

Start with configuration (rules/context), then modernize with clarity and review

[Click to view code](#)

 **Before**

Callback Hell

Nested callbacks making code unreadable

Inline Queries

Database logic scattered throughout

Inconsistent Patterns

Mixed validation and error handling

[Click to view code](#)

 **After**

Async/Await

Clean, readable asynchronous code

Service + Repository

Proper separation of concerns

Best Practices

Consistent validation and error handling



Developer configures best practices → AI suggests → Developer reviews



Docs-as-Code in Node

Swagger/OpenAPI, READMEs, ADRs - generated and kept in sync



Jest Unit Tests

Auto-generate test suites with edge cases and mocking



API Documentation

Create Swagger/OpenAPI docs automatically from your code



Inline Comments

Generate meaningful comments for complex business logic



Always Review

Generated content still should receive human validation



Demo 2: TDD Acceleration in Node

Red > Green > Refactor with Jest and focused prompts

[Click to view details](#)

RED: Failing Tests

GitHub Issue #2

Feature: Add GET /users/:id/posts endpoint

E2E Tests Written

3 tests: user posts, 404, 400 validation

Tests Failing

2 failed, 1 passed - endpoint doesn't exist

[Click to view implementation](#)

GREEN: Passing Tests

AI with MCP

Reads issue, understands codebase patterns

Generates Code

Controller, service, DTOs, routes, Swagger

Tests Pass

3 passed, 0 failed - feature implemented!



Write failing test → AI implements → Tests pass → Refactor with confidence





Demo 3: CI/CD + Copilot PR Review

Matrix builds, security gates, and AI-assisted reviews for Node



Release Notes

Auto-generate from PRs and commits



Pipeline Fixes

Suggest fixes for build failures



Flaky Tests

Detect and stabilize unstable tests



GitHub Integration


GitHub Actions and Copilot Chat




AI Isn't Magic - The Pitfalls


Power with responsibility. You are the guardrail.

Business Impact


 Faster delivery and iteration

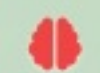
 Pattern reuse and consistency


 Throughput on high-value work


 Creative problem-solving

Risks without Guardrails

 Security & compliance exposure (secrets, licenses, PII)

 Confidently wrong outputs (hallucinations)

 Skipping review or blindly trusting

 Context gaps in distributed systems

Driving AI Adoption & Leadership Buy-in

Building the business case for AI coding tools investment



Measure

PR Cycles, Time to first commit, Review Time, MTTR



Pilot

Start small with a low-risk, high-impact use case



Prove

Build ROI case from results. Is it truly the right investment?



Scale

Enable broader organizational adoption

→ AI should be a net positive for your team! ←



Call to Action

Start your AI-accelerated Node.js journey today



Try It Now

Start with your next refactor or project



Define Rules

Set team guidelines for AI usage



Share Patterns

Document successful AI patterns



Remember

AI is here to accelerate, not to replace



Thank You!

Questions & Discussion

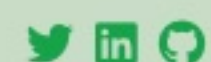


AI is here to accelerate, not replace

Let's build the future of Node.js development together

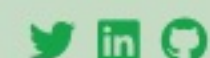
Tyler Steck

Packsize International



Lucas Kamakura

Augment Code



Rate this session

devm.io/reader/experience/devmio-ijs

