payara®

The Essential Guide to AI Tools

for Jakarta EE Developers

Power Up Your Enterprise Java

User Guide

Guide Updated: **November 2025**

Contents

Summary

Artificial Intelligence (AI) is rapidly transforming the Jakarta EE development landscape. Leveraging AI-powered tools can dramatically accelerate your workflow, whether you're new to Jakarta EE or a seasoned developer. In particular, key AI-powered solutions can help you improve code quality and productivity while helping you focus on innovation rather than repetitive tasks. This guide explores the latest AI tools available for Jakarta EE developers, their unique advantages and practical examples to help you get started.

Introduction

Jakarta EE, the evolution of Java EE and J2EE, has long been the backbone of enterprise Java applications, helping developers build mission-critical systems for a variety of operations. Yet, as the demands of modern software development have grown, so too have the expectations placed on Jakarta EE developers. In particular, faster prototyping and delivery demand optimum productivity and efficiency.

Recent innovations can help Jakarta EE developers and software engineers keep up with today's challenges while minimizing cognitive overload. One of these advances is associated to AI, which has moved from the realm of research and novelty into a practical, indispensable part of the developer's toolkit over the las few years.

Traditional Challenges in Jakarta EE Application Development

Before diving into the AI revolution, it's worth reflecting on the traditional landscape of Jakarta EE application development. For years, specialists have been able to leverage a number of accessories to support their activities, such as project generators, visual designers and auto-completion tools within integrated development environments (IDEs). While these have been providing a solid foundation and helped developers optimize their productivity, they also faced key limitations:

- **Project Generators and Archetypes:** These tools help scaffold new projects but often lock developers into rigid, predefined templates. Customizing these templates to fit real-world requirements can require considerable efforts, especially for large or complex applications.
- **Visual Designers:** Tools for designing database schemas or user interfaces (UIs) are helpful, but they lack context-awareness. Developers must manually place each component, risking inconsistencies and slowing down the process.
- **IDE Assistance:** While IDEs offer auto-completion, their suggestions are often limited to syntax and don't fully grasp the project's broader context or business logic.

While helpful, these solutions are no longer sufficient to effectively address today's developers' pain point. However, the next generation of developer tools can outperform the capabilities of traditional tools, reducing the time needed to perform repetitive and/or time-consuming tasks.

The AI Advantage: How Jakarta EE Application Development can be Enhanced

When infused in developer tools, AI can help offer highly responsive, context-aware, prompt-driven assistance that automates a number of tasks while boosting the performance and productivity of software development activities. In particular, AI-powered tools address the restrictions of traditional solutions by introducing:

- **Adaptive Project Generators:** Scaffold full-stack applications from natural language prompts, adapting templates to your needs.
- **Context-Aware Visual Designers:** Design UIs and databases using natural language, with AI helping drive consistency and best practices.
- **Natural Language Archetypes and Vibe Coding Tools:** Generate code (e.g., REST endpoints) by simply describing the required functionality.
- **Automated Test & Documentation Generation Tools:** AI analyzes code to create relevant tests and up-to-date documentation.
- **Intelligent Code Completion Tools:** Context-aware suggestions that examine the entire project, not just syntax.

These capabilities can have considerable impact on the day-to-day operations within an application development team. While the corpus of research on the topic is rapidly expanding, current studies quantified key benefits. The next session discusses the top reports on the topic.

The Broader Impact: Stats and Trends

Recent surveys confirm that AI adoption is no longer a fringe trend among Java developers, and the use of AI-powered solutions is only likely to rise in the industry. The 2025 JRebel Java Developer Productivity Report reveals that 50% of Java developers are now using AI tools to boost productivity [1]. This marks a significant shift from just a few years ago, when AI assistants were seen as experimental or niche.

In effect, the capabilities of such solutions improved considerably. While concrete Java and Jakarta EE-specific statistics are not currently available, industry research shows that AI-powered code generation can accelerate software development by nearly 56% [2]. More precisely, according to the JRebel report, among those using AI tools, there are clear trends in what tasks they help save time. Primarily, developers turn to AI tools for code completion. Other popular use cases include error detection (30%), documentation generation (28%), debugging assistance (26%) and automated testing (21%) [1].

In addition to helping enhance developer productivity, AI-powered tools for documentation and testing can support measurable reductions in bugs and faster onboarding. Specifically, while a 2023 GitHub Copilot study found that developers using AI assistants reported higher code quality and fewer bugs, especially for repetitive or boilerplate code [3]. Besides, AI tools for documentation and code explanation can help accelerate onboarding and knowledge transfer. Finally, software specialist can benefit from a better work experience, with improvements in happiness, flow and fulfilment.

Thanks to these multiple advantages, organizations empowering their software development teams with AI tools can reduce tech debt and deliver software faster. Ultimately, this can lead to productivity gains equivalent to saving 20 to 45% of current annual spending on software engineering.

Comparative Snapshot: Traditional vs. AI-Enhanced Workflow

Task	Traditional Approach	AI-Enhanced Workflow
Project Setup	Archetypes + manual edits	Natural language prompts + smart scaffolding
UI Design	Manual placement in visual tools, limited context awareness	AI-generated from context
Code & REST Endpoints	Write from scratch	Describe in plain English
Test Coverage	Manually written	Auto-generated per method/class
Documentation	Manually written	Real-time, consistent Javadoc
Code Suggestions	Limited by IDE syntax	Context-aware with full project understanding

Deep Dive: AI Tools Every Jakarta EE Developer Should Know

As technology rapidly advances, new AI-powered developer tools are continuously being released. Let's look at two standout tools that are pushing the boundaries for Jakarta EE developers specifically: Payara Starter and Jeddict AI Assistant.

Payara AI Agent (Experimental Feature)

Website: https://www.payara.fish/downloads/payara-platform-community-edition/

The **Payara AI Agent** is an experimental capability integrated into the **Payara Server Maven Plugin**, designed to bring **intelligent automation and observability** into Jakarta EE application development.

By default, the AI Agent runs in **development mode** (disabled in other modes for safety). Developers can also enable it manually using the payara.ai.agent system property or the PAYARA_AI_AGENT environment variable.

What It Does

- The AI Agent augments the developer experience by interpreting **natural language instructions** and extracting **contextual insights** from a running Payara Server instance. Its core abilities include:
- Translating plain-English requests into **asadmin commands** and providing human-readable explanations.
- Querying **server configuration and runtime state** through Domain REST endpoints and JMX MBeans.
- Executing **safe, read-only commands** (e.g., listing deployed applications, viewing JDBC connection pool status).
- Parsing and summarizing **server logs** and configuration files (like domain.xml) to surface key insights.
- Offering **intelligent prompts and auto-completions** to streamline common server administration and troubleshooting tasks.
- Configurable to work with **multiple AI** service providers (e.g., OpenAI, Google, Anthropic, Mistral, Ollama, and others), giving developers flexibility in choosing their preferred backend.

Benefits

- Simplifies server administration by reducing reliance on memorized CLI commands.
- Improves developer productivity with **faster diagnostics and insights** during local development.
- Enhances onboarding for new team members through **natural language guidance**.
- Reduces errors by ensuring only **safe, non-destructive commands** are executed.

Payara Starter

Website: https://start.payara.fish/

Part of Payara Platform Enterprise's Developer Tools, Payara Starter is a code and entity relationship (ER) diagram generator. It incorporates AI functions and natural language commands to help users generate, customize, edit and refine code and diagrams to address project needs while ensuring developers can maintain full control over the end Jakarta EE/MicroProfile application. The project scaffolding tool can then be fed to any of Payara Platform runtimes, including Payara Server and Payara Cloud.

What It Does

Here's a brief overview of the capabilities and core functionalities of Payara Starter Enterprise:

- **AI-Driven ER Diagram Generation:** Describe your app in natural language (e.g., "Software Development Conference"), and Payara Starter Enterprise generates an ER diagram with relevant entities and relationships, such as "Conference", "Attendees" and "Sessions".
- **Iterative Design:** Expand your diagram with chat commands or the "+ enlarge" button to add new entities, e.g. "Feedback", "Sponsors" and "Workshop". It's possible to go back or reduce the diagram at any time.
- **Real-Time Feedback:** Simply describe the functionalities needed in the chatbox and the AI proactively generates or modifies the code.
- **One-Click App Generation:** Once the design is ready, it's possible to generate the full application.
- **Full-Stack Output:** Backend (JPA entities, relationships) and frontend (UI with titles, icons and descriptions) are automatically generated.

Benefits

- Reduces manual setup and customization.
- Helps ensure consistent UI and data model design.
- Accelerates prototyping and production.

Jeddict AI Assistant

Website: https://jeddict.github.io/

Integrated with Apache NetBeans IDE, open-source Jeddict brings a suite of AI-powered features within its AI Assistant. These can improve software development workflows, making it easier for professionals to build robust Java/Jakarta EE applications efficiently. While it is designed for Java and Jakarta EE, the tool is flexible enough to assist with other programming languages.

What It Does

Here's a brief overview of the capabilities and core functionalities of Jeddict:

- **Intelligent Code Completion:** Get context-aware suggestions as you type.
- **Inline Hints:** View method usage, expected parameters and best practices directly in the editor.
- **Context-Aware Naming:** Get suggested meaningful variable and method names based on conventions and best practices.
- **Inline SQL Completion:** Benefit from real-time help with database queries.
- **Automated Logging & Documentation:** Automate the generation of logs and Javadocs for classes/methods.
- **Test Case Generation:** Automatically create relevant test cases for the code provided.
- **LLM Provider Selection:** Choose the preferred AI backend, while using secure API integration.
- **REST Endpoint Generation:** Instantly create resource classes, annotations and models.
- **AI Chat:** Select files/packages for context-aware conversations and suggestions.
- **AI-Generated Commit Messages:** Benefit from clear, meaningful commit messages for better collaboration.

Benefits

- Automates boilerplate and repetitive tasks.
- Helps ensure code quality and consistency.
- Saves time on testing, documentation and endpoint creation.

Conclusions: Embracing the AI-Powered Future

AI-powered solutions are becoming essential for anyone working in the software industry, including Jakarta EE developers. These are key to help professional competitive and efficient in an increasingly more demanding sector, as they empower developers to focus on the creative and strategic aspects of application development. As a result, it is possible to accelerate project delivery while raising the bar for code quality, maintainability and team collaboration.

When it comes to Jakarta EE application development, Payara AI Agent, Payara Starter and Jeddict are redefining what's possible, promising even greater productivity for developers of all experience levels. As these technologies continue to evolve, the opportunities for smarter, more efficient workflows will only grow.

Ready to experience the next generation of Jakarta EE development?

Start building your next application with Payara today. Join the thriving Payara Platform Community for free and experience first-hand how the developer tools provided by the technologies can accelerate your software development workflow, from concept to deployment.

References

[1] JRebel. (2025). Java Developer Productivity Report 2025. Available at: https://www.jrebel.com/resources/java-developer-productivity-report-2025

[2] Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. arXiv preprint arXiv:2302.06590. Available at: https://arxiv.org/abs/2302.06590

[3] GitHub. (2023). GitHub Copilot X: The AI-powered developer experience. Available at: https://github.blog/2023-03-22-github-copilot-x-the-ai-powered-developer-experience/

[4] JetBrains. (2024). The State of Developer Ecosystem 2024. Available at: https://www.jetbrains.com/lp/devecosystem-2024/

Interested in Payara? ***Book a Free Demo***

PAYARA PLATFORM:
POWER UP YOUR ENTERPRISE JAVA

BOOK A FREE DEMO

payara ENTERPRISE payara QUBE payara MICRO

sales@payara.fish

UK: +44 800 538 5490
Intl: +1 888 239 8941

www.payara.fish