

10 good reads from the Java Magazine archives

[How the JVM Locates, Loads, and Runs Libraries by Oleg Šelajev](#)

[Building Apps Using WebSockets, Part 1, and Pushing Data in Both Directions with WebSockets, Part 2, by Danny Coward](#)

[Runtime Code Generation with Byte Buddy by Fabian Lange](#)

[jsoup HTML Parsing Library by Mert Çalışkan](#)

[Escape Analysis in the HotSpot JIT Compiler by Chris Newland and Ben Evans](#)

[Project Lombok: Clean, Concise Code by Josh Juneau](#)

[For Faster Java Collections, Make Them Lazy by Mike Duigou](#)

[Annotations: An Inside Look by Cedric Beust](#)

[Blockchain: Using Cryptocurrency with Java by Conor Svensson](#)

[Showtime! Java 7 is here: Interview with Mark Reinhold by Michael Meloan](#)

[Conclusion](#)

[Dig deeper](#)

## CODING

# 10 good reads from the Java Magazine archives

For years, some important articles have not been available on the magazine website. They are now.

by *Java Magazine Staff*

June 11, 2021

*Java Magazine* has gone through several transitions in its decade-long history. With each transition, some of the earlier material has become harder to find.

As part of our 10th anniversary retrospective, the editors spent some time locked in the vault searching for articles that are still relevant and important, but that have not been available on the [magazine website](#). They are now! Here are 10 of the best articles we found in the archives.



## **How the JVM Locates, Loads, and Runs Libraries by Oleg Šelajev**

Class loading remains one of the least understood operations of the JVM. Many an article discusses code execution or garbage collection, but both of those actions first require a class to be parsed and loaded. This remarkably lucid overview of the process gives you an understanding of how much is done to prepare a class for execution. At the time this article was written, Java 8 was the current release. While everything in the article still applies, there are today additional wrinkles to the process due to the advent of modules. However, if your code doesn't use modules, the process described here applies to your code.

**[Building Apps Using WebSockets, Part 1, and Pushing Data in Both Directions with WebSockets, Part 2, by Danny Coward](#)**

WebSockets are an invaluable way of communicating between server and client, and they can be programmed rather easily in Java EE or Jakarta EE. This two-part tutorial shows how to set up both ends of a chat application and how to push data back and forth asynchronously. If you don't use WebSockets in your present projects, these two articles might well convince you to use them and thereby solve many otherwise difficult communication problems.

### **Runtime Code Generation with Byte Buddy by Fabian Lange**

Multiple solutions are available for manipulating Java bytecode. ASM is perhaps the best known. While it is used by many tools, it lacks the user-friendliness of Byte Buddy, which does not require you to be a low-level cognoscenti to be able to create Java classes on the fly or modify existing classes. This overview with several examples opens the door to doing magical things in your programs. (For more on ASM, see Ben Evans' ["Real-World Bytecode Handling with ASM."](#))

### **jsoup HTML Parsing Library by Mert Çalışkan**

HTML is notoriously difficult to parse, in part because web browsers are so forgiving of ill-formed pages. So, a reliable, easy-to-use library for parsing, validating, and manipulating HTML files is a valuable addition to a Java toolbox. jsoup provides just such capability with a comparatively intuitive API. jsoup even enables you to sanitize HTML files by allowing only the elements you specify in an allowlist. Basically, for anything you need to do with HTML files, jsoup delivers an easy-to-use API.

### **Escape Analysis in the HotSpot JIT Compiler by Chris Newland and Ben Evans**

The JVM is universally viewed as a wondrous technology for performance that puts it on a par with native binaries and for its capacity to run code from the earliest releases of Java to the present. Part of its speed comes from extreme optimizations that squeeze performance from unlikely sources. One of those is escape analysis, which examines the scope in which a variable's lifespan occurs. And when analysis shows definite limits to the scope, the JVM can implement a variety of micro-optimizations: creating objects on the stack rather than the heap, not bothering with locks for certain multithreaded operations, and so forth. This article discusses how the JVM performs escape analysis and how your code can benefit from it.

### **Project Lombok: Clean, Concise Code by Josh Juneau**

Even the most ardent supporters of Java concede that it is a verbose language. While that verbosity helps to make Java programs readable, it can be frustrating when writing code. Project Lombok is the most widely adopted of the many proposed solutions for reducing some of that verbosity. It uses annotations to generate code (such as getters and setters) and to add safety (null safety) to programs. This article gives an excellent overview of Lombok, with an eye to its most useful capabilities.

### **For Faster Java Collections, Make Them Lazy by Mike Duigou**

Laziness in programming means deferring an operation until the software can definitively ascertain that the operation must be performed. It's the conceptual opposite of eagerness, in which operations are performed at the earliest possible moment. In this article, the former lead developer of the Java Collections framework describes how the addition of laziness to collections diminished memory usage significantly in certain applications. In addition, he explains how to think about including laziness in your own programs to gain the widest benefit.

### **Annotations: An Inside Look by Cedric Beust**

Annotations are ever more widely used in Java but remain little understood. In this article, Cedric Beust, who worked on annotation technology in the JDK, explains their unique syntax and shows how to write your own annotations. He then dives into the Java annotation processor and shows how your annotation would be processed by the JDK and ultimately by the JVM at runtime.

### **Blockchain: Using Cryptocurrency with Java by Conor Svensson**

If you're itching to work with blockchains and with cryptocurrencies such as Ether, this article gives you all the conceptual basics as well as a thorough programming introduction. It explores web3j, a key Java library for accessing the Ethereum blockchain, and demonstrates how to execute simple transactions, create contracts, and even how to issue your own virtual tokens—all in Java.

### **Showtime! Java 7 is here: Interview with Mark Reinhold by Michael Meloan**

The Java SE 7 release addressed important areas in both the language and the platform, reflecting trends in the programming community, developments in hardware architectures, and a continuing commitment to the success of Java technologies. Mark Reinhold, chief architect of the Java platform, walks through the highlights, including Project Coin, the

[invokedynamic](#) bytecode instruction, [New I/O Part 2](#), and the [fork/join](#) framework. The interview also touches on future releases, including Java SE 8's [Project Lambda](#).

## Conclusion

These 10 articles focus a lot on what the JDK does with your code and how to make best use of its features. Like all runtime environments, the JVM benefits from deep programmer knowledge of its operations, both by improved performance but also by writing code that exploits little-used capabilities. From these articles, you'll find many new instruments to put in your toolbox and certainly new ways to consider your code.

## Dig deeper

- [Happy anniversary! Celebrating 10 years of Java Magazine](#)
- [The best of 2020: The 10 most popular Java Magazine articles](#)



## Java Magazine Staff

Java Magazine staff ([javamag\\_us@oracle.com](mailto:javamag_us@oracle.com), [@Oraclejavamag](#)) deliver authoritative information about Java, the JVM, and JVM languages to a community of more than one-quarter million developers.

## Share this Page



### Contact

US Sales: +1.800.633.0738  
Global Contacts  
Support Directory  
Subscribe to Emails

### About Us

Careers  
Communities  
Company Information  
Social Responsibility Emails

### Downloads and Trials

Java for Developers  
Java Runtime Download  
Software Downloads  
Try Oracle Cloud

### News and Events

Acquisitions  
Blogs  
Events  
Newsroom

ORACLE

Integrated Cloud  
Applications & Platform Services



© Oracle | [Site Map](#) | [Terms of Use & Privacy](#) | [Cookie Preferences](#) | [Ad Choices](#)