**拓展阅读**

电子书有效期:12月10日

**Java**

* <https://www.oracle.com/java/index.html> Oracle Java
* <https://developer.oracle.com/index.html> Java in the Cloud
* <https://go.java/netflix.html> Netflix and Java
* <https://go.java/twitter.html> Twitter and Java
* <https://google.github.io/styleguide/javaguide.html> Google Java Style Guide
* <https://www.codecademy.com/learn/learn-java> Learn Java
* <https://github.com/trending/java> GitHub Java Trending Repos
* <http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html>JDK 8 Document

**Java Concurrency**

* <http://docs.oracle.com/javase/tutorial/essential/concurrency/threads.html> Java Threads
* <http://docs.oracle.com/javase/tutorial/essential/concurrency/sync.html> Java

**Synchronization**

* <http://docs.oracle.com/javase/tutorial/essential/concurrency/immutable.html> Immutable

**Objects**

* <http://docs.oracle.com/javase/tutorial/essential/concurrency/executors.html> Executors
* <https://sourceforge.net/projects/javaconcurrenta/> Java Concurrency Animated

JVM

* <http://www.cubrid.org/blog/dev-platform/understanding-jvm-internals/> Understanding JVM Internals
* <https://docs.oracle.com/javase/specs/jvms/se8/jvms8.pdf> JVM Spec
* <https://docs.oracle.com/javase/specs/jvms/se8/html/jvms-5.html#jvms-5.3.1> JVM Bootstrap class loader
* <http://www.javaworld.com/article/2078645/java-se/jvm-performance-optimization-part-3-garbage-collection.html> JVM Performance Optimization - Garbage Collection
* <http://www.javaworld.com/article/2078731/java-platform/jvm-performance-optimization--part-5--is-java-scalability-an-oxymoron-.html> JVM Performance Optimization - Scability

**Java Servlet**

* <http://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html> Java Servlet Technology
* <http://docs.oracle.com/javaee/6/tutorial/doc/bnafi.html> Servlet lifecycle

**Servlet Container**

* <https://dzone.com/articles/what-servlet-container> Servlet container overview
* <http://stackoverflow.com/questions/5039354/difference-between-an-application-server-and-a-servlet-container> Application Server VS Servlet Container
* <http://tomcat.apache.org/> Apache Tomcat
* <http://www.eclipse.org/jetty/> Jetty
* <http://netty.io/> Netty async event driven web framework

**Spring Framework**

* <https://projects.spring.io/spring-framework/> Spring Framework project home
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#overview-dependency-injection> DI and IoC
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#overview-core-container> Spring Core Container
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#beans-factory-scopes> Bean Scope
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#beans-java> Java Based Bean Config

**Spring MVC**

* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#mvc> Spring MVC overview
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#mvc-servlet>Dispatcher Servlet
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#mvc-ann-methods> Request mapping handler
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#mvc-uri-building>Building URI
* <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#mvc-config>Configure Spring MVC

**Maven**

* <https://maven.apache.org/what-is-maven.html> Maven introduction
* <https://maven.apache.org/plugins/index.html> Maven Plugins
* <https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html> Maven dependency management
* <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html> Maven in 5 minutues

**Spring Boot**

<https://projects.spring.io/spring-boot/> Spring Boot Project Home <http://docs.spring.io/spring-boot/docs/1.4.2.RELEASE/reference/htmlsingle/#getting-started-introducing-spring-boot> Spring Boot Overview <http://docs.spring.io/spring-boot/docs/1.4.2.RELEASE/reference/htmlsingle/#getting-started-first-application> Spring Boot Tutorial <http://docs.spring.io/spring-boot/docs/1.4.2.RELEASE/reference/htmlsingle/#using-boot> Using Spring Boot

**Spring Cloud**

* <http://projects.spring.io/spring-cloud/> Spring Cloud Home
* <https://cloud.spring.io/spring-cloud-netflix/> Spring Cloud Netflix
* <http://techblog.netflix.com/2015/10/evolution-of-open-source-at-netflix.html> Open Source at Netflix
* <https://netflix.github.io/> Netflix OpenSource Center

**WebSocket**

* <https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API> WebSocket API overview
* <https://www.html5rocks.com/en/tutorials/websockets/basics/> WebSocket tutorial
* <https://github.com/sockjs> SockJS
* <http://jmesnil.net/stomp-websocket/doc/> STOMP over WebSocket

**Microservices**

* <http://www.martinfowler.com/articles/microservices.html> Martin Fowler about Microservices
* <http://microservices.io/patterns/microservices.html> Microservice Architecture
* <https://www.nginx.com/blog/introduction-to-microservices/> Introduction to Microservices

**12 Factor App**

* <https://12factor.net/> 12 Factor Manifesto
* <http://www.clearlytech.com/2014/01/04/12-factor-apps-plain-english/> 12 Factor explained

**Restful API**

* <http://job.xdnice.com/content/BiShiJingYan/2012-11/3992.htm> HTTP介绍
* <http://blog.csdn.net/tuantuanls/article/details/41288311> Web Service介绍
* <https://github.com/WhiteHouse/api-standards> Restful API设计原则

**NoSQL**

* <http://www.zhihu.com/question/20059632> NoSQL vs 关系型数据库
* <http://www.zhihu.com/question/23602133> 知乎经典问答

**MongoDB**

* <https://university.mongodb.com/> MongoDB官方教程
* <http://www.slideshare.net/spf13/replication-durability-and-disaster-recovery> 数据库的复制与容错

**Redis**

* <http://www.zhihu.com/question/20795043> redis怎么做消息队列？
* <http://www.zhihu.com/question/27738066> 如何用redis/memcache做Mysql缓存层?
* <http://blog.nosqlfan.com/html/4147.html> Twemproxy – Twitter 开源的 Redis proxy

**Docker**

* <http://dockone.io/article/101> Docker入门教程
* <https://zhuanlan.zhihu.com/p/19902938> 分布式机器学习的故事：Docker改变世界
* <http://dockone.io/article/126> 八个Docker的真实应用场景
* <http://metaphor.space/2016/01/07/docker-a-metaphor/> docker——从隐喻说起

**Cassandra**

* <https://www.ibm.com/developerworks/cn/opensource/os-cn-cassandra/> Cassandra 入门
* <http://www.slideshare.net/benjaminblack/introduction-to-cassandra-replication-and-consistency>
* <http://blog.csdn.net/duxingxia356/article/details/43992015> 概念理解（CAP,BASE, ACID）