

# Spis treści

1. JAVA.....	1
2. SQL i NoSQL.....	1
3. Frameworks (SPRING, HIBERNATE).....	1
3. OTHERS.....	2

## 1. JAVA

### Kurs Java dla początkujących

<https://nullpointerexception.pl/kurs-java-dla-poczatkujacych-1-wprowadzenie/>

### Zastosowanie Stream API z Java 8

<https://justjoin.it/blog/zastosowanie-stream-api-z-java-8-przyklady>

### Wielowątkowość - Future i CompletableFuture

<https://www.youtube.com/watch?v=xlkSa44Mdz8>

### Wątki w języku Java (Thread and Runnable)

<https://www.samouczekprogramisty.pl/watki-w-jezyku-java/>

### Algorytm fizzbuzz

<https://stormit.pl/fizzbuzz/>

### Try-with-resources

<https://www.samouczekprogramisty.pl/konstrukcja-try-with-resources-w-jezyku-java/>

### Jakie znamy wzorce do współbieżności w Javie

<https://howtointerview.pl/pytania-rekrutacyjne/jakie-znamy-wzorce-do-wspolbieznosci-w-javie/10488/>

### Interfejsy funkcyjne w Javie

<https://javappa.com/kurs-java/java8/przeglad-interfejsow-funkcyjnych>

### Garbage Collector – zarządzanie pamięcią

<https://softwareskill.pl/jak-dziala-garbage-collector-zarzadzanie-pamiecia-jvm>

## 2. SQL i NoSQL

### SQL Bez Tajemnic - Pytania i Odpowiedzi

<https://datacraze.pl/sql-bez-tajemnic-pytania-rekrutacyjne>

### SQL vs NoSQL

<https://www.ovhcloud.com/pl/learn/sql-vs-nosql/>

## 3. Frameworks (SPRING, HIBERNATE)

### Spring Framework

<https://javappa.com/kurs-spring/spring-framework>

### Spring Boot | ResponseEntity vs Object -> When to Use ResponseEntity<?> ?

<https://medium.com/@rahul302619/spring-boot-responseentity-vs-object-d530695a8481>

### **TransactionManager Spring Boot**

<https://medium.com/@Gotit.GotIT/mastering-transactions-in-spring-boot-with-platformtransactionmanager-a-practical-guide-4f51bb9e8e42>  
<https://www.youtube.com/watch?v=wDMMD2pYNmg>

### **Hibernate – najpopularniejsze pytania z rozmów kwalifikacyjnych**

<https://codebykris.com/hibernate-najpopularniejsze-pytania/>

### **Zakres Beanów w Spring Boot**

<https://javappa.com/kurs-spring/zakresy-beanow>

### **Jak działa rest controller w springu - głębiej**

<https://medium.com/@viranthrocky/how-the-restcontrollers-are-working-internally-in-spring-boot-487c3ed370b0>

### **Cała Prawda o Spring IoC Containers**

<https://www.youtube.com/watch?v=pYoKGbf8fm8>  
<https://bykowski.pl/spring-context-jak-dziala-kontener-ioc/docke>

### **Hibernate: Sequence vs Identity**

<https://www.geeksforgeeks.org/sql-server/difference-between-sequence-and-identity-in-sql-server/>

### **Kurs Spring Boot Mateusz Dąbrowski**

<https://www.youtube.com/playlist?list=PLLIgVI2WVN6ugud2cc3OShwWoTt65jzSL>

### **Kurs Spring Boot #08 - Spring Cache**

<https://www.youtube.com/watch?v=IWv3uBLO2LU&list=PLLIgVI2WVN6ugud2cc3OShwWoTt65jzSL&index=10>

### **Spring AOP - Implementacja aspektów**

<https://www.youtube.com/watch?v=vVwtkYdpSec&t=1539s>  
<https://nullpointerexception.pl/jak-korzystac-ze-spring-aop/>  
<https://reflectoring.io/aop-spring/>

### **Hibernate i JPA różnica:**

<https://clockworkjava.pl/2020/11/jpa-orm-jpa-i-hibernate/>

### **Spring Data JPA - projekcja danych w zapytaniach dynamicznych**

<https://walczak.it/pl/blog/spring-data-jpa-projekcja-danych-zapytaniach-dynamicznych>

### **Spring Boot: N+1 Problem**

<https://marcinszewczyk.pl/problem-n-plus-1/>  
<https://medium.com/@ayoubseiddiki132/what-is-the-n-1-query-problem-a56e39774cf7>  
<https://nullpointerexception.pl/hibernate-i-problem-n-plus-1-zapytan/>

### **The best way to use JOIN FETCH and Pagination with Spring**

<https://vladmihalcea.com/join-fetch-pagination-spring/>

### **How Projections Affect Pagination in Spring Data JPA**

<http://medium.com/@neharathig/how-projections-affect-pagination-in-spring-data-jpa-c1ecd8475e53>

## **3. OTHERS**

### **Spring Boot Microservices with RabbitMQ Example:**

<https://rameshfadatare.medium.com/spring-boot-microservices-with-rabbitmq-example-92a38cbe08fc>  
<https://medium.com/@sachinmewar800/rabbitmq-with-spring-amqp-using-spring-boot-4a62e68a6f1b>  
<https://github.com/mzerek/RabbitMqExample/tree/main>

**Jak przygotować się do rekrutacji na programistę:**

<https://www.youtube.com/watch?v=6Se64Wg1qQQ>

**Co powinien umieć senior?**

<https://www.youtube.com/watch?v=tdgWeF7XVyw>

**Servlety w aplikacjach webowych**

<https://www.samouczekprogramisty.pl/servlety-w-aplikacjach-webowych/>

**Wzorce Projektowe - Observer**

[https://www.youtube.com/watch?v=gFevma4a\\_qQ](https://www.youtube.com/watch?v=gFevma4a_qQ)

**Przemysław Bykowski**

<https://github.com/bykowski>

**Wzorce projektowe**

<https://refactoring.guru/pl/design-patterns>

**Wzorzec dekorator**

<https://www.samouczekprogramisty.pl/wzorzec-projektowy-dekorator/>

**[Kurs Dockera #03] Docker Networks – Jak działają sieci w Dockerze?**

<https://www.youtube.com/watch?v=-XvpLnWFMaI>

**[Kurs Dockera #04] Docker Ports – Mapowanie i konfiguracja portów**

<https://www.youtube.com/watch?v=j5Lj6XkEk70>

**[Kurs Dockera #05] Docker Volumes – Jak zarządzać danymi w kontenerach?**

<https://www.youtube.com/watch?v=SqMyER2GN5s>

**[Kurs Dockera #06] Docker Compose – Sieci i konfiguracja krok po kroku**

<https://www.youtube.com/watch?v=5iPq4YtkLkM>

**[Kurs Dockera #07] Docker Compose – Ports i użycie w developmentcie**

<https://www.youtube.com/watch?v=thQ8hWvYa00>

**[Kurs Dockera #08] Jak tworzyć własne obrazy Dockerowe - Podstawy pracy z Dockerfile**

<https://www.youtube.com/watch?v=ji8nC2z3wpq>

**[Kurs Dockera #09] Dockerfile Multistage Build – Optymalizacja obrazów**

<https://www.youtube.com/watch?v=La9BN8MAFRk>

**Bykowski blog**

<https://bykowski.pl/blog/>

**Microservice Orchestration vs Choreography – what's the difference?**

<https://camunda.com/blog/2023/02/orchestration-vs-choreography/>

**Circuit Breaker Pattern (Bezpiecznik)**

<https://softwareskill.pl/circuit-breaker-pattern>

**Jak nie zgubić zdarzenia, czyli wzorce Outbox i Inbox**

<https://oskar-dudycz.netlify.app/en/jak-nie-zgubic-zdarzenia-czyli-outbox-i-inbox-w-praktyce/>

**Komunikacja między mikroserwisami**

<https://grzegorzdziedzic.pl/2020/07/19/komunikacja-miedzy-mikroserwisami/>

**Apache Kafka - opis działania i zastosowania**

<https://bulldogjob.pl/readme/apache-kafka-opis-dzialania-i-zastosowania>

example: <https://github.com/dsu/kafka-streams-redis-lambda-poc/tree/main>

example: <https://github.com/dsu/kafka-docker-subscription/tree/master>

**OWASP – co to jest?**

<https://semcore.pl/co-to-jest-owasp-wyjasnienie-i-omowienie-owasp-top-10/>

<https://owasp.org/www-project-top-ten/>