



**Bucket4j**

**Throttle all the things!**

# Bucket4j 7.0.0-beta-2020-04-18

## Reference

### Table of Contents

About the author .....	1
Quick start examples .....	1
Add dependency .....	1
Create your first Bucket .....	2
Section 2 .....	2
About Bucket4j .....	2
What is Bucket4j .....	2
Bucket4j features .....	2
Quick start examples .....	3
Add dependency .....	3
Create your first Bucket .....	3
Section 2 .....	3
Distributed facilities .....	3
Math model explanation .....	4
Release notes bla-bla 7.0.0-beta-2020-04-18 .....	4

### About the author

#### Vladimir Bukhtoyarov

Lead Java developer at Dino Systems/RingCentral

Saint-Petersburg, Russia

[jsecoder@mail.ru](mailto:jsecoder@mail.ru)



### Quick start examples

#### Add dependency

The Bucket4j is distributed through [Maven Central](<https://mvnrepository.com/artifact/com.github.vladimir-bukhtoyarov/bucket4j-core>)

```
<dependency>
  <groupId>com.github.vladimir-bukhtoyarov</groupId>
  <artifactId>bucket4j-core</artifactId>
  <version>7.0.0-beta-2020-04-18</version>
</dependency>
```

```
implementation 'com.github.vladimir-bukhtoyarov:bucket4j-core:7.0.0-beta-2020-04-18'
```

## Create your first Bucket

### Section 2

TODO

## About Bucket4j

### What is Bucket4j

Bucket4j is Java rate-limiting library is mainly based on token-bucket algorithm, which are by default standard for rate limiting in the IT industry.



*Bucket4j is more than direct implementation of token-bucket*

Its math model provides several useful extensions that are not mentioned in the classic token-bucket interpretations, such as multiple limits per bucket or overdraft. These math extensions will be detailed described later.

### Bucket4j features

- **Absolutely non-compromise precision** - Bucket4j does not operate with floats or doubles, all calculation are performed in the integer arithmetic, this feature protects end users from calculation errors involved by rounding.
- **Effective implementation in terms of concurrency:**
  - Bucket4j is good scalable for multi-threading case it by defaults uses lock-free implementation.
  - In same time, library provides different concurrency strategies that can be chosen when default lock-free strategy is not desired.
- **Effective API in terms of garbage collector footprint:** Bucket4j API tries to use primitive types as much as it is possible in order to avoid boxing and other types of floating garbage.

- **Ability to switch from one JVM to cluster** - using Bucket4j you are able to limiting something in the cluster of JVMs:
  - Bucket4j out of the box supports any GRID solution which compatible with JCache API (JSR 107) specification.
  - Bucket4j provides the framework that allows to quickly build integration with your own persistent technology like RDMS or a key-value storage.
  - For clustered usage scenarios Bucket4j supports asynchronous API that extremely matters when going to distribute world, because asynchronous API allows avoiding blocking your application threads each time when you need to execute Network request.
- **Pluggable listener API** that allows to implement monitoring and logging.
- **Rich diagnostic API** that allows to investigate internal state.
- **Rich configuration management** - configuration of the bucket can be changed on fly

## Quick start examples

### Add dependency

The Bucket4j is distributed through [Maven Central](<https://mvnrepository.com/artifact/com.github.vladimir-bukhtoyarov/bucket4j-core>)

*Maven dependency*

```
<dependency>
  <groupId>com.github.vladimir-bukhtoyarov</groupId>
  <artifactId>bucket4j-core</artifactId>
  <version>7.0.0-beta-2020-04-18</version>
</dependency>
```

*Gradle dependency*

```
implementation 'com.github.vladimir-bukhtoyarov:bucket4j-core:7.0.0-beta-2020-04-18'
```

## Create your first Bucket

### Section 2

TODO

## Distributed facilities

o-lo-lo

# Math model explanation

Math model of...

## Release notes bla-bla 7.0.0-beta-2020-04-18

Release notes bla-bla