



# Red Hat Data Grid 8.3

## Data Grid Code Tutorials

Learn how to use Data Grid capabilities



## Red Hat Data Grid 8.3 Data Grid Code Tutorials

---

Learn how to use Data Grid capabilities

## Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Run code tutorials for remote caches and embedded caches that demonstrate various Data Grid capabilities and usage patterns.

---

## Table of Contents

<b>RED HAT DATA GRID</b> .....	<b>3</b>
<b>DATA GRID DOCUMENTATION</b> .....	<b>4</b>
<b>DATA GRID DOWNLOADS</b> .....	<b>5</b>
<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>6</b>
<b>CHAPTER 1. REMOTE CACHES</b> .....	<b>7</b>
1.1. REMOTE CACHE TUTORIALS	7
1.2. HOT ROD JAVA CLIENT TUTORIALS	7
Data Grid documentation	8
<b>CHAPTER 2. EMBEDDED CACHES</b> .....	<b>9</b>
2.1. EMBEDDED CACHE TUTORIALS	9
Data Grid documentation	9
<b>CHAPTER 3. SPRING AND SPRING BOOT</b> .....	<b>11</b>
3.1. SPRING AND SPRING BOOT TUTORIALS	11
Data Grid documentation	12



# RED HAT DATA GRID

Data Grid is a high-performance, distributed in-memory data store.

## **Schemaless data structure**

Flexibility to store different objects as key-value pairs.

## **Grid-based data storage**

Designed to distribute and replicate data across clusters.

## **Elastic scaling**

Dynamically adjust the number of nodes to meet demand without service disruption.

## **Data interoperability**

Store, retrieve, and query data in the grid from different endpoints.

## DATA GRID DOCUMENTATION

Documentation for Data Grid is available on the Red Hat customer portal.

- [Data Grid 8.3 Documentation](#)
- [Data Grid 8.3 Component Details](#)
- [Supported Configurations for Data Grid 8.3](#)
- [Data Grid 8 Feature Support](#)
- [Data Grid Deprecated Features and Functionality](#)



## DATA GRID DOWNLOADS

Access the [Data Grid Software Downloads](#) on the Red Hat customer portal.



### NOTE

You must have a Red Hat account to access and download Data Grid software.

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

## CHAPTER 1. REMOTE CACHES

Deploy multiple Data Grid Server instances to create remote cache clusters that give you a fault-tolerant and scalable data tier with high-speed access from Hot Rod and REST clients.

### 1.1. REMOTE CACHE TUTORIALS

To run these tutorials you need at least one locally running instance of Data Grid Server.

You can [download](#) the distribution and run the following commands:

```
$ ./bin/cli.sh user create admin -p "password"
$ ./bin/server.sh
```



#### NOTE

Data Grid Server enables authentication and authorization by default. Creating a user named **admin** gives you administrative access to Infinispan Server.

#### Building and running remote cache tutorials

You can build and run remote cache tutorials directly in your IDE or from the command line as follows:

```
$ mvn -s /path/to/maven-settings.xml clean package exec:exec
```

### 1.2. HOT ROD JAVA CLIENT TUTORIALS

- Hot Rod Java clients require JDK 8 or later. However, Data Grid recommends using Java 11 at a minimum.

Tutorial link	Description
<a href="#">Remote cache use example</a>	The simplest code example that demonstrates how a remote distributed cache works.
<a href="#">Per cache configuration</a>	Demonstrates how to configure caches dynamically when we connect to the Data Grid Server.
<a href="#">Near caches</a>	Demonstrates how configure near caching to improve the read performance in remote caches.
<a href="#">Cache Admin API</a>	Demonstrates how to use the Administration API to create caches and cache templates dynamically.
<a href="#">Encoding</a>	Demonstrates how encoding of caches work.
<a href="#">Client listeners</a>	Detect when data changes in a remote cache with Client Listeners.

Tutorial link	Description
<a href="#">Query</a>	Demonstrates how to query remote cache values.
<a href="#">Continuous query</a>	Demonstrates how to use Continuous Query and remote caches.
<a href="#">Transactions</a>	Demonstrates how remote transactions work.
<a href="#">Secured caches</a>	Demonstrates how to configure caches that have authorization enabled.
<a href="#">TLS authorization</a>	Demonstrates how to connect to Data Grid Server with TLS authorization.
<a href="#">Counters</a>	Demonstrates how remote counters work.
<a href="#">Multimap</a>	Demonstrates how remote multimap works.
<a href="#">Task execution</a>	Demonstrates how to register server tasks and how to execute them from the Hot Rod client.
<a href="#">JUnit 5 and Testcontainers</a>	Demonstrates how to use the Data Grid and JUnit 5 extension.
<a href="#">Persistence</a>	Demonstrates how to use the Data Grid and persistent caches.

## Data Grid documentation

You can find more resources for Hot Rod Java clients in our documentation at:

- [Hot Rod Java client guide](#)
- [Marshalling and Encoding Data Guide](#)
- [Querying Data Grid caches](#)
- [Developer guide](#)
- [REST API](#)

## CHAPTER 2. EMBEDDED CACHES

Add Data Grid as a dependency to your Java project and use embedded caches that increase application performance and give you capabilities to handle complex use cases.

### 2.1. EMBEDDED CACHE TUTORIALS

You can run embedded cache tutorials directly in your IDE or from the command line as follows:

```
$ mvn -s /path/to/maven-settings.xml clean package exec:exec
```

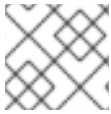
Tutorial link	Description
<a href="#">Distributed caches</a>	Demonstrates how Distributed Caches work.
<a href="#">Replicated caches</a>	Demonstrates how Replicated Caches work.
<a href="#">Invalidated caches</a>	Demonstrates how Invalidated Caches work.
<a href="#">Transactions</a>	Demonstrates how transactions work.
<a href="#">Streams</a>	Demonstrates how Distributed Streams work.
<a href="#">JCache integration</a>	Demonstrates how JCache works.
<a href="#">Functional Maps</a>	Demonstrates how Functional Map API works.
<a href="#">Map API</a>	Demonstrates how the Map API works with Data Grid caches.
<a href="#">Multimap</a>	Demonstrates how to use Multimap.
<a href="#">Queries</a>	Uses Data Grid Query to perform full-text queries on cache values.
<a href="#">Clustered Listeners</a>	Detects when data changes in an embedded cache with Clustered Listeners.
<a href="#">Counters</a>	Demonstrates how to use an embedded Clustered Counter.
<a href="#">Clustered Locks</a>	Demonstrates how to use an embedded Clustered Lock.
<a href="#">Clustered execution</a>	Demonstrates how to use an embedded Clustered Counter.

You can find more resources about embedded caches in our documentation at:

- [Embedding Data Grid Caches](#)
- [Data Grid Developer Guide](#)
- [Querying Data Grid caches](#)

## CHAPTER 3. SPRING AND SPRING BOOT

### 3.1. SPRING AND SPRING BOOT TUTORIALS



#### NOTE

These code tutorials use Data Grid Server and require at least one running instance.

#### Run Spring examples

```
$ mvn -s /path/to/maven-settings.xml package exec:exec
```

#### Run Spring Boot examples

```
$ mvn -s /path/to/maven-settings.xml spring-boot:run
```

#### Displaying actuator statistics

Navigate to <http://localhost:8080/actuator/metrics> in your browser to display a list of available metrics. Cache metrics are prefixed with "cache." Display each metric for each cache using tags. For example for the 'puts' stats in the basque-names cache:

<http://localhost:8080/actuator/metrics/cache.puts?tag=name:basque-names>

#### Collecting statistics with Prometheus

The `prometheus.yml` file in this project contains a `host.docker.internal` binding that allows Prometheus to scrap metrics that the Spring actuator exposes.

Change the **YOUR\_PATH** value in the following command to the directory where Prometheus is running and then run:

#### Podman

```
$ podman run -d --name=prometheus -p 9090:9090 -v YOUR_PATH/integrations/spring-
boot/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus --
config.file=/etc/prometheus/prometheus.yml
```

Tutorial link	Description
<a href="#">Spring Boot and Spring Cache remote mode</a>	Demonstrates how to use Spring Caches with Spring Boot and the Data Grid Server.
<a href="#">Spring Boot and Spring Session remote mode</a>	Demonstrates how to use Spring Session with Spring Boot and the Data Grid Server.
<a href="#">Spring Boot and Spring Cache embedded mode</a>	Demonstrates how to use Spring Caches with Spring Boot and Data Grid Embedded.

Tutorial link	Description
<a href="#">Spring Boot and Spring Session embedded mode</a>	Demonstrates how to use Spring Session with Spring Boot and Data Grid Embedded.
<a href="#">Spring cache embedded without Spring Boot</a>	Demonstrates how to use Spring Cache and Data Grid Embedded without Spring Boot.
<a href="#">Spring session embedded without Spring Boot</a>	Demonstrates how to use Spring Session and Data Grid Embedded without Spring Boot.

### Data Grid documentation

You can find more resources in our documentation at:

- [Using Data Grid with Spring](#)
- [Data Grid Spring Boot Starter](#)