



## Red Hat Data Grid 8.3

# Hot Rod C++ クライアントガイド

Hot Rod C++ クライアントの設定および使用





## 法律上の通知

Copyright © 2023 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.

## 概要

Hot Rod .NET/C# クライアントを使用すると、C# ランタイムアプリケーションがリモートの Data Grid クラスターに接続し、対話できます。

---

## 目次

RED HAT DATA GRID .....	3
DATA GRID のドキュメント .....	4
DATA GRID のダウンロード .....	5
多様性を受け入れるオープンソースの強化 .....	6
第1章 HOT ROD C++ クライアントのインストール .....	7
1.1. C++ コンパイラ要件 .....	7
1.2. RED HAT ENTERPRISE LINUX(RHEL) への HOT ROD C++ クライアントのインストール .....	7
第2章 PROTOBUF スキーマのコンパイル .....	8
2.1. RED HAT ENTERPRISE LINUX(RHEL) での PROTOBUF スキーマのコンパイル .....	8
第3章 HOT ROD C++ クライアントの設定 .....	9
3.1. 設定およびリモートキャッシュマネージャー API .....	9



---

# RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリーデータストアです。

## スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

## グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

## エラスティックスケールリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

## データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

## DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- [Data Grid 8.3 ドキュメント](#)
- [Data Grid 8.3 コンポーネントの詳細](#)
- [Data Grid 8.3 でサポートされる設定](#)
- [Data Grid 8 機能のサポート](#)
- [Data Grid で非推奨の機能](#)

## DATA GRID のダウンロード

Red Hat カスタマーポータルで [Data Grid Software Downloads](#) にアクセスします。



### 注記

Data Grid ソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

## 第1章 HOT ROD C++ クライアントのインストール

ホストシステムに Hot Rod C++ クライアントを動的ライブラリーとしてインストールします。

### 1.1. C++ コンパイラ要件

オペレーティングシステム	必須コンパイラ
Red Hat Enterprise Linux(RHEL)7、64 ビット	C++ 11 コンパイラ (GCC 4.8.1)
RHEL 8、64 ビット	C++ 11 コンパイラ (GCC 4.8.1)

### 1.2. RED HAT ENTERPRISE LINUX(RHEL) への HOT ROD C++ クライアントのインストール

Data Grid は、RHEL 用の Hot Rod C++ クライアントの RPM ディストリビューションを提供します。

#### 手順

1. RHEL で Hot Rod C++ クライアントのリポジトリを有効にします。

RHEL のバージョン	リポジトリ
RHEL 7	<b>jb-datagrid-8.1-for-rhel-7-server-rpms</b>
RHEL 8	<b>jb-datagrid-8.1-for-rhel-8-x86_64-rpms</b>

2. Hot Rod C++ クライアントをインストールします。

```
# yum install jdgcpp-client
```

#### 関連情報

- [Enabling or disabling a repository using Red Hat Subscription Management](#) (ナレッジベース)
- [Red Hat パッケージブラウザー](#)

## 第2章 PROTOBUF スキーマのコンパイル

Data Grid は ProtoStream API を使用して、データを Protobuf でエンコードされたエントリーとして保存します。

Protobuf は、Hot Rod エンドポイントと REST エンドポイントの両方を使用して、クライアントがリモートキャッシュでエントリーを作成および取得できるようにする言語に依存しない形式です。

### 2.1. RED HAT ENTERPRISE LINUX(RHEL) での PROTOBUF スキーマのコンパイル

Protobuf スキーマ **.proto** ファイルを C++ ヘッダーおよびソースファイルにコンパイルし、データを Data Grid に記述します。

#### 前提条件

- Protobuf ライブラリーおよび **protobuf-devel** パッケージをインストールします。

```
# yum install protobuf
# yum install protobuf-devel
```

#### 手順

1. **LD\_LIBRARY\_PATH** 環境変数が設定されていない場合は設定します。

```
# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/lib64
```

2. 必要に応じて、Hot Rod C++ クライアントの Protobuf スキーマをコンパイルします。

```
# /bin/protoc --cpp_out dllexport_decl=HR_PROTO_EXPORT:/path/to/output/ $FILE
```

**HR\_PROTO\_EXPORT** は、Protobuf スキーマをコンパイルする際に Hot Rod C++ クライアントが拡張するマクロです。

3. クエリーを使用する予定の場合は、Protobuf スキーマを Data Grid に登録します。

#### 関連情報

- [Protobuf スキーマの登録](#)

## 第3章 HOT ROD C++ クライアントの設定

Hot Rod C++ クライアントは、**RemoteCache** API 経由でリモート Data Grid クラスターと対話します。

### 3.1. 設定およびリモートキャッシュマネージャー API

**ConfigurationBuilder** API を使用して、Hot Rod C++ クライアント接続と **RemoteCacheManager** API を設定してリモートキャッシュを取得および設定します。

#### 設定ビルダー

```
#include "infinispan/hotrod/ConfigurationBuilder.h"
#include "infinispan/hotrod/RemoteCacheManager.h"
#include <infinispan/hotrod/RemoteCache.h>
#include <iostream>
int main () {
    ConfigurationBuilder builder;
    // Configure a cache manager to connect with Hot Rod version 2.8
    builder.protocolVersion(Configuration::PROTOCOL_VERSION_28);
    // Connect to a server at localhost with the default port.
    builder.addServer().host("127.0.0.1").port(11222);
    // Create and start a RemoteCacheManager to interact with caches.
    RemoteCacheManager cacheManager(builder.build(), false);
    cacheManager.start();
    ...
}
```

#### クロスサイトレプリケーション

```
ConfigurationBuilder builder;
builder.addServer().host("127.0.0.1").port(11222);
// Configure a remote cluster and node when using cross-site replication.
builder.addCluster("NYC").addClusterNode("192.0.2.0", 11322);
```

#### ニアキャッシュ

```
ConfigurationBuilder builder;
builder.addServer().host("127.0.0.1").port(11222);
// Enable near-caching for the client.
builder.nearCache().mode(NearCacheMode::INVALIDATED).maxEntries(4);
```

#### 関連情報

- [Hot Rod C++ client API](#)