



Red Hat JBoss Enterprise Application Platform 7.3

JBoss EAP の紹介

サブシステムや操作モードなど、一般的な Red Hat JBoss Enterprise Application Platform の概念の説明

Red Hat JBoss Enterprise Application Platform 7.3 JBoss EAP の紹介

サブシステムや操作モードなど、一般的な Red Hat JBoss Enterprise Application Platform の概念の説明

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

本ドキュメントでは、Red Hat JBoss Enterprise Application Platform (JBoss EAP) の全般的な概要を説明します。また、本書は JBoss EAP のサブシステムや、異なる操作モードについても紹介します。

目次

第1章 一般概念の概要	3
1.1. JAVA	3
1.2. アプリケーションサーバー	3
1.3. JAKARTA EE 8	3
1.4. JAVA ENTERPRISE EDITION 8	3
1.5. JAVA EE から JAKARTA EE への移行	4
1.6. JBOSS EAP と対応する JAKARTA EE 仕様の JAVA EE 仕様	4
第2章 JBOSS EAP の概要	6
2.1. JBOSS EAP 7 について	6
2.2. サブシステム	7
2.3. 高可用性	7
2.4. 操作モード	7
第3章 例	8
3.1. 簡単な例	8
3.2. 広範な例	8

第1章 一般概念の概要

Red Hat JBoss Enterprise Application Platform の設定方法やデプロイ方法を理解する前に、重要な概念の一部を理解する必要があります。

1.1. JAVA

Java は、オブジェクト指向、クラス、および同時実行などの概念を取り入れたプログラミング言語およびコンピューティングプラットフォームです。Java アプリケーションはバイトコードにコンパイルされ、Java 仮想マシン (JVM) 内で実行されます。

1.2. アプリケーションサーバー

アプリケーションサーバーは、Web アプリケーションを実行する環境を提供するソフトウェアです。また、ほとんどのアプリケーションサーバーは、環境で実行されている web アプリケーションに API で機能を提供します。たとえば、アプリケーションサーバーでは、データベースへ接続するための API を利用できます。

1.3. JAKARTA EE 8

Eclipse Foundation によって管理されている Jakarta EE Platform 8 は、Jakarta EE アプリケーションをホストする標準プラットフォームとなっています。

Jakarta EE Platform 8 の詳細は、[Jakarta EE Platform 8 仕様](#) を参照してください。

1.4. JAVA ENTERPRISE EDITION 8

Java Platform, Enterprise Edition (Java EE) は、Java アプリケーションの実行および開発を行うための API とランタイム環境を両方提供する、標準ベースのエンタープライズプラットフォームです。この目的は、簡単に消費できるフレームワークに豊富なエンタープライズ機能を提供してボイラープレートを排除し、技術的な負担を削減することで、開発者の生産性を向上することです。Java EE を設定するフレームワークの組み合わせは厳しくテストされています。

[JSR 366](#) をベースとする Java EE 8 は Java EE 7 上に構築されます。本リリースでは主に、現在のアプリケーションに必要な API およびプログラミングモデルを向上することに焦点を置きました。Java EE (JSR 366) の仕様の他に、以下の仕様が Java EE で新規導入または更新されました。

- [JSR 250](#): Common Annotations 1.3
- [JSR 338](#): Java Persistence 2.2
- [JSR 356](#): Java API for WebSocket 1.1
- [JSR 365](#): Contexts and Dependency Injection (CDI) 2.0
- [JSR 367](#): The Java API for JSON Binding (JSON-B) 1.0
- [JSR 369](#): Java Servlet 4.0
- [JSR 370](#): Java API for RESTful Web Services (JAX-RS) 2.1
- [JSR 372](#): JavaServer Faces (JSF) 2.3
- [JSR 374](#): Java API for JSON Processing (JSON-P) 1.1

- [JSR 375](#): Java EE Security API 1.0
- [JSR 380](#): Bean Validation 2.0
- [JSR 919](#): JavaMail 1.6

1.5. JAVA EE から JAKARTA EE への移行

Java Enterprise Edition 8 リリース後、Oracle は Java EE を Eclipse Foundation に移しました。段階的な転送プロセスの一環として、API コード、実装コード、およびテクノロジープレビューキット (TCK) コードが移されました。新しい認定プロセス Jakarta EE Specification Process (JESP) が設定され、Eclipse Foundation Technology Compatibility Kit ライセンスの新しい仕様ライセンスが作成されました。

この移行プロセスの一環として、既存の Java EE 仕様に対応するすべての Jakarta 仕様に対して新しい名前が作成されました。すべての新規の名前は Jakarta で始まり、その後に仕様の簡単な説明が続きます。JBoss EAP ドキュメントに記載されている Java EE 仕様名と対応する Jakarta EE 仕様の名前は、Java EE Specifications Relevant for JBoss EAP and the Corresponding Jakarta EE Specifications の項に記載されています。

その他のリソース

[JBoss EAP と対応する Jakarta EE 仕様の Java EE 仕様](#) を参照してください。

1.6. JBOSS EAP と対応する JAKARTA EE 仕様の JAVA EE 仕様

表1.1 JBoss EAP と対応する Jakarta EE 仕様の Java EE 仕様

Java EE 8 仕様	Java EE 8 省略形	Jakarta EE 8 仕様
Java Servlet		Jakarta Servlet
JavaServer Faces	JSF	Jakarta Server Faces
Java API for WebSocket		Jakarta WebSocket
Concurrency Utilities for Java EE		Jakarta Concurrency
Interceptors		Jakarta Interceptors
Java Authentication Service Provider Interface for Containers	JASPIC	Jakarta Authentication
Java Authorization Contract for Containers	JACC	Jakarta Authorization
Java EE Security API		Jakarta Security
Java Message Service	JMS	Jakarta Messaging
Java Persistence API	JPA	Jakarta Persistence

Java EE 8 仕様	Java EE 8 省略形	Jakarta EE 8 仕様
Java Transaction API	JTA	Jakarta Transactions
Batch Applications for the Java Platform	JBatch	Jakarta Batch
JavaMail API		Jakarta Mail
Java EE Connector Architecture	JCA	Jakarta Connectors
Common Annotations for Java Platform		Jakarta Annotations
JavaBeans Activation Framework	JAF	Jakarta Activation
Bean Validation	JBV	Jakarta Bean Validation
Expression Language	JEL	Jakarta Expression Language
Enterprise JavaBeans 3.2	EJB	Jakarta Enterprise Beans
Java Architecture for XML Binding	JAXB	JAXB
Java API for JSON Binding	JSON-B	Jakarta JSON Binding
JavaServer Pages	JSP	Jakarta Server Pages
Java API for XML-Based Web Services	JAX-WS	Jakarta XML Web Services
Java API for RESTful Web Services	JAX-RS	Jakarta RESTful Web Services
JavaServer Pages Standard Tag Library	JSTL	Jakarta Standard Tag Library
Contexts and Dependency Injections	CDI	Jakarta Contexts and Dependency Injection
Java API for JSON Processing	JSON-P	Jakarta JSON Processing
Java API for XML-Based RPC	JAX-RPC	Jakarta XML RPC
Java API for XML Registries	JAXR	Jakarta XML Registries

第2章 JBOSS EAP の概要

2.1. JBOSS EAP 7 について

Red Hat JBoss Enterprise Application Platform (JBoss EAP) 7.3 は、Web Profile と Full Platform 仕様の両方に対応する Jakarta EE 8 対応実装です。また、Java Enterprise Edition (Java EE) 8 仕様の認定実装でもあります。JBoss EAP のメジャーバージョンは、[WildFly](#) コミュニティープロジェクトの機能の完全性が希望のレベルに達した時点で、このコミュニティプロジェクトからフォークされます。その後、テストと製品化が長期間行われ、実稼働向けに安定化、認定、および改良されます。JBoss EAP メジャーバージョンのライフタイムの間、選択された機能はコミュニティプロジェクトからチェリーピックおよびバックポートされ、同じメジャーバージョン内の機能が強化されたマイナーリリースに追加されます。

JBoss EAP は、高可用性クラスターリング、メッセージング、分散キャッシュなど、事前設定されたオプションを機能に提供します。これにより、ユーザーは JBoss EAP が提供するさまざまな API やサービスを使用して、アプリケーションを書き込み、デプロイ、および実行することもできます。

JBoss EAP には、必要時のみにサービスを有効にできるモジュラー後続が含まれているため、起動速度が改善されます。web ベースの管理コンソールと管理コマンドラインインターフェイス (CLI) により、XML 設定ファイルを編集する必要がなく、タスクをスクリプト化および自動化する機能が追加されます。さらに、JBoss EAP には、セキュアでスケーラブルな Java EE アプリケーションを迅速に開発するための API と開発フレームワークが含まれています。

表2.1 JBoss EAP の機能

機能	説明
Jakarta EE 対応	Web Profile と Full Platform 仕様の Jakarta EE 8 対応実装。
Java EE 準拠	Java Enterprise Edition 8 のフルプラットフォームおよび Web プロファイルは認定されています。
管理対象ドメイン	複数のサーバーインスタンスと物理ホストを一元管理し、スタンドアロンサーバーは単一のサーバーインスタンスを許可します。設定、デプロイメント、ソケットバインディング、モジュール、拡張、およびシステムプロパティをサーバーグループごとに管理します。アプリケーションセキュリティ(セキュリティドメインを含む)を一元的および簡単に管理します。
管理コンソールおよび管理 CLI	新しいドメインまたはスタンドアロンサーバー管理インターフェイスです。管理 CLI には、管理タスクをスクリプト化および自動化できるバッチモードも含まれています。JBoss EAP の XML 設定ファイルの直接編集は推奨されません。

機能	説明
簡素化されたディレクトリーのレイアウト	モジュールディレクトリーにはすべてのアプリケーションサーバーモジュールが含まれています。ドメインディレクトリーにはドメインデプロイメントのアーティファクトが含まれ、スタンドアロンディレクトリーにはスタンドアロンデプロイメントの設定ファイルが含まれています。
モジュラークラスローディングの仕組み	モジュールは必要に応じてロードおよびアンロードされます。これにより、パフォーマンスの向上およびセキュリティの強化が実現され、起動および再起動時間が短縮されます。
データソース管理の合理化	データベースドライバーは他のサービスと同様にデプロイされます。さらに、データソースは管理コンソールと管理 CLI を使用して作成および管理されます。
統一されたセキュリティーフレームワーク	Elytron では、スタンドアロンサーバーと管理対象ドメインの両方のアクセスを管理および設定できる単一の統合フレームワークを利用できます。JBoss EAP サーバーにデプロイされたアプリケーションのセキュリティーアクセスを設定するためにも使用できます。

2.2. サブシステム

JBoss EAP にデプロイされたアプリケーションに公開された API と機能の多くは、サブシステムに分けられます。管理者は、アプリケーションの目的に応じてこれらのサブシステムを設定し、異なる動作を提供できます。たとえば、アプリケーションにデータベースが必要な場合、データソースを **datasources** サブシステムで設定することができ、その JBoss EAP サーバーまたはドメインへデプロイされた後に、そのアプリケーションによるアクセスが可能になります。

2.3. 高可用性

JBoss EAP では、高可用性 (HA) とは、複数の JBoss EAP インスタンスが連携し、トラフィックの変動、サーバーの負荷、およびサーバーの障害に対してより堅牢なアプリケーションを提供することを言います。HA はスケーラビリティ、ロードバランシング、およびフォールトトレランスなどの概念を取り入れます。

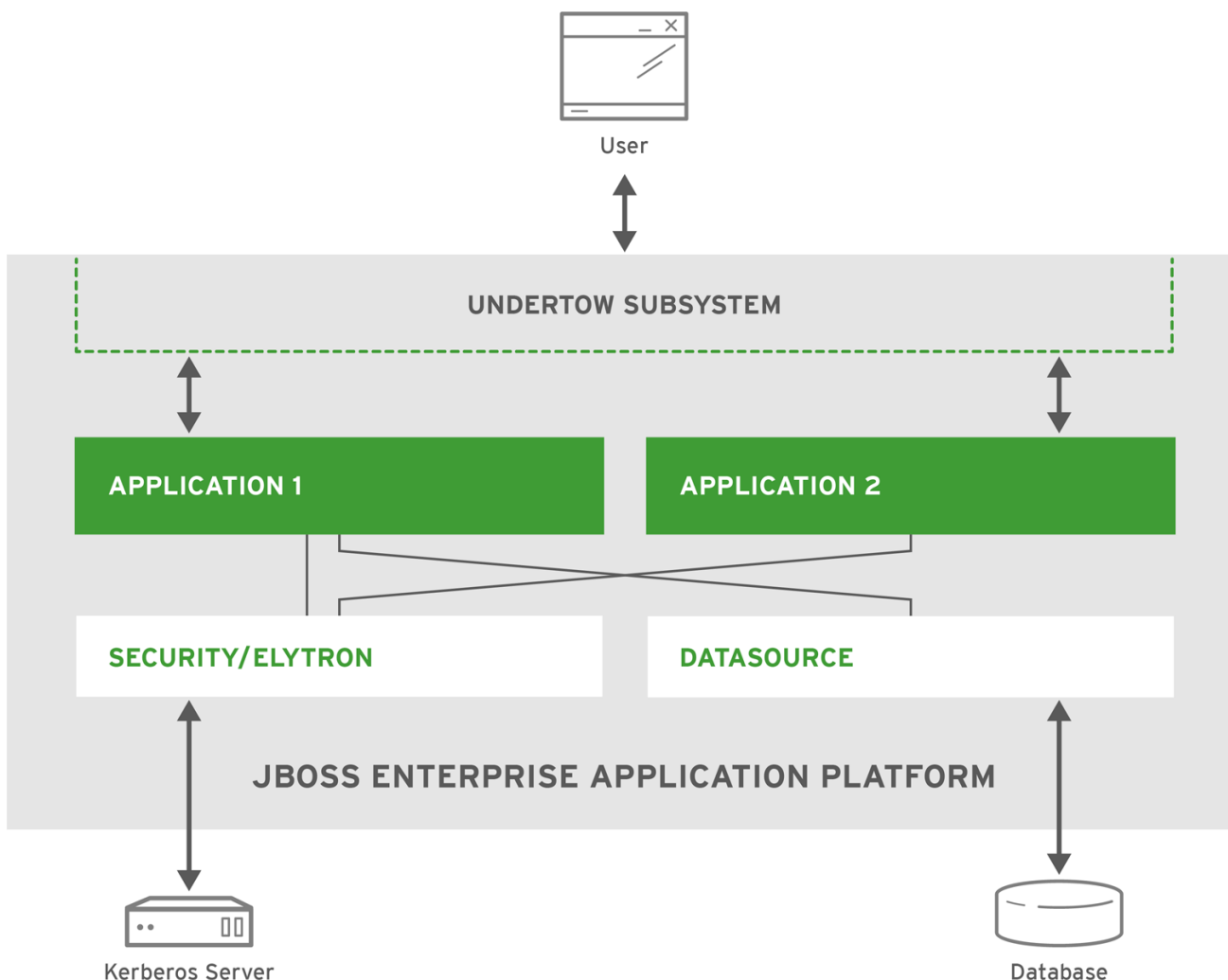
2.4. 操作モード

JBoss EAP は、機能と API をアプリケーションに提供する他に、強力な管理機能も備えています。これらの管理機能は、JBoss EAP の起動に使用される操作モードによって異なります。JBoss EAP は、個別のインスタンスの管理には **スタンドアロンサーバー** 操作モードを提供し、単一制御点からインスタンスのグループを管理するために **管理対象ドメイン** を提供します。

第3章 例

以下の例は、JBoss EAP の動作と対象の環境を表しています。

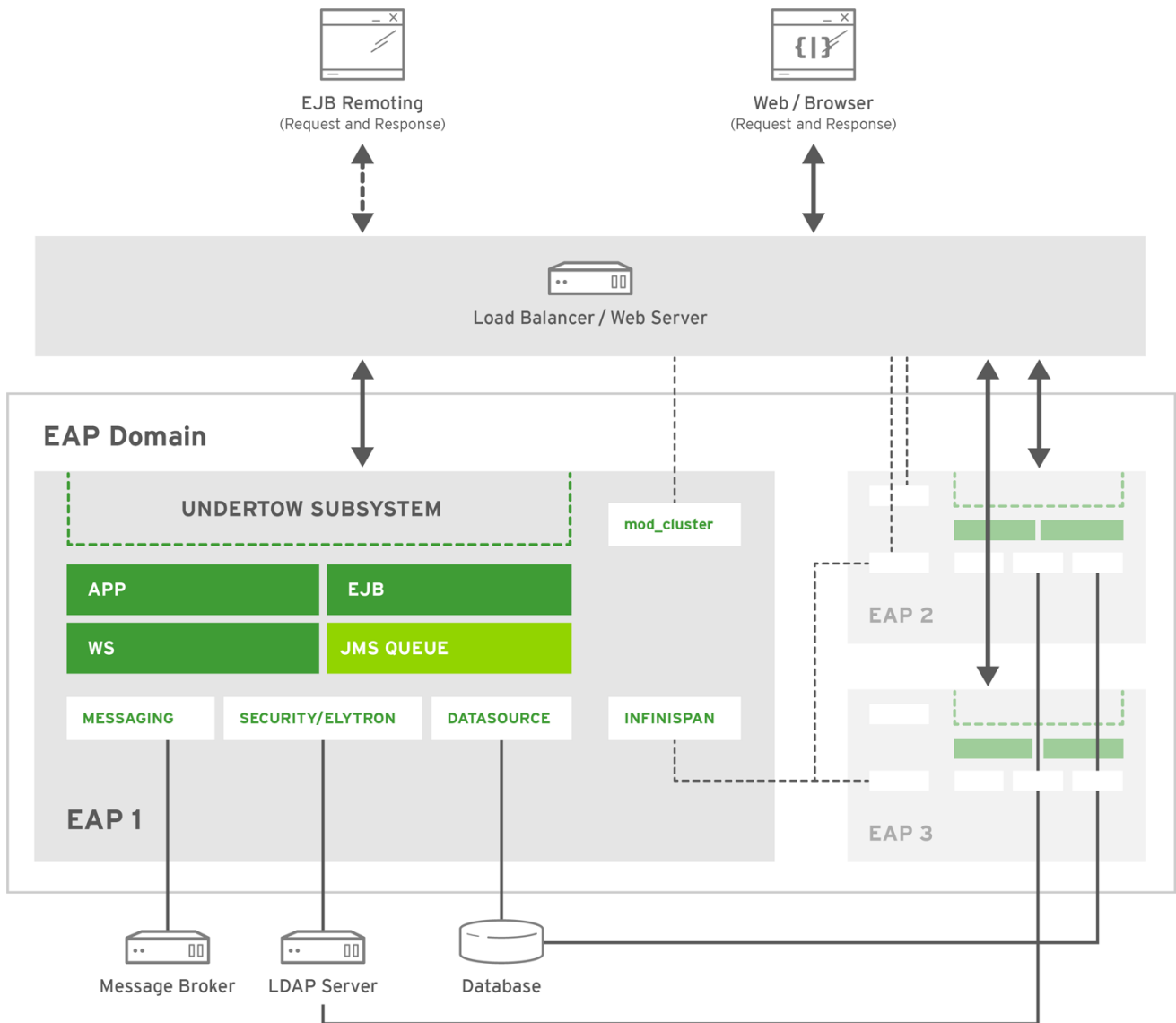
3.1. 簡単な例



JBOSS_430110_1216

以下の例は簡単な JBoss EAP のセットアップを表しています。JBoss EAP インスタンスには 2 つのアプリケーションがデプロイされています。また、**datasources** サブシステムを使用してデータベースに接続するよう設定され、レガシー **security** サブシステムまたは **elytron** サブシステムのいずれかを使用できる Kerberos サーバーに接続するよう設定されています。これらの接続は、デプロイされたアプリケーションに公開されます。JBoss EAP インスタンスは **undertow** サブシステム経由でリクエストを処理し、これらのリクエストを適切なアプリケーションに転送します。アプリケーションは JBoss EAP によって公開される API を使用してデータベースおよび Kerberos サーバーに接続し、実装されたビジネスロジックを実行します。実行完了後、アプリケーションは **undertow** サブシステム経由で応答を要求元に返信します。

3.2. 広範な例



JBOSSE_430110_1216

この例は、ロードバランサーまたは Web サーバーのいずれかを使用する管理対象ドメインの 3 つの JBoss EAP インスタンスが関係するより複雑なセットアップを表しています。また、3 つのインスタンスは、`mod_cluster` を使用するロードバランシングと `Infinispan` を使用するセッションレプリケーションを介して高可用性をサポートするよう設定されています。3 つの JBoss EAP インスタンスにはすべて web アプリケーション、web サービス、および EJB がデプロイされています。1 つの JBoss EAP インスタンスには、**messaging-activemq** サブシステムより設定された JMS キューがあります。3 つの JBoss EAP インスタンスはすべてデータソースを介してデータベースに接続します。また、レガシー **security** サブシステムまたは **elytron** サブシステムのいずれかを使用して LDAP サーバーに接続します。さらに、1 つの JBoss EAP インスタンスは、**messaging-activemq** サブシステム経由で外部メッセージブローカーに接続するよう設定されています。設定済みのこれらの接続は、対応するインスタンスにデプロイされたアプリケーション、Web サービス、EJB、および JMS キューに公開されます。

アプリケーション、Web サービス、または EJB 向けの受信リクエストは、すべて最初にロードバランサーまたは Web サーバーによって受信されます。設定したロードバランシングアルゴリズムと各 JBoss EAP インスタンスによって提供される情報を基に、web サーバーまたはロードバランサーはそのリクエストを適切な JBoss EAP インスタンスに転送します。JBoss EAP インスタンスは **undertow** サブシステム経由でリクエストを処理し、これらのリクエストを適切なアプリケーションに転送します。アプリケーションは JBoss EAP によって公開される API を使用してデータベースおよび Kerberos サーバーに接続し、実装されたビジネスロジックを実行します。実行完了後、アプリケーションは **undertow** サブシステム経由で応答を要求元に返信します。セッション情報などの永続化されない情報は、**infinispan** サブシステム経由で JBoss EAP インスタンスの間で伝搬されます。

Revised on 2023-01-28 12:57:51 +1000