



Red Hat build of Eclipse Vert.x 4.0

Eclipse Vert.x 4.0 リリースノート

Eclipse Vert.x 4.0.3 での使用

Red Hat build of Eclipse Vert.x 4.0 Eclipse Vert.x 4.0 リリースノート

Eclipse Vert.x 4.0.3 での使用

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Release_Notes_for_Eclipse_Vert.x_4.0.ent file |.

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

本リリースノートには、Eclipse Vert.x 4.0.3 に関連する重要な情報が含まれています。

目次

はじめに	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
多様性を受け入れるオープンソースの強化	5
第1章 必要なインフラストラクチャーコンポーネントのバージョン	6
第2章 サポートされる ECLIPSE VERT.X RUNTIME コンポーネント設定および統合	7
第3章 機能	8
3.1. 新機能および変更された機能	8
3.1.1. すべてのタイプのデータをバインドしてより安全になったコンテキストサーバーインターセプター	8
3.1.2. ワイルドカード文字で終わるルートパスの末尾のスラッシュ (/) の一致が不要	8
3.1.3. サービス検出オプションから autoRegistrationOfImporters 属性を削除	9
3.1.4. 入力クレデンシャルとして token をサポートするように認証プロバイダークラスの認証メソッドを更新	9
3.1.5. PEM キーの Get メソッドが String ではなく Buffer を返す	9
3.1.6. Kubernetes サービスインポーターが自動的に登録されなくなる	9
3.1.7. 非同期操作に future メソッドを使用	10
3.1.8. Jackson Databind ライブラリーの依存関係がない	10
3.1.9. 非推奨と削除の処理	10
3.1.10. 分散トレースのサポート	10
3.1.11. EventBus JavaScript Client の新しい公開場所	11
3.1.12. OpenShift Maven プラグインを使用した Eclipse Vert.x アプリケーションのデプロイ	11
3.1.13. OpenShift の Eclipse Vert.x メタリングラベル	11
3.1.14. OpenJDK 8 および OpenJDK 11 RHEL 8 Universal Base Images (UBI8) のサポート	11
3.2. 非推奨の機能	11
第4章 リリースコンポーネント	20
4.1. 本リリースで導入されたサポート対象のアーティファクト	20
4.2. 本リリースで導入されたテクニカルプレビューアーティファクト	20
4.3. 本リリースで削除されたアーティファクト	20
4.4. 本リリースで非推奨となったアーティファクト	20
第5章 修正された問題	21
5.1. GRAPHQL ビルドに含まれる GOOGLE GUAVA クラス	21
5.2. ECLIPSE VERT.X ビルドで利用可能な VERTX-OPENTRACING	21
第6章 既知の問題	22
6.1. KUBERNETESSERVICEIMPORTER() を ECLIPSE VERT.X REACTIVE EXTENSIONS (RX) に直接登録できない	22
6.2. IBM Z および IBM POWER SYSTEMS では、RED HAT AMQ STREAMS イメージが利用できない	22
6.3. TLS プロトコルのバージョンが一致しないため、RHEL 8 ベースのデータベースアプリケーションと RHEL 7 ベースの MYSQL 5.7 データベース間の接続が失敗する	22
6.4. FALSE CONNECTION がアプリケーションエンドポイントを呼び出す際にピアエラーメッセージによってリセットされる	23
第7章 本リリースに関連するアドバイザリー	24

はじめに

リリース日: 2021-03-29

RED HAT ドキュメントへのフィードバック (英語のみ)

弊社のドキュメントに関するご意見やご感想をお寄せください。フィードバックをお寄せいただくには、ドキュメントのテキストを強調表示し、コメントを追加できます。

本セクションでは、フィードバックの送信方法を説明します。

前提条件

- Red Hat カスタマーポータルにログインしている。
- Red Hat カスタマーポータルで、**マルチページ HTML** 形式でドキュメントを表示している。

手順

フィードバックを提供するには、以下の手順を実施します。

1. ドキュメントの右上隅にある **フィードバック** ボタンをクリックして、既存のフィードバックを確認します。



注記

フィードバック機能は、**Multi-page HTML** 形式でのみ有効です。

2. フィードバックを提供するドキュメントのセクションを強調表示します。
3. ハイライトされたテキストの近くに表示される **Add Feedback** ポップアップをクリックします。
ページの右側のフィードバックセクションにテキストボックスが表示されます。
4. テキストボックスにフィードバックを入力し、**Submit** をクリックします。
ドキュメントに関する問題が作成されます。
5. この問題を確認するには、フィードバックビューで問題トラッカーをクリックします。

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 必要なインフラストラクチャーコンポーネントのバージョン

Red Hat は、サポートとして明示的に指定されたコンポーネントを除き、以下のコンポーネントに対するサポートは提供していません。

コンポーネント名	Version
Maven	3.6.0
OpenShift Maven プラグイン	1.1.1
JDK ^{[a][b]}	OpenJDK 8 または OpenJDK 11 ^[c]
Red Hat Enterprise Linux 7 ^[d]	7.9
Red Hat Enterprise Linux 8 ^[e]	8.3
OpenShift Container Platform (OCP) ^[f]	3.11、4.7
Minishift	1.34.3 以降
CDK ^[g]	3.11.0
git	2.0 以降
oc コマンドラインツール	3.11 以降 ^[h]

[a] JRE はソースから Java アプリケーションをコンパイルするツールを提供しないため、完全な JDK インストールが必要です。

[b] Red Hat OpenJDK は、Red Hat によってサポートされます。

[c] Red Hat は、JDK の LTS リリースのみをサポートします。

[d] 実稼働環境のスタンドアロン RHEL に CNR をベースとしたアプリケーションをデプロイする場合。

[e] 実稼働環境のスタンドアロン RHEL に CNR をベースとしたアプリケーションをデプロイする場合。

[f] OCP は Red Hat によってサポートされます

[g] CDK は Red Hat によってサポートされています。

[h] CLI ツール **oc** のバージョンは、使用している OCP のバージョンに対応する必要があります。

第2章 サポートされる ECLIPSE VERT.X RUNTIME コンポーネント設定および統合

以下のリソースは、Eclipse Vert.x と Red Hat 製品のサポートされる設定および統合を定義します。

- 実稼働環境で Eclipse Vert.x との統合でサポートされる技術の一覧は、[「Red Hat Runtimes Supported Configurations」](#) を参照してください。
- Eclipse Vert.x ランタイムアーティファクトおよびそのバージョンの一覧は、[コンポーネントの詳細ページ](#) を参照してください。

第3章 機能

3.1. 新機能および変更された機能

本項では、本リリースで導入された新機能を説明します。また、既存の機能の変更に関する情報も含まれます。

3.1.1. すべてのタイプのデータをバインドしてより安全になったコンテキストサーバーインターセプター

Eclipse Vert.x 4.0.3 以降、**ContextServerInterceptor.bind ()** メソッドはすべてのタイプのデータをコンテキストにバインドします。このメソッドはストレージの詳細を公開しないため、より安全になりました。

Eclipse Vert.x 4.0.3 より前のリリースでは、このメソッドは 'String' データ型のみをコンテキストにバインドしていました。また、ストレージの詳細も公開しました。

更新された **ContextServerInterceptor.bind ()** メソッドを使用するには、アプリケーションを更新する必要があります。

以下の例は、Eclipse Vert.x 4.0.3 より前のリリースのコードを示しています。

```
// Example code from previous releases

class X extends ContextServerInterceptor {
    @Override
    public void bind(Metadata metadata, ConcurrentMap<String, String> context) {
```

次の例は、Eclipse Vert.x 4.0.3 リリースの置換コードを示しています。

```
// Replacing code for Eclipse Vert.x 4.0.3 release

class X extends ContextServerInterceptor {
    @Override
    public void bind(Metadata metadata) {
```

3.1.2. ワイルドカード文字で終わるルートパスの末尾のスラッシュ (/) の一致が不要

Eclipse Vert.x 4.0.3 より前のリリースでは、ルートがスラッシュで終わるパスとワイルドカード/*で定義されている場合、一致するリクエストにも末尾のスラッシュ/が含まれている場合にのみ、ルートが呼び出されました。このルールは、ワイルドカードが空の場合に問題を引き起こしました。

Eclipse Vert.x 4.0.3 以降では、このルールは適用されなくなりました。パスがスラッシュ (/) で終わるルートを作成できます。ただし、リクエスト URL にスラッシュを指定することは必須ではありません。

また、リクエスト URL を作成および使用し、パスにスラッシュ (/) ではなく、ワイルドカードで終わるルートを呼び出すこともできます。たとえば、ワイルドカードが含まれるルートは **/foo/*** として定義できます。ここでは、ルートはパスの最後にあるオープンワイルドカードと一致する必要があります。リクエスト URL は **/foo** にすることができます。

この表は、Eclipse Vert.x 4.0.3 以前のリリースでのリクエスト URL `/foo/*` を送信するときの動作を示しています。Eclipse Vert.x 4.0.3 では終了スラッシュが任意であり、要求はルートに一致することがわかります。

ルート	Eclipse Vert.x 4.0.3	Eclipse Vert.x 4.0.3 より前のリリース
<code>/foo</code>	Match	No Match
<code>/foofighters</code>	No Match	No Match
<code>/foo/</code>	Match	Match
<code>/foo/bar</code>	Match	Match

3.1.3. サービス検出オプションから `autoRegistrationOfImporters` 属性を削除

`AutoRegistrationOfImporters` 属性はサービス検出オプションから削除されました。

3.1.4. 入力クレデンシャルとして `token` をサポートするように認証プロバイダクラスの認証メソッドを更新

Eclipse Vert.x 4.0.3 より前のリリースでは、`AuthenticationProvider.authenticate()` メソッドが入力クレデンシャルとして `jwt: someValue` を誤って取得していました。

Eclipse Vert.x 4.0.3 以降、`AuthenticationProvider.authenticate()` メソッドが更新され、`token: someValue` が入力クレデンシャルとして取得されます。この変更により、JSON API と型指定された API の両方が一貫性を保ち、同じ意味で使用できるようになります。

以下のコードは、Eclipse Vert.x 4.0.3 より前のリリースでの認証メソッドの実装を示しています。

```
new JsonObject().put("jwt", "token...");
```

以下のコードは、Eclipse Vert.x 4.0.3 リリースの認証メソッドの実装を示しています。

```
new JsonObject().put("token", "token...");
```

3.1.5. PEM キーの `Get` メソッドが `String` ではなく `Buffer` を返す

`PubSecKeyOptions.getBuffer()` メソッドは、PEM またはシークレットキーバッファを返します。Eclipse Vert.x 4.0.2 より前のリリースでは、キーバッファが保存され `String` として返されました。ただし、シークレットを `Buffer` として保存することを推奨します。Eclipse Vert.x 4.0.2 以降では、メソッドはキーバッファを保存し、`Buffer` として保存し返します。この変更により、シークレットのセキュリティおよび処理が改善します。

`PubSecKeyOptions.setBuffer()` メソッドは `String` 引数を受け入れます。set メソッドでは、ASCII 以外のシークレット資料を安全に処理するために、バッファのオーバーロードが追加されました。この変更には、既存のコードを変更する必要はありません。

3.1.6. Kubernetes サービスインポーターが自動的に登録されなくなる

Eclipse Vert.x 4 以降、**KubernetesServiceImporter** 検出ブリッジは自動的に登録されなくなりました。Maven プロジェクトのクラスパスにブリッジを追加しても、自動的に登録されません。

ServiceDiscovery インスタンスの作成後にブリッジを手動で登録する必要があります。

3.1.7. 非同期操作に `future` メソッドを使用

Eclipse Vert.x 4 は、非同期操作に `future` を使用します。すべての `callback` メソッドには、対応する `future` メソッドがあります。

`future` は非同期操作の作成に使用できます。`future` を使用する場合は、エラー処理の方が優れています。したがって、アプリケーションでコールバックと `future` の組み合わせを使用することが推奨されません。

3.1.8. Jackson Databind ライブラリーの依存関係がない

Eclipse Vert.x 4 では、Jackson Databind は任意の Maven 依存関係です。この依存関係を使用する場合は、クラスパスに明示的に追加する必要があります。たとえば、オブジェクトマッピング JSON の場合は、依存関係を明示的に追加する必要があります。

3.1.9. 非推奨と削除の処理

Eclipse Vert.x 4 では、新機能が追加されています。以前の機能および機能は Eclipse Vert.x 4 で非推奨または削除されました。アプリケーションを Eclipse Vert.x 4 に移行する前に、非推奨および削除を確認します。

Java コンパイラーは、非推奨の API が使用されたときに警告を生成します。アプリケーションを Eclipse Vert.x 4 に移行する際に、コンパイラーを使用して非推奨のメソッドを確認できます。

3.1.10. 分散トレースのサポート

Eclipse Vert.x 4 は分散トレースをサポートします。トレースを使用してマイクロサービスを監視し、パフォーマンスの問題を特定することができます。

Eclipse Vert.x 4 は [OpenTracing](#) システムと統合します。

以下の Eclipse Vert.x コンポーネントはトレースをログに記録できます。

- HTTP サーバーおよび HTTP クライアント
- Eclipse Vert.x SQL クライアント
- Eclipse Vert.x Kafka クライアント



重要

トレースはテクノロジープレビューとして利用できます。テクノロジープレビュー機能は、Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

[テクノロジープレビュー機能のサポート範囲](#) については、Red Hat カスタマーポータル の「テクノロジープレビュー機能のサポート範囲」を参照してください。

3.1.11. EventBus JavaScript Client の新しい公開場所

Eclipse Vert.x 4 では、EventBus JavaScript クライアント **vertx-web-client.js** は Maven リポジトリーの Red Hat アーティファクトとして公開されません。

クライアントは npm リポジトリーに公開されます。@vertx/eventbus-bridge-client.js からクライアントにアクセスできます。

3.1.12. OpenShift Maven プラグインを使用した Eclipse Vert.x アプリケーションのデプロイ

OpenShift Maven プラグインを使用して、OpenShift に Eclipse Vert.x アプリケーションをデプロイします。Fabric8 Maven プラグインはサポート対象外になりました。詳細は、「[migrating from Fabric8 Maven Plugin to Eclipse JKube](#)」セクションを参照してください。

3.1.13. OpenShift の Eclipse Vert.x メータリングラベル

メータリングラベルを Eclipse Vert.x Pod に追加し、OpenShift Metering Operator で Red Hat サブスクリプションの詳細を確認できます。



注記

メータリングラベルは、Operator がデプロイおよび管理する Pod に追加しないでください。

Eclipse Vert.x は、以下のメータリングラベルを使用する必要があります。

- **com.redhat.component-name: Vert.x**
- **com.redhat.component-type: application**
- **com.redhat.component-version: 4.0.3**
- **com.redhat.product-name: "Red_Hat_Runtimes"**
- **com.redhat.product-version: 2021-Q1**

詳細は、「[メータリング](#)」のドキュメントを参照してください。

ラベルの詳細は、「[ノードでラベルを更新する方法について](#)」を参照してください。

3.1.14. OpenJDK 8 および OpenJDK 11 RHEL 8 Universal Base Images (UBI8) のサポート

Eclipse Vert.x は、RHEL 8 上の Red Hat OpenJDK 8 および Red Hat OpenJDK 11 用の OCI 準拠の [ユニバーサルベースイメージ](#) を使用して、Eclipse Vert.x アプリケーションをビルドして OpenShift にデプロイするためのサポートを導入します。

RHEL 8 OpenJDK Universal Base イメージは、RHEL 8 OpenJDK ビルダイメージを置き換えます。RHEL 8 OpenJDK ベースイメージは、Eclipse Vert.x との使用がサポートされなくなりました。

3.2. 非推奨の機能

本セクションでは、本リリースで非推奨または削除された機能を紹介します。

- **HttpServerOptions**

削除されたメソッド	置き換えメソッド
<code>getMaxWebsocketFrameSize()</code>	<code>getMaxWebSocketFrameSize()</code>
<code>setMaxWebsocketFrameSize()</code>	<code>setMaxWebSocketFrameSize()</code>
<code>getMaxWebsocketMessageSize()</code>	<code>getMaxWebSocketMessageSize()</code>
<code>setMaxWebsocketMessageSize()</code>	<code>setMaxWebSocketMessageSize()</code>
<code>getPerFrameWebsocketCompressionSupported()</code>	<code>getPerFrameWebSocketCompressionSupported()</code>
<code>setPerFrameWebsocketCompressionSupported()</code>	<code>setPerFrameWebSocketCompressionSupported()</code>
<code>getPerMessageWebsocketCompressionSupported()</code>	<code>getPerMessageWebSocketCompressionSupported()</code>
<code>setPerMessageWebsocketCompressionSupported()</code>	<code>setPerMessageWebSocketCompressionSupported()</code>
<code>getWebsocketAllowServerNoContext()</code>	<code>getWebSocketAllowServerNoContext()</code>
<code>setWebsocketAllowServerNoContext()</code>	<code>setWebSocketAllowServerNoContext()</code>
<code>getWebsocketCompressionLevel()</code>	<code>getWebSocketCompressionLevel()</code>
<code>setWebsocketCompressionLevel()</code>	<code>setWebSocketCompressionLevel()</code>
<code>getWebsocketPreferredClientNoContext()</code>	<code>getWebSocketPreferredClientNoContext()</code>
<code>setWebsocketPreferredClientNoContext()</code>	<code>setWebSocketPreferredClientNoContext()</code>
<code>getWebsocketSubProtocols()</code>	<code>getWebSocketSubProtocols()</code>
<code>setWebsocketSubProtocols()</code>	<code>setWebSocketSubProtocols()</code>

- **Eclipse Vert.x Web**

削除された要素	置き換える要素
<code>io.vertx.ext.web.Cookie</code>	<code>io.vertx.core.http.Cookie</code>
<code>io.vertx.ext.web.handler.CookieHandler</code>	<code>io.vertx.core.http.Cookie</code>

削除された要素	置き換える要素
<code>io.vertx.ext.web.Locale</code>	<code>io.vertx.ext.web.LanguageHeader</code>
<code>RoutingContext.acceptableLocales()</code>	<code>RoutingContext.acceptableLanguages()</code>
<code>StaticHandler.create(String, ClassLoader)</code>	---
<code>SessionHandler.setAuthProvider(AuthProvider)</code>	<code>SessionHandler.addAuthProvider()</code>
<code>HandlebarsTemplateEngine.getHandlebars()</code> <code>HandlebarsTemplateEngine.getResolvers()</code> <code>HandlebarsTemplateEngine.setResolvers()</code> <code>JadeTemplateEngine.getJadeConfiguration()</code> <code>ThymeleafTemplateEngine.getThymeleafTemplateEngine()</code> <code>ThymeleafTemplateEngine.setMode()</code>	<code>TemplateEngine.unwrap()</code>

- Messaging

削除されたメソッド	置き換えメソッド
<code>MessageProducer<T>.send(T)</code>	<code>MessageProducer<T>.write(T)</code>
<code>MessageProducer.send(T,Handler)</code>	<code>EventBus.request(String,Object,Handler)</code>

- EventBus

削除されたメソッド	置き換えメソッド
<code>EventBus.send(..., Handler<AsyncResult<Message<T>>>)</code> <code>Message.reply(..., Handler<AsyncResult<Message<T>>>)</code>	<code>replyAndRequest</code>

- Handlers

削除されたメソッド	置き換えメソッド
<code>Future<T>.setHandler()</code>	<code>Future<T>.onComplete()</code> <code>Future<T>.onSuccess()</code> <code>Future<T>.onFailure()</code>

削除されたメソッド	置き換えメソッド
<code>HttpClientRequest.connectionHandler()</code>	<code>HttpClient.connectionHandler()</code>

- JSON

削除されたフィールド/メソッド	新しいメソッド
<code>Json.mapper()</code> フィールド	<code>DatabindCodec.mapper()</code>
<code>Json.prettyMapper()</code> フィールド	<code>DatabindCodec.prettyMapper()</code>
<code>Json.decodeValue(Buffer, TypeReference<T>)</code>	<code>JacksonCodec.decodeValue(Buffer, TypeReference)</code>
<code>Json.decodeValue(String, TypeReference<T>)</code>	<code>JacksonCodec.decodeValue(String, TypeReference)</code>

- JUnit5

非推奨となったメソッド	新しいメソッド
<code>VertxTestContext.succeeding()</code>	<code>VertxTestContext.succeedingThenComplete()</code>
<code>VertxTestContext.failing()</code>	<code>VertxTestContext.failingThenComplete()</code>

- リアクティブエクステンション (Rx)

非推奨となったメソッド	新しいメソッド
<code>WriteStreamSubscriber.onComplete()</code>	<code>WriteStreamSubscriber.onWriteStreamEnd()</code> <code>WriteStreamSubscriber.onWriteStreamError()</code>

- サーキットブレーカー

削除されたメソッド	置き換えメソッド
<code>CircuitBreaker.executeCommand()</code>	<code>CircuitBreaker.execute()</code>
<code>CircuitBreaker.executeCommandWithFallback()</code>	<code>CircuitBreaker.executeWithFallback()</code>

- MQTT

削除されたメソッド	置き換えメソッド
MqttWill.willMessage()	MqttWill.getWillMessage()
MqttWill.willTopic()	MqttWill.getWillTopic()
MqttWill.willQos()	MqttWill.getWillQos()
MqttAuth.username()	MqttAuth.getUsername()
MqttAuth.password()	MqttAuth.getPassword()
MqttClientOptions.setKeepAliveTimeSeconds()	MqttClientOptions.setKeepAliveInterval()

- AMQP クライアント

削除されたメソッド	置き換えメソッド
AmqpClient.createReceiver(String address, Handler<AmqpMessage> messageHandler, ...)	AmqpClient createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)
AmqpConnection createReceiver(..., Handler<AsyncResult<AmqpReceiver>> completionHandler)	AmqpConnection createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)
AmqpConnection createReceiver(.., Handler<AmqpMessage> messageHandler, Handler<AsyncResult<AmqpReceiver>> completionHandler)	AmqpConnection createReceiver(String address, Handler<AsyncResult<AmqpReceiver>> completionHandler)

- 認証および認可

削除された要素	置き換える要素
OAuth2Options.isUseBasicAuthorizationHeader()	置き換えるメソッドなし
OAuth2Options.setUseBasicAuthorizationHeader()	置き換えるメソッドなし
OAuth2Options.getClientSecretParameterName()	置き換えるメソッドなし

削除された要素	置き換える要素
OAuth2Options.setClientSecretParameterName()	置き換えるメソッドなし
OAuth2Auth.createKeycloak()	KeycloakAuth.create(vertx, JsonObject) ()
OAuth2Auth.create(Vertx, OAuth2FlowType, OAuth2ClientOptions) ()	OAuth2Auth.create(vertx, new OAuth2ClientOptions().setFlow(YOUR_DESIRED_FLOW))
OAuth2Auth.create(Vertx, OAuth2FlowType)	OAuth2Auth.create(vertx, new OAuth2ClientOptions().setFlow(YOUR_DESIRED_FLOW))
User.isAuthorised()	User.isAuthorized()
AccessToken.refreshToken()	AccessToken.opaqueRefreshToken()
io.vertx.ext.auth.jwt.JWTOptions データオブジェクト	io.vertx.ext.jwt.JWTOptions データオブジェクト
SecretOptions クラス	PubSecKeyOptions クラス

非推奨となったメソッド	置き換えメソッド
OAuth2Auth.decodeToken()	AuthProvider.authenticate()
OAuth2Auth.introspectToken()	AuthProvider.authenticate()
OAuth2Auth.getFlowType()	置き換えるメソッドなし
OAuth2Auth.loadJWK()	OAuth2Auth.jwkSet()
OAuth2ClientOptions.isUseAuthorizationHeader()	置き換えるメソッドなし

非推奨のクラス	置き換えクラス
AbstractUser	<code>`User.create(JsonObject)`</code> メソッドを使用してユーザーオブジェクトを作成します。
AuthOptions	置き換えクラスなし

非推奨のクラス	置き換えクラス
JDBCAuthOptions	認証用 JDBCAuthenticationOptions 、および承認用 JDBCAuthorizationOptions
JDBCHashStrategy	置き換えクラスなし
OAuth2RBAC	AuthorizationProvider
Oauth2Response	WebClient クラスの使用が推奨
KeycloakHelper	置き換えクラスなし

- サービス検出

削除されたメソッド	置き換えメソッド
ServiceDiscovery.create(..., Handler<ServiceDiscovery> completionHandler)	ServiceDiscovery.create(Vertex)
ServiceDiscovery.create(..., Handler<ServiceDiscovery> completionHandler)	ServiceDiscovery.create(Vertex, ServiceDiscoveryOptions)

- Eclipse Vert.x 設定

削除されたメソッド	置き換えメソッド
ConfigRetriever.getConfigAsFuture()	retriever.getConfig()

- MongoDB クライアント

削除されたメソッド	置き換えメソッド
MongoClient.update()	MongoClient.updateCollection()
MongoClient.updateWithOptions()	MongoClient.updateCollectionWithOptions()
MongoClient.replace()	MongoClient.replaceDocuments()
MongoClient.replaceWithOptions()	MongoClient.replaceDocumentsWithOptions()
MongoClient.remove()	MongoClient.removeDocuments()

削除されたメソッド	置き換えメソッド
<code>MongoClient.removeWithOptions()</code>	<code>MongoClient.removeDocumentsWithOptions()</code>
<code>MongoClient.removeOne()</code>	<code>MongoClient.removeDocument()</code>
<code>MongoClient.removeOneWithOptions</code>	<code>MongoClient.removeDocumentsWithOptions()</code>

- 共有データソースのないクライアント

非推奨となったメソッド	新しいメソッド
<code>MongoClient.createNonShared()</code>	<code>MongoClient.create()</code>
<code>JDBCClient.createNonShared()</code>	<code>wJDBCClient.create()</code>
<code>CassandraClient.createNonShared()</code>	<code>CassandraClient.create()</code>
<code>MailClient.createNonShared()</code>	<code>MailClient.create()</code>

- Hook メソッド

削除されたメソッド	新しいメソッド
<code>Context.addCloseHook()</code>	置き換えるメソッドなし
<code>Context.removeCloseHook()</code>	置き換えるメソッドなし

- クローンメソッド

削除されたメソッド	新しいメソッド
<code>KeyCertOptions.clone()</code>	<code>KeyCertOptions.copy()</code>
<code>TrustOptions.clone()</code>	<code>TrustOptions.copy()</code>
<code>SSLEngineOptions.clone()</code>	<code>SSLEngineOptions.copy()</code>

- VertxOptions

削除されたメソッド	新しいメソッド
<code>VertxOptions.equals()</code>	置き換えるメソッドなし

削除されたメソッド	新しいメソッド
VertxOptions.hashCode()	置き換えるメソッドなし
VertxOptions.fileResolverCachingEnabled()	FileSystemOptions.isFileCachingEnabled()

- プールされたバッファ

削除されたメソッド	新しいメソッド
TCPSSLOptions.isUsePooledBuffers()	置き換えるメソッドなし
TCPSSLOptions.setUsePooledBuffers()	置き換えるメソッドなし

第4章 リリースコンポーネント

4.1. 本リリースで導入されたサポート対象のアーティファクト

本リリースでは、テクノロジープレビューからフルサポートになったアーティファクトはありません。

4.2. 本リリースで導入されたテクニカルプレビューアーティファクト

本リリースでは、以下のアーティファクトがテクノロジープレビューとして提供されます。

- **vertx-auth-webauthn**

Eclipse Vert.x 認証モジュール **io.vertx.ext.auth.AuthProvider** インターフェースは、2つの新しいインターフェースに分割されました。

- **io.vertx.ext.auth.authentication.AuthenticationProvider**
- **io.vertx.ext.auth.authorization.AuthorizationProvider**

認証は Eclipse Vert.x 4 の新機能です。以前のリリースでは、ユーザーが **User** オブジェクトでタスクを実行することが許可されているかどうかを確認できました。そのため、プロバイダーはユーザーの認証と承認の両方を行います。

Eclipse Vert.x 4 では、**User** オブジェクトインスタンスは特定の認証プロバイダーに関連付けられていません。そのため、異なるプロバイダーを使用してユーザーを認証および承認できます。

- **vertx-opentracing**

Eclipse Vert.x 4 は分散トレースをサポートします。トレースを使用してマイクロサービスを監視し、パフォーマンスの問題を特定することができます。

Eclipse Vert.x 4 は [OpenTracing](#) システムと統合します。

以下の Eclipse Vert.x コンポーネントはトレースをログに記録できます。

- HTTP サーバーおよび HTTP クライアント
- Eclipse Vert.x SQL クライアント
- Eclipse Vert.x Kafka クライアント



注記

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、「[テクノロジープレビュー機能のサポート範囲](#)」を参照してください。

4.3. 本リリースで削除されたアーティファクト

本リリースでは削除になったアーティファクトはありません。

4.4. 本リリースで非推奨となったアーティファクト

本リリースでは、非推奨となったアーティファクトはありません。

第5章 修正された問題

この Eclipse Vert.x リリースには、バージョン 4.0.3 のコミュニティーリリースのすべてのバグ修正が含まれます。コミュニティーリリースで解決された問題は、[Eclipse Vert.x 4.0.3](#) に記載されています。

5.1. GRAPHQL ビルドに含まれる GOOGLE GUAVA クラス

Eclipse Vert.x 4.0.0 および 4.0.2 リリースでは、**vertx-web-graphql** 依存関係を使用できませんでした。これは、バージョン 16.1.0.redhat-00001 を持つ GraphQL Java の不完全なビルドが使用されたためです。不完全な GraphQL ビルドには、Guava クラスがありませんでした。

この問題は Eclipse Vert.x 4.0.3 リリースで解決されました。このリリースには、Guava クラスの完全なビルドである GraphQL Java 16.1.0.redhat-00002 バージョンが含まれます。これらの Guava クラスは shaded jar になります。

5.2. ECLIPSE VERT.X ビルドで利用可能な VERTX-OPENTRACING

vertx-opentracing 依存関係は、Eclipse Vert.x 4.0.0 でテクノロジープレビュー機能として導入されました。ただし、この依存関係は Eclipse Vert.x 4.0.0 および 4.0.2 リリースでは利用できませんでした。

この問題は Eclipse Vert.x 4.0.3 リリースで解決されました。このリリースには **vertx-opentracing** 依存関係が含まれます。

第6章 既知の問題

6.1. KUBERNETESSERVICEIMPORTER() を ECLIPSE VERT.X REACTIVE EXTENSIONS (RX) に直接登録できない

説明

Eclipse Vert.x の Reactive Extensions (Rx) で **KubernetesServiceImporter()** を直接登録することはできません。

原因

サービスインポーターには生成された RxJava 2 実装がありません。

回避策

以下の例のように、**KubernetesServiceImporter** のインスタンスを作成し、{@link io.vertx.reactivex.servicediscovery.spi.ServiceImporter} でカプセル化する必要があります。

```
{@link
examples.RxServiceDiscoveryExamples#register(io.vertx.reactivex.servicediscovery.ServiceDiscovery)}
```

以下の例は、Eclipse Vert.x Reactive Extensions (Rx) で **KubernetesServiceImporter()** を登録する方法を示しています。

```
ServiceDiscovery discovery = ServiceDiscovery.create(vertx);
discovery.getDelegate().registerServiceImporter(new KubernetesServiceImporter(), new
JsonObject());
```

6.2. IBM Z および IBM POWER SYSTEMS では、RED HAT AMQ STREAMS イメージが利用できない

Red Hat AMQ Streams Operator および Kafka イメージは、IBM Z および IBM Power Systems では利用できません。イメージは利用できないため、**vertx-kafka-client** モジュールは IBM Z および IBM Power Systems 上の AMQ Streams と動作することが認定されていません。

6.3. TLS プロトコルのバージョンが一致しないため、RHEL 8 ベースのデータベースアプリケーションと RHEL 7 ベースの MYSQL 5.7 データベース間の接続が失敗する

説明

RHEL 8 ベースの OpenJDK ビルダイメージで構築されたアプリケーションコンテナと、RHEL 7 ベースの MySQL 5.7 コンテナイメージ上に構築されたデータベースコンテナとの間で、OpenSSL を使用して TLS でセキュア化された接続を開くと、ランタイム時に **javax.net.ssl.SSLHandshakeException** によって接続に失敗します。詳細は、[JIRA の問題](#) を参照してください。

```
...
Caused by: javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or
cipher suites are inappropriate)
...
```

原因

この問題は、RHEL 7 から RHEL 8 でサポートされる最新の TLS プロトコルバージョンが異なるために発生します。RHEL 7 の TLS 実装は、TLS プロトコルバージョン 1.0 (非推奨)、1.1、および 1.2 に対応しています。RHEL 8 の TLS 実装は、TLS プロトコルバージョン 1.3 にも対応しています。これは、RHEL 8 ベースのビルダーイメージで使用されるデフォルトの TLS バージョンでもあります。この不一致により、TLS ハンドシェイクのネゴシエーション中に、アプリケーションコンポーネント間で TLS プロトコルバージョンの不一致が発生し、アプリケーションとデータベースコンテナとの間の接続が失敗する可能性があります。

回避策

上記の問題を防ぐには、データベース接続文字列の両方のオペレーティングシステムバージョンでサポートされる TLS プロトコルバージョンを手動で指定します。以下に例を示します。

```
jdbc:mysql://testdb-mysql:3306/testdb?enabledTLSProtocols=TLSv1.2
```

6.4. FALSE CONNECTION がアプリケーションエンドポイントを呼び出す際にピアエラーメッセージによってリセットされる

curl ツールまたは Java HTTP クライアントのいずれかを使用して Eclipse Vert.x アプリケーションのエンドポイントで HTTP リクエストを行うと、リクエストごとに以下のエラーがログに出力されます。

```
io.vertx.core.net.impl.ConnectionBase  
SEVERE: java.io.IOException: Connection reset by peer
```

この動作は、Netty アプリケーションフレームワークと、OpenShift によって使用される HAProxy ロードバランサーの対話によって生じます。このエラーは、HAProxy が閉じずに既存の HTTP 接続が再使用されるため発生します。エラーメッセージがログに記録されても、エラー状態は発生しません。HTTP リクエストが正しく処理され、アプリケーションは予想通りに応答します。

第7章 本リリースに関連するアドバイザリー

本リリースに含まれる拡張機能、バグ修正、および CVE 修正を文書化するために、以下のアドバイザリーが発行されています。

- [RHSA-2021:0943](#)