



Red Hat build of MicroShift 4.12

API リファレンス

MicroShift API リファレンス

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.

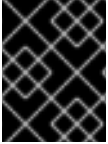
概要

このドキュメントでは、MicroShift の REST API 関数の詳細を説明します。

目次

第1章 APIティアについて	3
1.1. APIティア	3
1.2. APIティアのAPIグループへのマッピング	3
1.3. APIの非推奨ポリシー	4
第2章 API互換性ガイドラインについて	7
2.1. API互換性ガイドライン	7
2.2. APIの互換性例外	8
2.3. APIの互換性の一般的な用語	8
第3章 ネットワークAPI	11
3.1. ルート [ROUTE.OPENSIFT.IO/V1]	11
第4章 セキュリティーAPI	48
4.1. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSIFT.IO/V1]	48

第1章 APIティアについて



重要

このガイドンスでは、階層化された Red Hat build of MicroShift オファリングは説明しません。

Red Hat をお使いのアプリケーション開発者は、未特定の実装に特有の動作への依存や、特定の API 実装におけるバグへの依存を防ぐために、アプリケーション開発者が依存する動作が、API の公式ドキュメントで明示的に定義されていることを確認することが求められます。たとえば、入力ルーターの新しいリリースは、アプリケーションで、文書化されていない API を使用するか、未定義の動作に依存している場合に、古いリリースとは互換性がない場合があります。

1.1. APIティア

市販のすべての API、コンポーネント、および機能は、次のいずれかのサポートレベルに関連付けられています。

APIティア1

API とアプリケーションオペレーティング環境 (AOE) は、メジャーリリース内で安定しています。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。

APIティア2

API と AOE は、メジャーリリース内で最低 9 か月間、または非推奨の発表から 3 つのマイナーリリースのいずれか長い方で安定しています。

APIティア3

このレベルは、Operator Hub を通じて Red Hat build of MicroShift に含まれる言語、ツール、アプリケーション、およびオプションの Operator に適用されます。各コンポーネントは、API および AOE がサポートされるライフサイクルを指定します。言語ランタイム固有のコンポーネントの新しいバージョンは、マイナーバージョンからマイナーバージョンまで、API および AOE と可能な限り互換性があるように試みます。ただし、マイナーバージョンとマイナーバージョンの互換性は保証されません。

演算子およびオペランドと呼ばれる、演算子ハブを介して継続的な更新を受け取るコンポーネントおよび開発ツールは、APIティア3と見なす必要があります。開発者は注意を払い、これらのコンポーネントがマイナーリリースごとにどのように変化するかを理解する必要があります。ユーザーは、コンポーネントによって文書化された互換性ガイドラインを参照することが推奨されます。

APIティア4

互換性は提供されません。API および AOE はいつでも変更できます。これらの機能は、長期サポートを必要とするアプリケーションで使用できません。

Operator がタスクを実行するためにカスタムリソース定義 (CRD) を内部で使用する方法は一般的な方法です。これらのオブジェクトは Operator の外部にあるアクターによって使用されることが意図されておらず、非表示にされることが意図されています。CRD が Operator の外部のアクターによる使用を意図していない場合は、**ClusterServiceVersion (CSV) の operators.operatorframework.io/internal-objects** アノテーションを指定して、対応するリソースが内部使用のみである必要があります。また、CRD は、ティア4として排他的にラベル付けできます。

1.2. APIティアのAPIグループへのマッピング

Red Hat によって定義された APIティアごとに、アップストリームコミュニティが上位互換性を維持することを約束している特定の API グループのマッピングテーブルを提供します。明示的な互換性レベ

ルを指定せず、以下で具体的に説明されていない API グループには、デフォルトでティア 4 が割り当てられている **v1alpha1** API を除き、デフォルトで API ティア 3 が割り当てられます。

1.2.1. Kubernetes API グループのサポート

接尾辞 ***.k8s.io** で終わる、または接尾辞のないフォーム **version.<name>** を持つ API グループは、Kubernetes の非推奨ポリシーによって管理され、特に指定がない限り、公開されている API バージョンと対応するサポート階層の間の一般的なマッピングに従います。

API バージョンの例	API ティア
v1	ティア 1
v1beta1	ティア 2
v1alpha1	ティア 4

1.2.2. OpenShift API グループのサポート

接尾辞 ***.openshift.io** で終わる API グループは、Red Hat build of MicroShift の非推奨ポリシーによって管理され、特に指定がない限り、公開されている API バージョンと対応する互換性レベルの間の一般的なマッピングに従います。

API バージョンの例	API ティア
route.openshift.io/v1	ティア 1
security.openshift.io/v1	RangeAllocation (ティア 4) と *Reviews (ティア 2) を除くティア 1

1.3. API の非推奨ポリシー

Red Hat build of MicroShift は、多くのアップストリームコミュニティから供給された多くのコンポーネントで設定されています。コンポーネントのセット、関連する API インターフェイス、および相關する機能は時間の経過とともに進化し、機能を削除するために正式な非推奨が必要になる可能性がありますと予想されます。

1.3.1. API の一部の使用の終了

Red Hat build of MicroShift は、一連の構造化された API を介して、複数のコンポーネントがクラスターコントロールプレーンによって管理される共有状態と相互作用する分散システムです。Kubernetes の規則に従って、Red Hat build of MicroShift によって提示される各 API はグループ識別子に関連付けられ、各 API グループは個別にバージョン管理されます。各 API グループは、Kubernetes、Metal3、Multus、Operator Framework、Open Cluster Management、OpenShift 自体などを含む個別のアップストリームコミュニティで管理されます。

各アップストリームコミュニティは、特定の API グループおよびバージョンに対して固有の非推奨ポリシーを定義する場合がありますが、Red Hat はコミュニティ固有のポリシーを各アップストリームコミュニティの統合と認識に基づいて定義された互換性レベルのいずれかに正規化し、エンドユーザーの消費やサポートを簡素化します。

APIの非推奨ポリシーとスケジュールは、互換性レベルによって異なります。

非推奨ポリシーは、APIのすべての要素を対象とします。以下に例を示します。

- APIオブジェクトとしても知られる REST リソース
- REST リソースのフィールド
- バージョン固有の修飾子を除く、REST リソースのアノテーション
- 列挙値または定数値

各グループの最新の API バージョンを除き、古い API バージョンは、次の期間以上、非推奨が発表された後にサポートされる必要があります。

APIティア	期間
ティア1	メジャーリリース内で安定しています。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。
ティア2	非推奨の発表から9か月または3回のリリースのいずれか長い方。
ティア3	コンポーネント固有のスケジュールを参照してください。
ティア4	なし。互換性は保証されません。

次のルールは、すべての Tier1API に適用されます。

- API要素は、グループのバージョンをインクリメントすることによってのみ削除できます。
- APIオブジェクトは、一部のバージョンに存在しない REST リソース全体を除いて、情報を失うことなく API バージョン間をラウンドトリップする必要があります。バージョン間に同等のフィールドが存在しない場合、データは変換中に注釈の形式で保存されます。
- 特定のグループの API バージョンは、API オブジェクト全体が削除される場合を除いて、少なくとも安定した新しい API バージョンがリリースされるまで非推奨にすることはできません。

1.3.2. CLI 要素の廃止

クライアント向けの CLI コマンドは、API と同じ方法でバージョン管理されていませんが、ユーザー向けのコンポーネントシステムです。ユーザーが CLI を操作する2つの主な方法は、コマンドまたはフラグを使用することです。これは、このコンテキストでは CLI 要素と呼ばれます。

特に明記されていない限り、または CLI が下位層の API に依存している場合を除き、すべての CLI 要素はデフォルトで API 層1になります。

	要素	APIティア
一般提供 (GA)	フラグおよびコマンド	ティア 1
テクノロジープレビュー	フラグおよびコマンド	ティア 3
開発者プレビュー	フラグおよびコマンド	ティア 4

1.3.3. コンポーネント全体の非推奨化

コンポーネント全体を非推奨にする期間とスケジュールは、そのコンポーネントによって公開されている API の最上位 API 層に関連付けられている期間に直接マップされます。たとえば、ティア 1 とティア 2 の API を表面化したコンポーネントは、ティア 1 の非推奨スケジュールが満たされるまで削除できませんでした。

APIティア	期間
ティア 1	メジャーリリース内で安定しています。これらはメジャーリリース内で非推奨になる可能性があります。ただし、次のメジャーリリースまで削除されません。
ティア 2	非推奨の発表から 9 か月または 3 回のリリースのいずれか長い方。
ティア 3	コンポーネント固有のスケジュールを参照してください。
ティア 4	なし。互換性は保証されません。

第2章 API 互換性ガイドラインについて



重要

このガイダンスでは、階層化された Red Hat build of MicroShift オファリングは説明しません。

2.1. API 互換性ガイドライン

Red Hat では、Red Hat build of MicroShift との互換性を改善するために、アプリケーション開発者が以下の原則を採用することを推奨しています。

- アプリケーションのニーズに一致するサポート層を備えた API とコンポーネントを使用します。
- 可能な場合は、公開されたクライアントライブラリーを使用してアプリケーションを構築します。
- アプリケーションは、実行対象として構築された環境と同じくらい新しい環境で実行される場合にのみ、正しく実行されることが保証されます。Red Hat build of MicroShift 4.7 用にビルドされたアプリケーションは、Red Hat build of MicroShift 4.6 で適切に機能することが保証されていません。
- システムパッケージまたは他のコンポーネントにより提供される設定ファイルに依存するアプリケーションを設計しないでください。アップストリームコミュニティが、それを保存するように明示的にコミットしない限り、このファイルはバージョン間で変更する場合があります。必要に応じて、前方互換性を維持するために、これらの設定ファイルを通じて Red Hat が提供するインターフェイスの抽象化により異なります。設定ファイルを直接ファイルシステムで変更することはお勧めしません。デュアルライターの競合を回避するために、利用可能な場合は、Operator が提供する API と統合することが強く推奨されます。
- **unsupported<FieldName>** の接頭辞が付いた API フィールドや、製品ドキュメントに明示的に記載されていないアノテーションに依存しないでください。
- お使いのアプリケーションより互換性の保証が短いコンポーネントには依存しないでください。
- etcd サーバーで直接ストレージ操作を実行しないでください。すべての etcd アクセスは、api-server を介して、または文書化されたバックアップと復元の手順を介して実行する必要があります。

Red Hat は、アプリケーション開発者が Red Hat Enterprise Linux (RHEL) によって定義された [互換性ガイドライン](#) に従うことを推奨しています。Red Hat build of MicroShift は、プラットフォーム上でアプリケーションを構築またはホストする際に、以下のガイドラインを強く推奨しています。

- 特定の Linux カーネルまたは Red Hat build of MicroShift バージョンに依存しないでください。
- **proc**、**sys**、および **debug** ファイルシステム、または他の擬似ファイルシステムから読み込みは避けてください。
- **ioctl**s は使用せず、ハードウェアと直接対話してください。
- コンテナ実行環境を提供する Red Hat build of MicroShift ホストエージェントと競合しないように、**cgroups** との直接的な対話は避けてください。



注記

リリースのライフサイクルの間、Red Hat は、すべてのマイナーリリースと z-stream リリース間で API とアプリケーションオペレーティング環境 (AOE) の互換性を維持するために商業的に合理的な努力をします。ただし、必要に応じて、重大影響を及ぼすセキュリティ問題や、その他の重要な問題に対処するために、この互換性維持の目標に Red Hat が例外を設けることがあります。

2.2. API の互換性例外

以下は、Red Hat build of MicroShift での互換性の例外になります。

アップグレードされたクラスターと新規インストールの間の機能のデフォルト

現時点では、製品のマイナーリリースの新規インストールが、以前のマイナーリリースでインストールされ、同等のバージョンにアップグレードされた製品のバージョンと同じ機能デフォルトになるという保証はありません。たとえば、製品の将来のバージョンでは、以前のマイナーバージョンとは異なるデフォルトでクラウドインフラストラクチャーをプロビジョニングする可能性があります。さらに、製品の過去のバージョンで行われたものとは異なるデフォルトのセキュリティの選択が、製品の将来のバージョンで行われる可能性があります。製品の過去のバージョンはアップグレードを転送しますが、特に下位互換性を維持するために、必要に応じてレガシーの選択を保持します。

unsupported または undocumented annotations の接頭辞を持つ API フィールドの使用

製品で API を選択すると、接頭辞 **unsupported<FieldName>** が付いたフィールドが公開されます。現時点では、このフィールドの使用は複数リリースまたは1つリリース内でサポートされるようになりました。製品サポートは、特定の問題をデバッグするときにこのフィールドに値を指定するように顧客に要求できますが、その相互作用以外での使用はサポートされていません。明示的に文書化されていないオブジェクトでの注釈の使用は、マイナーリリース全体でのサポートが保証されていません。

製品インストールトポロジーごとの API の可用性

OpenShift ディストリビューションは、サポートされているインストールトポロジーを進化させ続け、あるインストールトポロジーのすべての API が必ずしも別のトポロジーに含まれるとは限りません。たとえば、特定のトポロジーでは、製品のインストールトポロジーと競合している場合、またはそのトポロジーに関連しない場合は特定の API がまったく含まれていない場合に、特定の API への読み取り/書き込みアクセスを制限できます。特定のトポロジーに存在する API は、上記で定義した互換性層に従ってサポートされます。

2.3. API の互換性の一般的な用語

2.3.1. アプリケーションプログラミングインターフェイス (API)

API は、他のソフトウェアとの対話を可能にするソフトウェアプログラムによって実装されるパブリックインターフェイスです。Red Hat build of MicroShift では、API は中央の API サーバーから提供され、すべてのシステム対話のハブとして使用されます。

2.3.2. アプリケーションオペレーティング環境 (AOE)

AOE は、エンドユーザーアプリケーションプログラムを実行する統合環境です。AOE は、ホストオペレーティングシステム (OS) からの分離を提供するコンテナ化された環境です。少なくとも、AOE を使用すると、アプリケーションをホスト OS ライブラリーおよびバイナリーから分離して実行できますが、ホスト上の他のすべてのコンテナと同じ OS カーネルを共有できます。AOE はランタイム時に適用され、アプリケーションとそのオペレーティングシステム間のインターフェイスを記述します。これには、プラットフォーム、オペレーティングシステム、および環境間の交差点が含まれ、ユーザーアプリケーションには、Downward API、DNS、リソースアカウンティング、デバイスアクセス、プラットフォームワークロード ID、コンテナ間の分離、コンテナとホスト OS 間の分離が含まれます。

AOE には、Container Network Interface (CNI) プラグインの選択や、アドミッションフックなどの製品の拡張機能など、インストールによって異なる可能性のあるコンポーネントは含まれていません。コンテナ環境より下のレベルでクラスターと統合するコンポーネントは、バージョン間で追加のバリエーションが適用される場合があります。

2.3.3. 仮想化環境における互換性

仮想環境は、ベアメタル環境で動作する非特権アプリケーションが、対応する仮想環境で変更されずに実行されるベアメタル環境をエミュレートします。仮想環境は、物理リソースが抽象化された簡易ビューを表示するため、いくつかの違いが存在する場合があります。

2.3.4. クラウド環境での互換性

Red Hat build of MicroShift は、クラウドプロバイダー固有の統合を介してホスティングクラウド環境との統合ポイントを提供することを選択する場合があります。これらの統合ポイントの互換性は、ネイティブクラウドベンダーによって提供される保証と、Red Hat build of MicroShift 互換性ウィンドウとの交差部分に固有のものであります。Red Hat build of MicroShift が、デフォルトのインストールの一部としてクラウド環境との統合をネイティブに提供する場合、Red Hat は安定したクラウド API エンドポイントに対して開発し、安定した非推奨ポリシーを含む将来を見据えた互換性を備えた商業的に合理的なサポートを提供します。クラウドプロバイダーと Red Hat build of MicroShift の統合の例には、動的ボリュームプロビジョニング、サービスロードバランサー統合、Pod ワークロード ID、コンピューティングの動的管理、初期インストールの一部としてのプロビジョニングされるインフラストラクチャーが含まれますが、これらに限定されるものではありません。

2.3.5. major、minor、および z-stream リリース

Red Hat メジャーリリースは、製品開発における重要なステップを表します。マイナーリリースは、メジャーリリースの範囲内でより頻繁に表示され、今後のアプリケーションの互換性に影響を与える可能性のある非推奨の境界を表します。z-stream リリースは、マイナーリリースへの更新で、関連するマイナーリリースに継続的な修正のストリームを提供します。予期しないセキュリティの影響に対応するためにこのポリシーが明示的にオーバーライドされる場合を除いて、API と AOE の互換性が z-stream リリースで破られることはありません。

たとえば、リリース 4.3.2 では、以下のようになります。

- 4 は、メジャーリリースのバージョンです。
- 3 は、マイナーリリースのバージョンです。
- 2 は、z-stream リリースバージョンです。

2.3.6. 拡張ユーザーサポート (EUS)

重大なバグ修正のためのサポート期間が延長された、Red Hat build of MicroShift メジャーリリースのマイナーリリース。ユーザーは、EUS リリース間でマイナーバージョンを段階的に採用することにより、EUS リリース間で移行できます。非推奨のポリシーはマイナーリリース全体で定義され、EUS リリースではなく、留意することが重要です。その結果、EUS ユーザーは、マイナーリリースごとに順次アップグレードしながら、将来の EUS に移行するときに非推奨に対応する必要がある場合があります。

2.3.7. 開発者プレビュー

Red Hat によって公式にサポートされておらず、初期段階のテクノロジーを探索するメカニズムを提供することを目的としたオプションの製品機能。デフォルトでは、Developer Preview 機能はオプトインであり、いつでも削除される可能性があります。開発者プレビュー機能を有効にすると、機能のスコア

によってクラスターがサポートされない可能性があります。

2.3.8. テクノロジープレビュー

機能をテストし、開発プロセス中にフィードバックを提供するために、今後の製品イノベーションへの早期アクセスを提供するオプションの製品機能。この機能は完全にはサポートされておらず、機能的に完全ではない可能性があり、実稼働での使用を目的としていません。テクノロジープレビューの機能を使用するには、明示的なオプトインが必要です。[テクノロジープレビュー機能のサポートスコープ](#)の詳細をご覧ください。

第3章 ネットワーク API

3.1. ルート [ROUTE.OPENSIFT.IO/V1]

説明

ルートを使用すると、開発者は HTTP(S) 対応の負荷分散とパブリック DNS エントリーを介したプロキシレイヤーを介してサービスを公開できます。ルートは、TLS オプションと証明書をさらに指定するか、ルーターが HTTP および HTTPS トラフィックに対しても受け入れる必要があるパブリック CNAME を指定する場合があります。管理者は通常、ルーターをクラスターファイアウォールの外側に表示されるように設定し、サービスコンテンツにセキュリティー、キャッシュ、またはトラフィック制御を追加する場合があります。ルーターは通常、サービスエンドポイントと直接通信します。

ルートが作成されると、**host** フィールドは変更されない場合があります。通常、ルーターは、競合を解決するときに、特定のホストで最も古いルートを使用します。

ルーターは追加のカスタマイズの対象であり、注釈フィールドを介して追加の制御をサポートする場合があります。

管理者は複数のルーターを設定できるため、ルートステータスフィールドは、各ルーターの下でのルートの名前と状態に関する情報をクライアントに返すために使用されます。たとえば、クライアントが重複した名前を選択した場合、ルートステータス条件は、ルートを選択できないことを示すために使用されます。

ルートで HTTP/2 ALPN を有効にするには、カスタム (非ワイルドカード) 証明書が必要です。これにより、クライアント、特に Web ブラウザーによる接続の合体が防止されます。接続の再利用/合体のリスクがあるため、デフォルトの証明書を使用するルートでは HTTP/2ALPN をサポートしていません。そのため、独自のカスタム証明書を持たないルートは、フロントエンドまたはバックエンドのいずれかで HTTP/2 ALPN で有効にされません。

互換性レベル 1: メジャーリリース内で最低 12 か月または 3 つのマイナーリリース (どちらか長い方) の間安定しています。

型

object

必須

- **spec**

3.1.1. 仕様

プロパティ	型	説明
-------	---	----

プロパティ	型	説明
apiVersion	string	APIVersion はオブジェクトのこの表現のバージョンスキーマを定義します。サーバーは認識されたスキーマを最新の内部値に変換し、認識されない値は拒否することがあります。詳細は、 https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources を参照してください。
kind	string	kind はこのオブジェクトが表す REST リソースを表す文字列の値です。サーバーはクライアントが要求を送信するエンドポイントからこれを推測できることがあります。これを更新することはできません。CamelCase の場合、詳細は、 https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds を参照してください。
metadata	ObjectMeta_v2	

プロパティ	型	説明
spec	object	<p>RouteSpec は、ルートが公開するホスト名またはパス、セキュリティ情報、およびルートが指す1つから4つのバックエンド(サービス)を記述します。リクエストは、各バックエンドに割り当てられた重みに応じてバックエンドに分散されます。ラウンドロビンスケジューリングを使用する場合、各バックエンドに送信されるリクエストの部分は、バックエンドの重みをすべてのバックエンドの重みの合計で割ったものです。バックエンドに複数のエンドポイントがある場合、バックエンドで終了する要求は、エンドポイント間でラウンドロビン分散されます。重みは0~256で、デフォルトは100です。重み0は、バックエンドへの要求を引き起こしません。すべての重みがゼロの場合、ルートにはバックエンドがないと見なされ、標準の503応答が返されます。</p> <p>tls フィールドはオプションであり、ルートの特定の証明書または動作を許可します。ルーターは通常、ワイルドカードドメインでデフォルトの証明書を設定して、明示的な証明書なしでルートを終了しますが、カスタムホスト名は通常、パススルー (TLS サーバー - 名前 - 表示フィールドを介してトラフィックをバックエンドに直接送信する) を選択するか、証明書を提供する必要があります。</p>
status	object	<p>RouteStatus は、どのルーターがルートを確認するかなど、ルートのステータスに関する関連情報を提供します。</p>

3.1.1.1. .spec

説明

RouteSpec は、ルートが公開するホスト名またはパス、セキュリティ情報、およびルートが指す1つから4つのバックエンド(サービス)を記述します。リクエストは、各バックエンドに割り当てられた重みに応じてバックエンドに分散されます。ラウンドロビンスケジューリングを使用する場合、各バックエンドに送信されるリクエストの部分は、バックエンドの重みをすべてのバックエン

ドの重みの合計で割ったものです。バックエンドに複数のエンドポイントがある場合、バックエンドで終了する要求は、エンドポイント間でラウンドロビン分散されます。重みは 0~256 で、デフォルトは 100 です。重み 0 は、バックエンドへの要求を引き起こしません。すべての重みがゼロの場合、ルートにはバックエンドがないと見なされ、標準の 503 応答が返されます。

tls フィールドはオプションであり、ルートの特定の証明書または動作を許可します。ルーターは通常、ワイルドカードドメインでデフォルトの証明書を設定して、明示的な証明書なしでルートを終了しますが、カスタムホスト名は通常、パススルー (TLS サーバー - 名前 - 表示フィールドを介してトラフィックをバックエンドに直接送信する) を選択するか、証明書を提供する必要があります。

型

object

必須

- **to**

プロパティ	型	説明
alternateBackends	array	AlternativeBackends を使用すると、最大 3 つの追加のバックエンドをルートに割り当てることができます。サービスの種類のみが許可され、デフォルトでサービスになります。 RouteTargetReference オブジェクトの重みフィールドを使用して、相対的な優先順位を指定します。
alternateBackends[]	object	RouteTargetReference は、エンドポイントに解決されるターゲットを指定します。サービスの種類のみが許可されます。'weight' フィールドを使用して、1 つを他よりも強調します。
host	string	ホストは、サービスを指すエイリアス/DNS です。任意。指定しない場合、ルート名は通常自動的に選択されます。DNS952 サブドメインの規則に従う必要があります。
path	string	トラフィックをサービスにルーティングするために、ルーターが監視するパス。任意。
port	object	RoutePort は、ルーターからサービスエンドポイントのエンドポイントへのポートマッピングを定義します。

プロパティ	型	説明
subdomain	string	<p>サブドメインは、Ingress コントローラーのドメイン内で (サブドメインとして) 要求される DNS サブドメインです。ホストが設定されている場合、このフィールドは無視されます。Ingress コントローラーは、この提案された名前を無視できます。その場合、コントローラーは、割り当てられた名前を status.ingress 配列に報告するか、ルートの許可を拒否します。この値が設定されていて、サーバーがこのフィールドをサポートしていない場合、ホストは自動的に入力されます。それ以外の場合、ホストは空のままになります。フィールドにはドットで区切られた複数の部分が含まれる場合がありますが、すべての Ingress コントローラーが要求を受け入れるとは限りません。このフィールドは、ルートの更新/カスタムホスト権限を持つユーザーを除いて、作成後に変更することはできません。</p> <p>例: サブドメイン frontend は、ルーターサブドメイン apps.mycluster.com を自動的に受信して、完全なホスト名 frontend.apps.mycluster.com を取得します。</p>
tls	object	<p>TLSConfig は、ルートを保護し、終了を提供するために使用される設定を定義します。</p>
to	object	<p>RouteTargetReference は、エンドポイントに解決されるターゲットを指定します。サービスの種類のみが許可されます。'weight' フィールドを使用して、1つを他よりも強調します。</p>
wildcardPolicy	string	<p>ルートにワイルドカードポリシーがある場合。現在、'Subdomain' または 'None' のみが許可されています。</p>

3.1.1.2. .spec.alternateBackends

説明

AlternativeBackends を使用すると、最大 3 つの追加のバックエンドをルートに割り当てることができます。サービスの種類のみが許可され、デフォルトでサービスになります。

RouteTargetReference オブジェクトの重みフィールドを使用して、相対的な優先順位を指定します。

型

array

3.1.1.3. .spec.alternateBackends[]

説明

RouteTargetReference は、エンドポイントに解決されるターゲットを指定します。サービスの種類のみが許可されます。'weight' フィールドを使用して、1 つを他よりも強調します。

型

object

必須

- **kind**
- **name**

プロパティ	型	説明
kind	string	ルートが参照しているターゲットの種類。現在、Service のみが許可されています。
name	string	参照されているサービス/ターゲットの名前。例: サービスの名前
weight	integer	0~256 の整数としての重み (デフォルトは 100)。これは、他のターゲット参照オブジェクトに対するターゲットの相対的な重みを指定します。0 は、このバックエンドへの要求を抑制します。

3.1.1.4. .spec.port

説明

RoutePort は、ルーターからサービスエンドポイントのエンドポイントへのポートマッピングを定義します。

型

object

必須

- **targetPort**

プロパティ	型	説明
targetPort	IntOrString	このルートが指すサービスによって選択された Pod のターゲットポート。これが文字列の場合は、ターゲットエンドポイントのポートリストで名前付きポートとして検索されます。必須

3.1.1.5. .spec.tls

説明

TLSConfig は、ルートを保護し、終了を提供するために使用される設定を定義します。

型

object

必須

- 終了

プロパティ	型	説明
caCertificate	string	caCertificate は、証明書機関の証明書のコンテンツを提供します
certificate	string	証明書は証明書のコンテンツを提供します。これは、証明書チェーンではなく、単一のサービング証明書である必要があります。CA 証明書を含めないでください。
destinationCACertificate	string	destinationCACertificate は、最終的な宛先の CA 証明書のコンテンツを提供します。reencrypt の停止を使用する場合は、ルーターがセキュアな接続のヘルスチェックに使用するためにこのファイルを提供する必要があります。このフィールドが指定されていない場合、ルーターは独自の宛先 CA を提供し、短いサービス名 (service.namespace.svc) を使用してホスト名の検証を実行する可能性があります。これにより、インフラストラクチャーが生成した証明書を自動的に検証できます。

プロパティ	型	説明
insecureEdgeTerminationPolicy	string	<p>insecureEdgeTerminationPolicy は、ルートへの安全でない接続に必要な動作を示します。各ルーターは公開するポートを独自に決定することがありますが、通常はポート 80 になります。</p> <p>*許可 - トラフィックは安全でないポートでサーバーに送信されます (デフォルト)。*無効 - 安全でないポートでのトラフィックは許可されません。*リダイレクト - クライアントはセキュアポートにリダイレクトされます。</p>
key	string	key はキーファイルの内容を提供します。
終了	string	<p>ターミネーションはターミネーションタイプを示します。</p> <p>*エッジ-TLS ターミネーションはルーターによって行われ、http はバックエンドとの通信に使用されます (デフォルト)。*パススルー-トラフィックはルーターが TLS ターミネーションを提供せずに宛先に直接送信されます。*再暗号化-TLS ターミネーションはルーターとによって行われ、https はバックエンドとの通信に使用されます。</p>

3.1.1.6. .spec.to

説明

RouteTargetReference は、エンドポイントに解決されるターゲットを指定します。サービスの種類のみが許可されます。'weight' フィールドを使用して、1つを他よりも強調します。

型

object

必須

- **kind**
- **name**

プロパティ	型	説明
kind	string	ルートが参照しているターゲットの種類。現在、Service のみが許可されています。
name	string	参照されているサービス/ターゲットの名前。例: サービスの名前
weight	integer	0~256 の整数としての重み (デフォルトは 100)。これは、他のターゲット参照オブジェクトに対するターゲットの相対的な重みを指定します。0 は、このバックエンドへの要求を抑制します。

3.1.1.7. .status

説明

RouteStatus は、どのルーターがルートを確認するかなど、ルートのステータスに関する関連情報を提供します。

型

object

プロパティ	型	説明
ingress	array	ingress は、ルートが公開する可能性のある場所を表します。Ingress ポイントのリストには、Host または RouterName の値が重複して含まれる可能性があります。 Ready となると、ルートはライブと見なされます。
ingress[]	object	RouteIngress は、ルートが公開されている場所に関する情報を保持します。

3.1.1.8. .status.ingress

説明

ingress は、ルートが公開する可能性のある場所を表します。Ingress ポイントのリストには、Host または RouterName の値が重複して含まれる可能性があります。**Ready** となると、ルートはライブと見なされます。

型

array

3.1.1.9. .status.ingress[]

説明

Routelngress は、ルートが公開されている場所に関する情報を保持します。

型

object

プロパティ	型	説明
conditions	array	条件はルートの状態であり、空の場合があります。
conditions[]	object	RoutelngressCondition には、特定のルーターにおけるこのルートの現況に関する情報が含まれます。
host	string	Host は、ルートが公開されるホスト文字列です。この値は必須です。
routerCanonicalHostname	string	CanonicalHostname は、このルートに要求されたホストの CNAME として使用できるルーターの外部ホスト名です。この値はオプションであり、すべての場合に設定できるとは限りません。
routerName	string	名前は、ルーターが自身を識別するために選択した名前です。この値は必須です。
wildcardPolicy	string	ワイルドカードポリシーは、このルートが公開されている場所で許可されたワイルドカードポリシーです。

3.1.1.10. .status.ingress[].conditions

説明

条件はルートの状態であり、空の場合があります。

型

array

3.1.1.11. .status.ingress[].conditions[]

説明

RoutelngressCondition には、特定のルーターにおけるこのルートの現況に関する情報が含まれます。

型

object

必須

- type
- status

プロパティ	型	説明
lastTransitionTime	Time	この条件が最後に遷移した RFC3339 の日付と時刻。
message	string	最後にある状態から別の状態に遷移した時間。
reason	string	(簡単な) 状態の最後の遷移の理由であり、通常は機械と人間が読み取れる定数です。
status	string	ステータスは、状態のステータスです。True、False、Unknown の場合があります。
type	string	タイプは条件のタイプです。現在のみ承認されています。

3.1.2. API エンドポイント

以下の API エンドポイントを利用できます。

- **/apis/route.openshift.io/v1/routes**
 - **GET:** 種類 Route のオブジェクトを一覧表示または監視します。
- **/apis/route.openshift.io/v1/watch/routes**
 - **GET:** ルートのリストへの個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。
- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes**
 - **DELETE:** ルートのコレクションを削除します。
 - **GET:** 種類 Route のオブジェクトを一覧表示または監視します。
 - **POST:** ルートを作成します。
- **/apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes**
 - **GET:** ルートのリストへの個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。

- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}**
 - **DELETE:** ルートを削除します。
 - **GET:** 指定されたルートを読み取ります。
 - **PATCH:** 指定されたルートを部分的に更新します。
 - **PUT:** 指定されたルートを置き換えます。
- **/apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes/{name}**
 - **GET:** 種類 Route のオブジェクトに対する変更を監視します。非推奨: 代わりに、リスト操作で watch パラメーターを使用し、fieldSelector パラメーターで単一の項目にフィルター処理します。
- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status**
 - **GET:** 指定されたルートのステータスを読み取ります。
 - **PATCH:** 指定されたルートのステータスを部分的に更新します。
 - **PUT:** 指定されたルートのステータスを置き換えます。

3.1.2.1. /apis/route.openshift.io/v1/routes

表3.1 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>
labelSelector	string	<p>返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

種類 Route のオブジェクトを一覧表示して監視します。

表3.2 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	RouteList スキーマ
401 - Unauthorized	空白

3.1.2.2. /apis/route.openshift.io/v1/watch/routes

表3.3 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>
labelSelector	string	<p>返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
resourceVersionMatch	string	resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。 デフォルトは unset です。
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

ルートへのリストへの個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。

表3.4 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	WatchEvent スキーマ
401 - Unauthorized	空白

3.1.2.3. /apis/route.openshift.io/v1/namespaces/{namespace}/routes

表3.5 グローバルパスパラメーター

パラメーター	型	説明
namespace	string	チームやプロジェクトなどのオブジェクト名と認証スコープ

表3.6 グローバルクエリーパラメーター

パラメーター	型	説明
pretty	string	true の場合は、出力がきれいに印刷されます。

HTTP メソッド

DELETE

説明

ルートのコレクションを削除します。

表3.7 クエリーパラメーター

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
dryRun	string	<p>これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
gracePeriodSeconds	integer	オブジェクトが削除されるまでの時間(秒)。値は負の値ではない整数にする必要があります。値をゼロに指定すると、ただちに削除されます。この値が nil の場合は、指定されたタイプのデフォルトの猶予期間が使用されます。指定しない場合は、デフォルトでオブジェクトごとの値になります。ゼロはすぐに削除することを意味します。
labelSelector	string	返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム(最大 0 個のアイテム)が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
orphanDependents	boolean	非推奨。PropagationPolicy を使用してください。このフィールドは 1.7 で非推奨となります。依存オブジェクトが孤立している必要があります。true/false の場合は、孤立したファイナライザーがオブジェクトのファイナライザー一覧で追加/削除されます。このフィールドまたは PropagationPolicy のいずれかを設定できますが、両方を設定することはできません。

パラメーター	型	説明
propagationPolicy	string	ガベージコレクションが実行されるかどうか、およびその方法。このフィールドまたは OrphanDependents のどちらかを設定できますが、両方を設定することはできません。デフォルトポリシーは、metadata.finalizers に設定されている既存のファイナライザーとリソース固有のデフォルトポリシーによって決定されます。許容値は次のとおりです。 'Orphan' - 依存を削除して孤立させます。 'Background' - ガベージコレクターがバックグラウンドで依存関係を削除できるようにします。 'Foreground' - フォアグラウンド内のすべての依存関係を削除するカスケードポリシーです。
resourceVersion	string	resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。 デフォルトは unset です。
resourceVersionMatch	string	resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。 デフォルトは unset です。
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。

表3.8 ボディパラメーター

パラメーター	型	説明
body	DeleteOptions スキーマ	

表3.9 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Status スキーマ

HTTP コード	レスポンス本文
401 - Unauthorized	空白

HTTP メソッド

GET

説明

種類 Route のオブジェクトを一覧表示して監視します。

表3.10 クエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があります。ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。

パラメーター	型	説明
labelSelector	string	返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

表3.11 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	RouteList スキーマ
401 - Unauthorized	空白

HTTP メソッド

POST

説明

ルートの作成

表3.12 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。

表3.13 ボディパラメーター

パラメーター	型	説明
body	Route スキーマ	

表3.14 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
201 - Created	Route スキーマ
202 - Accepted	Route スキーマ
401 - Unauthorized	空白

3.1.2.4. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes

表3.15 グローバルパスパラメーター

パラメーター	型	説明
namespace	string	チームやプロジェクトなどのオブジェクト名と認証スコープ

表3.16 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>
labelSelector	string	<p>返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

ルートのリストへの個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。

表3.17 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	WatchEvent スキーマ
401 - Unauthorized	空白

3.1.2.5. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}

表3.18 グローバルパスパラメーター

パラメーター	型	説明
name	string	ルートの名前
namespace	string	チームやプロジェクトなどのオブジェクト名と認証スコープ

表3.19 グローバルクエリーパラメーター

パラメーター	型	説明
pretty	string	true の場合は、出力がきれいに印刷されます。

HTTP メソッド

DELETE

説明

ルーターの削除

表3.20 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
gracePeriodSeconds	integer	オブジェクトが削除されるまでの時間(秒)。値は負の値ではない整数にする必要があります。値をゼロに指定すると、ただちに削除されます。この値が nil の場合は、指定されたタイプのデフォルトの猶予期間が使用されます。指定しない場合は、デフォルトでオブジェクトごとの値になります。ゼロはすぐに削除することを意味します。
orphanDependents	boolean	非推奨。PropagationPolicy を使用してください。このフィールドは 1.7 で非推奨となります。依存オブジェクトが孤立している必要があります。true/false の場合は、孤立したファイナライザーがオブジェクトのファイナライザー一覧で追加/削除されます。このフィールドまたは PropagationPolicy のいずれかを設定できますが、両方を設定することはできません。
propagationPolicy	string	ガベージコレクションが実行されるかどうか、およびその方法。このフィールドまたは OrphanDependents のどちらかを設定できますが、両方を設定することはできません。デフォルトポリシーは、metadata.finalizers に設定されている既存のファイナライザーとリソース固有のデフォルトポリシーによって決定されます。許容値は次のとおりです。 'Orphan' - 依存を削除して孤立させます。 'Background' - ガベージコレクターがバックグラウンドで依存関係を削除できるようにします。 'Foreground' - フォアグラウンド内のすべての依存関係を削除するカスケードポリシーです。

表3.21 ボディパラメーター

パラメーター	型	説明
body	DeleteOptions スキーマ	

表3.22 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Status スキーマ
202 - Accepted	Status スキーマ
401 - Unauthorized	空白

HTTP メソッド

GET

説明

指定されたルートを読み取ります。

表3.23 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
401 - Unauthorized	空白

HTTP メソッド

PATCH

説明

指定されたルートを部分的に更新します。

表3.24 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない <code>dryRun</code> ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は <code>All</code> で、すべてのドライランステージが処理されます。
fieldManager	string	<code>fieldManager</code> は、これらの変更を行っているアクターまたはエンティティーに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。このフィールドは、適用要求 (<code>application/apply-patch</code>) には必須ですが、非適用パッチタイプ (<code>JsonPatch</code> 、 <code>MergePatch</code> 、 <code>StrategicMergePatch</code>) にはオプションです。

パラメーター	型	説明
force	boolean	Force は Apply リクエストを強制します。これは、ユーザーが他の人が所有する競合するフィールドを再取得することを意味します。非適用パッチ要求の場合は、強制フラグを設定解除する必要があります。

表3.25 ボディパラメーター

パラメーター	型	説明
body	Patch スキーマ	

表3.26 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
201 - Created	Route スキーマ
401 - Unauthorized	空白

HTTP メソッド

PUT

説明

指定されたルートを置き換えます。

表3.27 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティーに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。

表3.28 ボディパラメーター

パラメーター	型	説明
body	Route スキーマ	

表3.29 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
201 - Created	Route スキーマ
401 - Unauthorized	空白

3.1.2.6. /apis/route.openshift.io/v1/watch/namespaces/{namespace}/routes/{name}

表3.30 グローバルパスパラメーター

パラメーター	型	説明
name	string	ルートの名前
namespace	string	チームやプロジェクトなどのオブジェクト名と認証スコープ

表3.31 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>
labelSelector	string	<p>返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

ルートの種類のオブジェクトへの変更を監視します。非推奨: 代わりに、リスト操作で watch パラメーターを使用し、fieldSelector パラメーターで単一の項目にフィルター処理します。

表3.32 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	WatchEvent スキーマ
401 - Unauthorized	空白

3.1.2.7. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status

表3.33 グローバルパスパラメーター

パラメーター	型	説明
name	string	ルートの名前
namespace	string	チームやプロジェクトなどのオブジェクト名と認証スコープ

表3.34 グローバルクエリーパラメーター

パラメーター	型	説明
pretty	string	true の場合は、出力がきれいに印刷されます。

HTTP メソッド

GET

説明

指定されたルートのステータスを読み取ります。

表3.35 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
401 - Unauthorized	空白

HTTP メソッド

PATCH

説明

指定されたルートの状態を部分的に更新します。

表3.36 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティに関連付けられた名前です。値は 128 文字未満である必要があります。 https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。このフィールドは、適用要求 (application/apply-patch) には必須ですが、非適用パッチタイプ (JsonPatch、MergePatch、StrategicMergePatch) にはオプションです。
force	boolean	Force は Apply リクエストを強制します。これは、ユーザーが他の人が所有する競合するフィールドを再取得することを意味します。非適用パッチ要求の場合は、強制フラグを設定解除する必要があります。

表3.37 ボディパラメーター

パラメーター	型	説明
body	Patch スキーマ	

表3.38 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
201 - Created	Route スキーマ
401 - Unauthorized	空白

HTTP メソッド

PUT

説明

指定されたルートの状態を置き換えます。

表3.39 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。

表3.40 ボディパラメーター

パラメーター	型	説明
body	Route スキーマ	

表3.41 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Route スキーマ
201 - Created	Route スキーマ
401 - Unauthorized	空白

第4章 セキュリティー API

4.1. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSIFT.IO/V1]

説明

SecurityContextConstraints (SCC) は、コンテナに適用される SecurityContext に影響を与える要求を行う機能を管理します。security.openshift.io グループを使用して、SecurityContextConstraints を管理します。互換性レベル 1: メジャーリリース内で最低 12 か月または 3 つのマイナーリリース (どちらか長い方) の間安定しています。

型

object

必須

- **allowHostDirVolumePlugin**
- **allowHostIPC**
- **allowHostNetwork**
- **allowHostPID**
- **allowHostPorts**
- **allowPrivilegedContainer**
- **readOnlyRootFilesystem**

4.1.1. 仕様

プロパティ	型	説明
allowHostDirVolumePlugin	boolean	AllowHostDirVolumePlugin は、ポリシーがコンテナに HostDir ボリュームプラグインの使用を許可するかどうかを決定します
allowHostIPC	boolean	AllowHostIPC は、ポリシーがコンテナ内のホスト ipc を許可するかどうかを決定します。
allowHostNetwork	boolean	AllowHostNetwork は、ポリシーが Pod 仕様で HostNetwork の使用を許可するかどうかを決定します。
allowHostPID	boolean	AllowHostPID は、ポリシーがコンテナ内のホスト pid を許可するかどうかを決定します。

プロパティ	型	説明
allowHostPorts	boolean	AllowHostPorts は、ポリシーがコンテナ内のホストポートを許可するかどうかを決定します。
allowPrivilegeEscalation	``	AllowPrivilegeEscalation は、Pod が特権の昇格を許可するように要求できるかどうかを決定します。指定しない場合は、デフォルトで true になります。
allowPrivilegedContainer	boolean	AllowPrivilegedContainer は、コンテナが特権としての実行を要求できるかどうかを判別します。
allowedCapabilities	``	AllowedCapabilities は、コンテナに追加するように要求できる機能のリストです。このフィールドの機能は、Pod 作成者の判断で追加される場合があります。AllowedCapabilities と RequiredDropCapabilities の両方に機能を追加することはできません。すべての機能を許可するには、'*' を使用できます。
allowedFlexVolumes	``	AllowedFlexVolumes は、許可された Flexvolume のホワイトリストです。空または nil は、すべての Flexvolume を使用できることを示します。このパラメーターは、"Volumes" フィールドで Flexvolumes の使用が許可されている場合にのみ有効です。

プロパティ	型	説明
allowedUnsafeSysctls	``	AllowedUnsafeSysctls は、明示的に許可された安全でない sysctl のリストであり、デフォルトは none です。各エントリーは、プレーンな sysctl 名であるか、" で終わります。この場合は、許可された sysctl の接頭辞と見なされます。1つの * は、すべての安全でない sysctl が許可されることを意味します。Kubelet は、拒否を回避するために、許可されているすべての安全でない sysctl を明示的にホワイトリストに登録する必要があります。例:たとえば、"foo/" は、"foo/bar"、"foo/baz" などを許可し、"foo.*" は、"foo.bar"、"foo.baz" などを許可します。
apiVersion	string	APIVersion はオブジェクトのこの表現のバージョンスキーマを定義します。サーバーは認識されたスキーマを最新の内部値に変換し、認識されない値は拒否することがあります。詳細は、 https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources を参照してください。
defaultAddCapabilities	``	DefaultAddCapabilities は、Pod 仕様で機能が明確に削除されない限り、コンテナに追加されるデフォルトの機能セットです。DefaultAddCapabilities と RequiredDropCapabilities の両方に機能をリストすることはできません。
defaultAllowPrivilegeEscalation	``	DefaultAllowPrivilegeEscalation は、プロセスが親プロセスよりも多くの特権を取得できるかどうかのデフォルト設定を制御します。

プロパティ	型	説明
forbiddenSysctls	``	ForbiddenSysctls は、明示的に禁止されている sysctl のリストであり、デフォルトは none です。各エントリーは、プレーンな sysctl 名であるか、" " で終わります。この場合は、禁止されている sysctl の接頭辞と見なされます。1つの * は、すべての sysctl が禁止されていることを意味します。例: たとえば、"foo/" は、"foo/bar"、"foo/baz"などを禁止し、"foo.*" は、"foo.bar"、"foo.baz"などを禁止します。
fsGroup	``	FSGroup は、SecurityContext によって使用される fs グループを決定する戦略です。
groups	``	このセキュリティーコンテキスト制約を使用する権限を持つグループ
kind	string	kind はこのオブジェクトが表す REST リソースを表す文字列の値です。サーバーはクライアントが要求を送信するエンドポイントからこれを推測できることがあります。これを更新することはできません。CamelCase の場合、詳細は、 https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds を参照してください。
metadata	ObjectMeta	標準オブジェクトのメタデータ。詳細は、 https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata を参照してください。

プロパティ	型	説明
priority	``	優先度は、Users フィールドおよび Groups フィールドのアクセスに基づいて、特定の Pod 要求に対して最初に試行する SCC を評価するときに SCC の並べ替え順序に影響を与えます。int が高いほど、優先度が高くなります。値を設定しないと、優先度が 0 と見なされます。複数の SCC のスコアが等しい場合は、最も制限の厳しいものから最も制限の少ないものへと並び替えられます。優先度および制限のどちらも等しい場合、SCC は名前順に並べ替えられます。
readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem を true に設定すると、コンテナは読み取り専用のルートファイルシステムで実行します。コンテナが読み取り専用ではないルートファイルシステムでの実行を特に要求する場合、SCC は Pod を拒否する必要があります。false に設定すると、コンテナは必要に応じて読み取り専用のルートファイルシステムで実行できますが、強制されることはありません。
requiredDropCapabilities	``	RequiredDropCapabilities は、コンテナからドロップされる機能です。これらは削除する必要があり、追加することはできません。
runAsUser	``	RunAsUser は、SecurityContext で使用される RunAsUser を決定する戦略です。
seLinuxContext	``	SELinuxContext は、SecurityContext に設定されるラベルを指示する戦略です。

プロパティ	型	説明
seccompProfiles	``	SeccompProfiles は、Pod またはコンテナの seccomp アノテーションに設定できる許可されるプロファイルを一覧表示します。未設定 (nil) または空の値は、Pod またはコンテナによってプロファイルを指定できないことを意味します。ワイルドカード '*' を使用して、すべてのプロファイルを許可できます。Pod の値を生成するために使用される場合は、最初のワイルドカード以外のプロファイルがデフォルトとして使用されます。
supplementalGroups	``	SupplementalGroups は、SecurityContext によって使用される補足グループを指示する戦略です。
users	``	このセキュリティコンテキストの制約を使用する権限を持つユーザー
volumes	``	ボリュームは、許可されたボリュームプラグインのホワイトリストです。FSType は、VolumeSource のフィールド名 (azureFile、configMap、emptyDir) に直接対応します。すべてのボリュームを許可するには、"*" を使用できます。ボリュームを許可しない場合は、["none"] に設定します。

4.1.2. API エンドポイント

以下の API エンドポイントを利用できます。

- **/apis/security.openshift.io/v1/securitycontextconstraints**
 - **DELETE:** SecurityContextConstraints のコレクションを削除する
 - **GET:** SecurityContextConstraints 種類のオブジェクトを一覧表示する
 - **POST:** SecurityContextConstraints を作成する
- **/apis/security.openshift.io/v1/watch/securitycontextconstraints**

- **GET**: SecurityContextConstraints のリストに対する個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。
- **/apis/security.openshift.io/v1/securitycontextconstraints/{name}**
 - **DELETE**: SecurityContextConstraints を削除する
 - **GET**: 指定された SecurityContextConstraints を読み取る
 - **PATCH**: 指定された SecurityContextConstraints を部分的に更新する
 - **PUT**: 指定された SecurityContextConstraints を置き換える
- **/apis/security.openshift.io/v1/watch/securitycontextconstraints/{name}**
 - **GET**: SecurityContextConstraints 種類のオブジェクトへの変更を監視します。非推奨: 代わりに、リスト操作で watch パラメーターを使用し、fieldSelector パラメーターで単一の項目にフィルター処理します。

4.1.2.1. /apis/security.openshift.io/v1/securitycontextconstraints

表4.1 グローバルクエリーパラメーター

パラメーター	型	説明
pretty	string	true の場合は、出力がきれいに印刷されます。

HTTP メソッド

DELETE

説明

SecurityContextConstraints のコレクションを削除します。

表4.2 クエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があります。ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	<p>返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。</p>
labelSelector	string	<p>返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。</p>

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティーまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

表4.3 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	Status スキーマ
401 - Unauthorized	空白

HTTP メソッド

GET

説明

SecurityContextConstraints 種類のオブジェクトを一覧表示します。

表4.4 クエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。
labelSelector	string	返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

表4.5 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	SecurityContextConstraintsList スキーマ
401 - Unauthorized	空白

HTTP メソッド

POST

説明

SecurityContextConstraints を作成します。

表4.6 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。

パラメーター	型	説明
fieldValidation	string	<p>fieldValidation は、`ServerSideFieldValidation` 機能ゲートも有効になっている場合に、不明または重複するフィールドを含むリクエスト (POST/PUT/PATCH) 内のオブジェクトを処理する方法をサーバーに指示します。有効な値は Ignore です。オブジェクトからサイレントにドロップされた不明なフィールドを無視し、デコーダーが検出した最後の重複フィールドを除くすべてを無視します。これは v1.23 以前のデフォルトの動作であり、`ServerSideFieldValidation` 機能ゲートが無効な場合のデフォルトの動作でもあります。警告: オブジェクトから削除された未知のフィールドと、検出された重複したフィールドごとに、標準の警告レスポンスヘッダーを介して警告を送ります。他にエラーがなければリクエストは成功し、重複するフィールドの最後のものだけを保持します。これは、`ServerSideFieldValidation` 機能ゲートが有効になっている場合のデフォルトです。Strict: オブジェクトから不明なフィールドが削除される場合、または重複するフィールドが存在する場合に、リクエストは BadRequest エラーで失敗します。サーバーから返されるエラーには、検出されたすべての不明な重複フィールドが含まれます。</p>

表4.7 ボディパラメーター

パラメーター	型	説明
body	SecurityContextConstraints スキーマ	

表4.8 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	SecurityContextConstraints スキーマ
201 - Created	SecurityContextConstraints スキーマ
202 - Accepted	SecurityContextConstraints スキーマ
401 - Unauthorized	空白

4.1.2.2. /apis/security.openshift.io/v1/watch/securitycontextconstraints

表4.9 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があります。ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。
labelSelector	string	返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>
resourceVersionMatch	string	<p>resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

SecurityContextConstraints のリストに対する個々の変更を監視します。非推奨です。代わりに、リスト操作で watch パラメーターを使用してください。

表4.10 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	WatchEvent スキーマ
401 - Unauthorized	空白

4.1.2.3. /apis/security.openshift.io/v1/securitycontextconstraints/{name}

表4.11 グローバルパスパラメーター

パラメーター	型	説明
name	string	SecurityContextConstraints の名前

表4.12 グローバルクエリーパラメーター

パラメーター	型	説明
pretty	string	true の場合は、出力がきれいに印刷されます。

HTTP メソッド

DELETE

説明

SecurityContextConstraints を削除します。

表4.13 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
gracePeriodSeconds	integer	オブジェクトが削除されるまでの時間 (秒)。値は負の値ではない整数にする必要があります。値をゼロに指定すると、ただちに削除されます。この値が nil の場合は、指定されたタイプのデフォルトの猶予期間が使用されます。指定しない場合は、デフォルトでオブジェクトごとの値になります。ゼロはすぐに削除することを意味します。
orphanDependents	boolean	非推奨。PropagationPolicy を使用してください。このフィールドは 1.7 で非推奨となります。依存オブジェクトが孤立している必要があります。true/false の場合は、孤立したファイナライザーがオブジェクトのファイナライザー一覧で追加/削除されます。このフィールドまたは PropagationPolicy のいずれかを設定できますが、両方を設定することはできません。
propagationPolicy	string	ガベージコレクションが実行されるかどうか、およびその方法。このフィールドまたは OrphanDependents のどちらかを設定できますが、両方を設定することはできません。デフォルトポリシーは、metadata.finalizers に設定されている既存のファイナライザーとリソース固有のデフォルトポリシーによって決定されます。許容値は次のとおりです。 'Orphan' - 依存を削除して孤立させます。 'Background' - ガベージコレクターがバックグラウンドで依存関係を削除できるようにします。 'Foreground' - フォアグラウンド内のすべての依存関係を削除するカスケードポリシーです。

表4.14 ボディパラメーター

パラメーター	型	説明
body	DeleteOptions スキーマ	

表4.15 HTTP 応答

HTTP コード	レスポンス本文
----------	---------

HTTP コード	レスポンス本文
200 - OK	Status スキーマ
202 - Accepted	Status スキーマ
401 - Unauthorized	空白

HTTP メソッド

GET

説明

指定された SecurityContextConstraints を読み込みます。

表4.16 クエリーパラメーター

パラメーター	型	説明
resourceVersion	string	resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。 デフォルトは unset です。

表4.17 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	SecurityContextConstraints スキーマ
401 - Unauthorized	空白

HTTP メソッド

PATCH

説明

指定された SecurityContextConstraints を部分的に更新します。

表4.18 クエリーパラメーター

パラメーター	型	説明
--------	---	----

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティーに関連付けられた名前です。値は 128 文字未満である必要があります。値は https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。
fieldValidation	string	fieldValidation は、`ServerSideFieldValidation` 機能ゲートも有効になっている場合に、不明または重複するフィールドを含むリクエスト (POST/PUT/PATCH) 内のオブジェクトを処理する方法をサーバーに指示します。有効な値は Ignore です。オブジェクトからサイレントにドロップされた不明なフィールドを無視し、デコーダーが検出した最後の重複フィールドを除くすべてを無視します。これは v1.23 以前のデフォルトの動作であり、`ServerSideFieldValidation` 機能ゲートが無効な場合のデフォルトの動作でもあります。警告: オブジェクトから削除された未知のフィールドと、検出された重複したフィールドごとに、標準の警告レスポンスヘッダーを介して警告を送ります。他にエラーがなければリクエストは成功し、重複するフィールドの最後のものだけを保持します。これは、`ServerSideFieldValidation` 機能ゲートが有効になっている場合のデフォルトです。Strict: オブジェクトから不明なフィールドが削除される場合、または重複するフィールドが存在する場合に、リクエストは BadRequest エラーで失敗します。サーバーから返されるエラーには、検出されたすべての不明な重複フィールドが含まれます。

表4.19 ボディパラメーター

パラメーター	型	説明
body	Patch スキーマ	

表4.20 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	SecurityContextConstraints スキーマ

HTTP コード	レスポンス本文
401 - Unauthorized	空白

HTTP メソッド

PUT

説明

指定された SecurityContextConstraints を置き換えます。

表4.21 クエリーパラメーター

パラメーター	型	説明
dryRun	string	これが指定されている場合は、変更を永続化してはならないことを示します。無効または認識されない dryRun ディレクティブは、エラーレスポンスを引き起こし、リクエストをそれ以上処理しません。有効な値は All で、すべてのドライランステージが処理されます。
fieldManager	string	fieldManager は、これらの変更を行っているアクターまたはエンティティに関連付けられた名前です。値は 128 文字未満である必要があります。、 https://golang.org/pkg/unicode/#IsPrint で定義されているように、印刷可能な文字のみが含まれている必要があります。
fieldValidation	string	fieldValidation は、`ServerSideFieldValidation` 機能ゲートも有効になっている場合に、不明または重複するフィールドを含むリクエスト (POST/PUT/PATCH) 内のオブジェクトを処理する方法をサーバーに指示します。有効な値は Ignore です。オブジェクトからサイレントにドロップされた不明なフィールドを無視し、デコーダーが検出した最後の重複フィールドを除くすべてを無視します。これは v1.23 以前のデフォルトの動作であり、`ServerSideFieldValidation` 機能ゲートが無効な場合のデフォルトの動作でもあります。警告: オブジェクトから削除された未知のフィールドと、検出された重複したフィールドごとに、標準の警告レスポンスヘッダーを介して警告を送ります。他にエラーがなければリクエストは成功し、重複するフィールドの最後のものだけを保持します。これは、`ServerSideFieldValidation` 機能ゲートが有効になっている場合のデフォルトです。Strict: オブジェクトから不明なフィールドが削除される場合、または重複するフィールドが存在する場合に、リクエストは BadRequest エラーで失敗します。サーバーから返されるエラーには、検出されたすべての不明な重複フィールドが含まれます。

表4.22 ボディパラメーター

パラメーター	型	説明
body	SecurityContextConstraints スキーマ	

表4.23 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	SecurityContextConstraints スキーマ
201 - Created	SecurityContextConstraints スキーマ
401 - Unauthorized	空白

4.1.2.4. /apis/security.openshift.io/v1/watch/securitycontextconstraints/{name}

表4.24 グローバルパスパラメーター

パラメーター	型	説明
name	string	SecurityContextConstraints の名前

表4.25 グローバルクエリーパラメーター

パラメーター	型	説明
allowWatchBookmarks	boolean	allowWatchBookmarks は、タイプが BOOKMARK の監視イベントを要求します。ブックマークを実装していないサーバーはこのフラグを無視する可能性があり、ブックマークはサーバーの判断で送信されます。クライアントは、ブックマークが特定の間隔で返されると想定したり、サーバーがセッション中に BOOKMARK イベントを送信すると想定したりすることはできません。これが watch でない場合は、このフィールドが無視されます。

パラメーター	型	説明
continue	string	<p>サーバーからさらに結果を取得する場合は、continue オプションを設定する必要があります。この値はサーバー定義であるため、クライアントは同じクエリーパラメーター (continue の値を除く) で前のクエリー結果からの continue 値のみを使用でき、サーバーは認識しない continue 値を拒否できます。指定された continue の値が有効期限切れ (通常 5-15 分) やサーバーでの設定変更により無効となった場合、サーバーは 410 ResourceExpired エラーと continue トークンを返します。クライアントが一貫性のあるリストを必要とする場合は、続行フィールドなしでリストを再起動する必要があります。それ以外の場合は、クライアントは 410 エラーで受信したトークンを使用して別のリスト要求を送信できます。サーバーは次のキーから始まるリストで応答しますが、最新のスナップショットから始まるため、以前のリスト結果とは矛盾します。作成、変更、または削除されたオブジェクトは、キーが「次のキー」の後にある限り応答に含まれます。</p> <p>watch が true の場合、このフィールドはサポートされません。クライアントは、サーバーから返された最後の resourceVersion 値から監視を開始し、変更を見逃すことはありません。</p>
fieldSelector	string	返されるオブジェクトのリストをフィールドごとに制限するセレクター。デフォルトは everything です。
labelSelector	string	返されるオブジェクトのリストをラベルで制限するためのセレクター。デフォルトは everything です。

パラメーター	型	説明
limit	integer	<p>limit は、リスト呼び出しに対して返される応答の最大数です。さらにアイテムが存在する場合、サーバーはリストメタデータの continue フィールドを、同じ初期クエリーで使用して次の結果セットを取得できる値に設定します。制限を設定すると、要求されたすべてのオブジェクトが除外された場合に、要求された量より少ないアイテム (最大 0 個のアイテム) が返される場合があります。クライアントは、続行フィールドの存在のみを使用して、より多くの結果が利用可能かどうかを判断する必要があります。サーバーは limit 引数をサポートしないことを選択でき、利用可能なすべての結果を返します。制限が指定され、続行フィールドが空の場合、クライアントはこれ以上結果が利用できないと想定する場合があります。watch が true の場合、このフィールドはサポートされません。</p> <p>サーバーは、continue を使用したときに返されるオブジェクトが、制限なしで単一のリスト呼び出しを発行するのと同じであることを保証します。つまり、最初の要求が発行された後に作成、変更、または削除されたオブジェクトは、後続の継続要求に含まれません。これは、一貫性のあるスナップショットと呼ばれることもあり、制限を使用して非常に大きな結果の小さなチャンクを受信するクライアントが、すべての可能なオブジェクトを確実に表示できるようにします。チャンクリスト中にオブジェクトが更新された場合は、最初のリスト結果が計算されたときに存在していたオブジェクトのバージョンが返されます。</p>
pretty	string	true の場合は、出力がきれいに印刷されます。
resourceVersion	string	<p>resourceVersion は、リクエストが提供される可能性のあるリソースバージョンに制約を設定します。詳細は、https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。</p> <p>デフォルトは unset です。</p>

パラメーター	型	説明
resourceVersionMatch	string	resourceVersionMatch は、resourceVersion をリスト呼び出しに適用する方法を決定します。resourceVersion が設定されているリスト呼び出しには resourceVersionMatch を設定することが強く推奨されます。詳細は、 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions を参照してください。 デフォルトは unset です。
timeoutSeconds	integer	リスト/ウォッチコールのタイムアウト。これにより、アクティビティまたは非アクティブに関係なく、通話時間が制限されます。
watch	boolean	説明されているリソースへの変更を監視し、それらを追加、更新、および削除の通知のストリームとして返します。resourceVersion を指定します。

HTTP メソッド

GET

説明

SecurityContextConstraints 種類のオブジェクトへの変更を監視します。非推奨: 代わりに、リスト操作で watch パラメーターを使用し、fieldSelector パラメーターで単一の項目にフィルター処理します。

表4.26 HTTP 応答

HTTP コード	レスポンス本文
200 - OK	WatchEvent スキーマ
401 - Unauthorized	空白