



# ESB としてのプロダクションの 使用

Version 2024.1  
2024-06-06

## ESB としてのプロダクションの使用

InterSystems IRIS Data Platform Version 2024.1 2024-06-06

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# 目次

1 エンタープライズ・サービス・バスおよびレジストリの概要 .....	1
1.1 エンタープライズ・サービス・バスの概念 .....	1
1.2 エンタープライズ・サービス・バス・アーキテクチャ .....	2
1.3 エンタープライズ・サービス・バスの構成 .....	4
2 パブリック・サービス・レジストリへのパブリック REST API を使用したアクセス .....	5
2.1 サービス・レジストリ・パブリック API の概要 .....	6
2.2 サービスの JSON 記述 .....	7
2.3 [すべてのサービスの取得] .....	11
2.4 [ID によるサービスの取得] .....	11
2.5 選択したサービスの取得 .....	11
2.6 ファイルの内容の取得 .....	13
3 パブリック・サービス・レジストリおよび外部サービス・レジストリの管理 .....	15
3.1 パブリック・サービス・レジストリの管理 .....	15
3.2 外部サービス・レジストリの管理 .....	16
3.3 パブリックと外部の両方のサービス・レジストリのフィールド .....	16
3.4 パブリック・サービス・レジストリの内部フィールド .....	19
3.5 サービス・レジストリ・エントリの作成と保守管理 .....	20
3.6 サービス・レジストリの検索と表示 .....	22
4 ESB の構成 .....	25
4.1 ESB としての InterSystems IRIS の構成の概要 .....	25
4.2 パブリック・サービス・レジストリに対するロールとユーザの定義 .....	25
4.3 パブリック・サービス・レジストリ REST API への Web アプリケーションの構成 .....	27
4.4 ESB ホストの構成での外部サービス・レジストリの使用 .....	28
5 ESB のサービスおよびオペレーションの構成 .....	31
5.1 パススルー・ビジネス・サービスの構成 .....	31
5.2 パススルー・ビジネス・オペレーションの構成 .....	32
5.3 パススルー・サービスでの SAML 検証の使用法 .....	33
5.4 パススルー・サービスおよびパススルー・オペレーションでの永続メッセージの抑止 .....	34
5.5 その他のビジネス・サービス、ビジネス・プロセス、およびビジネス・オペレーションの使用 .....	34
5.6 パススルー・サービスおよびパススルー・オペレーションのパフォーマンス統計の追跡 .....	35
付録A: InterSystems IRIS システムの構成とネームスペースの作成 .....	37
付録B: パススルー・ビジネス・サービスへの Web アプリケーションの構成 .....	39
付録C: パススルー・サービスとパススルー・オペレーションのウォークスルー .....	41
C.1 REST パススルー・サービスおよびオペレーションの使用法 .....	41
C.1.1 パススルーの背景情報 .....	43
C.1.2 パススルーのトラブルシューティング .....	43
C.2 SOAP パススルー・サービスおよびオペレーションの使用法 .....	44
C.2.1 SOAP ツールキットからの外部サーバの呼び出し .....	45
C.2.2 SOAP パススルー・サービスおよびオペレーションの追加 .....	45
サービス・レジストリ・パブリック API リファレンス .....	47
レジストリ・エントリを伴った JSON 戻りメッセージのサンプル .....	48
一般的な URL パラメータ .....	50
[取得] .....	51

[ID によるファイルの取得] .....	52
[サービスの取得] .....	53
[ID によるサービスの取得] .....	54
[プロトコルによるサービスの取得] .....	55
[段階によるサービスの取得] .....	56
[バージョンによるサービスの取得] .....	57
[単語によるサービスの取得] .....	58
[変更以降のサービスの取得] .....	59

# 図一覧

図 1-1: 単純なエンタープライズ・サービス・バス・アーキテクチャ .....	3
--	---



# 1

## エンタープライズ・サービス・バスおよびレジストリの概要

ここでは、エンタープライズ・サービス・バスとしての InterSystems IRIS® の使用を紹介し、InterSystems IRIS ESB アーキテクチャについて説明して、ESB の配置の概要を示します。

### 1.1 エンタープライズ・サービス・バスの概念

エンタープライズ・サービス・バス (ESB) は、SOAP、REST、またはその他のネットワーク API を使用するアプリケーションへのアクセスとそれらのアプリケーションの管理を行うための単一ポイントを提供します。ESB には、以下の機能が用意されています。

- ・ サービスの検出とサービスへのアクセスを行うための一元化された場所を提供します。
- ・ サービスの詳細情報からアプリケーションを分離することにより、ユーザが単一箇所でサービスの説明を変更できるようにします。このとき、そのサービスに依存しているすべてのアプリケーションを更新する必要がありません。例えば、サーバのアドレスや、サービスの API を変更することもでき、それらの変更を ESB で処理できます。これによって、アプリケーションに対してプロトコルの独立性が確保されます。SOAP API を使用して開発したアプリケーションだが、新しいサービスで REST API を使用する追加機能を提供する場合、ESB はこの新しい REST サービスを REST と SOAP の両方の API として使用可能にできるので、ユーザは既存アプリケーションとの互換性を維持しながら新しいサービスの機能を追加できます。
- ・ 企業によって使用されるアプリケーション、およびそれらのアプリケーション間の依存関係を編成および追跡するメカニズムを提供します。例えば、アプリケーションに関する問題が起きたときにユーザは、ESB からテクニカル・サポートの問い合わせ先情報を入手できます。

ESB が必要な理由。以下のシナリオを検討してください。

1. 企業で重要なアプリケーションがたくさん開発されています。それぞれのアプリケーションが独自の環境とユーザ・インタフェースを有しており、他のアプリケーションから独立しています。
2. 効率性の向上において、新しい問題と厳しい要件に直面しています。これらの問題を解決するには、これらの独立したアプリケーションが互いに通信し、対話するように設定する必要があります。
3. SOAP API または REST API を使用したアクセスを提供するように、アプリケーションを変更します。
4. SOAP API および REST API を呼び出す新しいアプリケーションの開発を始めます。新しいアプリケーションに、既存アプリケーションの機能を組み込むことができ、ワークフローを組み込んで、手順を合理化することが可能になります。

5. ただし、新しいアプリケーションそれぞれに、使用するアプリケーションの詳細情報、つまり、そのアプリケーションを実行するサーバのアドレスおよびそのアプリケーションへのアクセスに必要な特定の API が必要です。
6. 新しいアプリケーションの数が増えるにつれ、アプリケーション全体をまとめて保守管理することが難しくなっていきます。サービスを新しいサーバに移動したり、サービスの機能に変更を加えるだけで、そのサービスに依存している各アプリケーションを更新する必要があるが生じます。サービスに依存しているアプリケーションを見つけ出すことさえ難しい場合があります。
7. ユーザは、問題が発生したときにどこに問い合わせればよいのかよくわかりません。

InterSystems IRIS を ESB として使用すれば、サービスにアクセスするための統合メカニズムを作成できます。これによって、一連のアプリケーションの保守管理が簡単になり、使用を追跡したり、潜在的な障害を特定したりするメカニズムが実現されます。

## 1.2 エンタープライズ・サービス・バス・アーキテクチャ

InterSystems IRIS ESB アーキテクチャのコンポーネントは、次のとおりです。

- ・ 受信要求をサーバに接続するルーティング・メカニズム。これは、専用のビジネス・サービスとビジネス・オペレーション、およびビジネス・プロセスのプロダクション (オブション) によって実装されます。
- ・ パブリック・サービス・レジストリ。このサービス・レジストリには、ESB パブリック REST API を使用して ESB クライアントからアクセスできます。開発者は、これを使用して、ESB 経由で使用可能なサービスに関する情報を取得できます。
- ・ 外部サービス・レジストリ。このサービスは、ESB プロダクション内でのみアクセス可能です。これは、ESB ビジネス・ホストにエンドポイントの情報を提供します。
- ・ SAML トークン検証サービス。

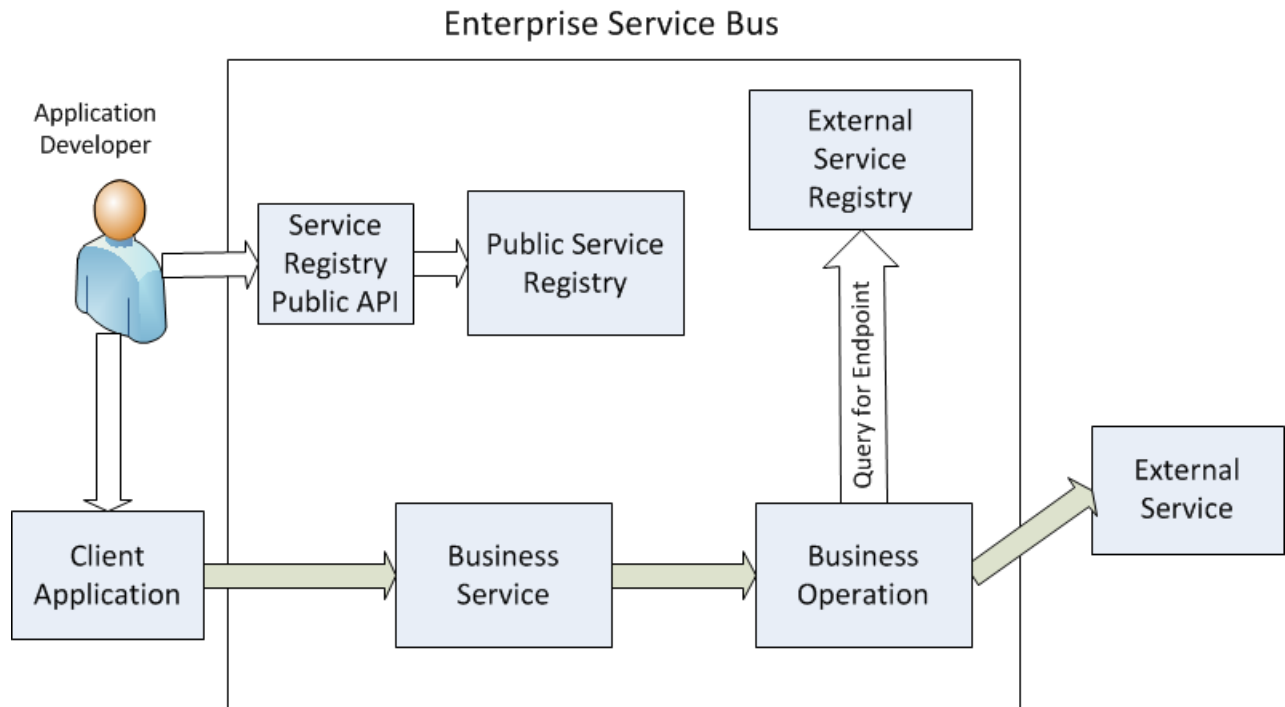
ESB プロダクションとサービス・レジストリの両方を、ESB 専用のネームスペースで定義する必要があります。

注釈 1 つの InterSystems IRIS インスタンスで 1 つの ESB プロダクションのみを実行することをお勧めします。

この最も単純な形では、ESB はパススルー・サービス、パススルー・オペレーション、およびサービス・レジストリで構成されます。次の図は、単純な ESB のアーキテクチャを示しています。



図 1-1: 単純なエンタープライズ・サービス・バス・アーキテクチャ



通常、アプリケーション開発者は、使用可能なサービスについて検索し、基盤となるサービスへのアクセスを提供するパススルー・ビジネス・サービスの URL を取得するために、Web ページまたはアプリケーションを使用して ESB サービス・レジストリにクエリを実行します。このクエリおよびサービスの検出プロセスは、開発者がクライアント・アプリケーションを作成しているときに実行されます。開発者が、サービスへのアクセスに必要な URL や説明などの情報を取得した後は、クライアント・アプリケーションからは、サービス呼び出すためにサービス・レジストリにアクセスする必要はありません。環境によっては、クライアント・アプリケーションからパブリック API に実行時呼び出しを行って、レジストリ・エントリが前回のアクセス以降に変更されていないことを確認する場合があります。

クライアント・アプリケーションは、パススルー・サービス呼び出します。パススルー・サービスは、ターゲットの設定に指定されているパススルー・オペレーションにメッセージを送信します。そのパススルー・オペレーションは、外部サービス・レジストリでエンドポイント URL を検索するよう構成されています。パススルー・オペレーションは、エンドポイント URL を使用して外部サービス呼び出します。外部サービスは、パススルー・オペレーションに応答を返し、パススルー・オペレーションはその応答をパススルー・サービスに転送します。次にパススルー・サービスは、その応答をクライアント・アプリケーションに返します。

パススルー・サービスとパススルー・オペレーションに加えて、より複雑なサービス、オペレーション、およびビジネス・プロセスを定義して、次のような機能を追加することができます。

- ・ 1 つの受信サービスからの呼び出しを、呼び出しの内容に応じて複数の外部サービスにルーティングする処理。これによって、ESB は、1 つの外部サービスからでは使用できないサービスを提供できます。クライアント・アプリケーションが、拡張されたサービスの実装に必要な外部サービスの影響を受けることはありません。
- ・ サービスに使用されるパラメータまたはプロトコルの変更。1 つのサービスを使用する一連のアプリケーションを開発したが、後に異なる API を使用するより優れたサービスが使用可能になった場合、ESB では、各アプリケーションを変更するのではなく、変換を実行できます。クライアント・アプリケーションが、1 つの外部サービスから別のサービスへの切り替えに必要な変更の影響を受けることはありません。
- ・ ESB へのサービスの直接の実装要件を満たす外部サービスがない場合、ObjectScript を使用して ESB に外部サービスを実装できます。

ただし、このようなより複雑なサービスを ESB に追加すると効率性が犠牲になります。複雑なサービスにより追加の処理が必要になり、ESB による要求の処理が遅くなり、スループットが低下します。

きわめて高度なスループットが必要な ESB システムの場合、永続メッセージを排除することによって、要求処理のオーバーヘッドを削減できます。永続メッセージは、パススルー・サービスからパススルー・オペレーションに送信され、オペレーションからサービスに返されるオブジェクトです。これらのオブジェクトは、InterSystems IRIS データベースに格納されます。これらの永続メッセージは、ESB で処理された呼び出しの追跡とレポート、および問題が発生した場合のトラブルシューティングに役立ちます。ただし、これらのオブジェクトの作成にはリソースが必要で、スループットがきわめて高度なシステムの場合、これらのオブジェクトに必要なストレージはかなり大きくなる場合があります。システムを保持するためには、これらのメッセージを頻繁にパージする必要があります。永続メッセージの使用を抑止し、柔軟性を下げることによって効率性を高めることができます。詳細は、“[パススルー・サービスおよびパススルー・オペレーションでの永続メッセージの抑止](#)”を参照してください。

**注釈** HealthShare 製品を実行している場合、HealthShare サービス・レジストリは、InterSystems IRIS サービス・レジストリとは異なります。HealthShare サービス・レジストリは、InterSystems IRIS 外部サービス・レジストリと同様の機能を提供します。ほとんどの場合、HealthShare サービス・レジストリの使用を続行し、InterSystems IRIS 外部サービス・レジストリは使用しないでください。

## 1.3 エンタープライズ・サービス・バスの構成

InterSystems IRIS のインストールと構成は、“[プロダクションの作成準備](#)”で説明されています。このドキュメントでは、ESB として使用する InterSystems IRIS インストール固有の構成手順について説明します。これらの手順は、次のとおりです。

- ・ ESB プロダクションおよびサービス・レジストリを格納するための相互運用対応ネームスペースを作成します。
- ・ Web ゲートウェイを構成します。
- ・ ESB で使用する外部サービスのエンドポイントを定義する外部サービス・レジストリのエントリを作成します。
- ・ ESB プロダクションを作成し、ビジネス・サービス、およびそれらのサービスを提供するビジネス・オペレーションを追加して、プロダクションを開始します。
- ・ 管理ポータルからパブリック REST API を使用してパブリック・サービス・レジストリにアクセスするときに必要なロールとユーザを作成します。
- ・ クライアントでビジネス・サービスを使用できるようにするために必要な Web アプリケーションを作成します。
- ・ ESB からアクセス可能なサービスを記述するパブリック・サービス・レジストリのエントリを作成します。

詳細は、“[ESB としての InterSystems IRIS の構成の概要](#)”を参照してください。

# 2

## パブリック・サービス・レジストリへのパブリック REST API を使用したアクセス

パブリック REST API は、パブリック・サービス・レジストリへの読み取りアクセスを提供します。API は、ESB 経由で使用可能なサービスに関する情報を提供します。すべてのサービスに関する情報を返す REST 呼び出しを実行することも、指定の条件を満たすサービスに対してクエリを実行することもできます。呼び出しは、サービスに関する情報を JSON で返します。

ESB がどのように構成されているかによって、ユーザ名とパスワードを使用して、または匿名アクセスで API にアクセスできます。レジストリは、ユーザまたは匿名アクセスに付与された権限に基づいて、サービスを返します。サービス・レジストリ管理者は、ログイン情報に基づいてどのサービスが返されるかを制御します。例えば、管理者は、次のいずれかの方法でレジストリを構成できます。

- ・ 匿名でログインしたユーザも含め、すべてのユーザにすべてのサービスに関する情報を返します。このポリシーでは、サービスに関する情報を取得できても、そのサービスを呼び出す権限があるわけではありません。
- ・ ログイン・アカウントまたは匿名アクセスに基づいて、ユーザが呼び出す権限を持つサービスのみ、返されるサービスを限定します。

各サービス・レジストリ・エントリには、以下の情報が含まれています。

- ・ サービスの ID。名前、ドメイン、およびバージョンで構成されます。
- ・ エンドポイント URL。サービスを呼び出すためのアドレスを提供します。通常、このアドレスは、ESB サーバ上にあります。サービスを実際に実行する外部サーバのエンドポイントは返されません。
- ・ サービスの呼び出しに使用するプロトコル。REST、SOAP、HTTP、TCP など。
- ・ サービスで使用する戻りのメカニズム。同期の戻りやコールバックなど。
- ・ 説明。サービスの概要を示した簡単なテキストを提供します。
- ・ トピック。サービスの検索に使用されます。
- ・ スキーマ。スキーマ言語を使用しているメッセージについて返されるメッセージのスキーマ説明が含まれます。スキーマ情報には、スキーマ形式が含まれており、スキーマ定義の URL またはスキーマのコンテンツが提供されます。
- ・ アクションは、指定されたエンドポイントで使用可能な HTTP 要求メソッドを表します。それぞれのアクションは、サーバに対する要求メソッドの影響を表すことができ、受信メッセージおよび応答メッセージのスキーマを指定できます。
- ・ 連絡先は、サービスの情報またはサポートを提供できるユーザまたは組織を表します。
- ・ 属性は、レジストリ管理者が各サービスについて提供される情報を拡張するためのメカニズムを提供します。属性は、一連の、名前と値のペアで構成されます。

- ファイルは、サービスに関する一連のファイルを提供するメカニズムを提供します。通常、これらはサービスを使用するための説明またはテンプレートの提供に使用されます。パブリック API がファイルに関する情報を返す場合、それにはファイルの内容は含まれていません。ファイルの内容はサイズがかなり大きい可能性があるためです。ファイルの内容にアクセスするには、明示的な要求を実行する必要があります。

## 2.1 サービス・レジストリ・パブリック API の概要

この節では、パブリック API での REST 呼び出しの概要を示します。各呼び出しの詳細は、“[サービス・レジストリ・パブリック API リファレンス](#)”を参照してください。

REST 呼び出し	URL および説明
[取得]	GET /about 実行可能なパブリック API REST 呼び出しが一覧表示されます。
[ID によるファイルの取得]	GET /services/id/name/domain/version/file/filename 指定されたファイルが返されます。
[サービスの取得]	GET /services ユーザがアクセス可能なすべてのサービスが表示されます。
[ID によるサービスの取得]	GET /services/id/name/domain/version ユーザからアクセス可能で、指定された名前、ドメイン、およびバージョンに一致するサービスが返されます。
[プロトコルによるサービスの取得]	GET /services/protocols/protocol-list ユーザからアクセス可能で、リスト内のいずれかのプロトコルに一致するすべてのサービスが表示されます。
[段階によるサービスの取得]	GET /services/stages/stage-list ユーザからアクセス可能で、リスト内のいずれかの段階に一致するすべてのサービスが表示されます。
[バージョンによるサービスの取得]	GET /services/version/version ユーザからアクセス可能で、指定されたバージョンに一致するすべてのサービスが表示されます。
[単語によるサービスの取得]	GET /services/includesword/search-text ユーザからアクセス可能で、指定された検索テキストに一致するすべてのサービスが表示されます。
[変更以降のサービスの取得]	GET /services/modifiedsince/date-time ユーザからアクセス可能で、指定された日時以降にレジストリ・エントリが作成されたか更新されているすべてのサービスが表示されます。

## 2.2 サービスの JSON 記述

サービスの取得呼び出しにより、共に返されるサービスの数に関係なく、1 つの JSON メッセージが返されます。サービスの取得要求に一致していて、さらにユーザからアクセス可能であるサービスがない場合、呼び出しは空の JSON メッセージを返します。それ以外の場合は、サービスの取得呼び出しにより、要求に一致し、ユーザからアクセス可能な 1 つ以上のサービスと共に JSON メッセージが返されます。この節では、返される JSON メッセージの例を示し、メッセージ内のフィールドについて説明します。

いずれのサービスの取得呼び出しによっても、次のようなメッセージが返されます。WSDL スキーマは、リスト内では省略形で示されていますが、JSON メッセージでは完全な形で表示されます。

```
[
{
  "Name": "MathServiceSOAP",
  "Domain": "UnitTest",
  "Version": "1.1",
  "Stage": "Live",
  "Protocol": "SOAP",
  "Description": "Add 2 Numbers",
  "Endpoint": "https://jgm6457/enslatest/csp/support/Demo.SOAP.MathService.cls",
  "ResponseStyle": "Sync",
  "LastModified": "2022-03-16 19:07:47.469",
  "Topics":
  [
    "Test",
    "Maths"
  ],
  "Contacts":
  [
    {
      "Identity": "QD Developer Moon",
      "Type": "Operator",
      "Details": "Details of contact",
      "BusinessPartner": "QD",
      "Notes": "This SOAP service is designed to have minimum moving parts"
    }
  ],
  "Schema":
  {
    "Type": "Notes",
    "Ref": "https://jgm6457/enslatest/csp/support/Demo.SOAP.MathService.cls?wsdl=1",
    "Content": "<definitions targetNamespace='http://tempuri.org'> ...</definitions>",
    "Notes": "Some WSDL"
  },
  "Public": true,
  "Attributes":
  [
    {
      "Name": "One",
      "Value": "1"
    }
  ],
  "Files":
  [
    {
      "Filename": "SOAPMathService.WSDL",
      "FileExtention": ".WSDL",
      "MimeType": "text/text",
      "CharEncoding": "UTF-8",
      "FileSize": "1.44 KB",
      "Contents": null
    }
  ],
  "Actions":
  [
    {
      "Name": "Sum",
      "Ref": "Sum",
      "Verb": "POST",
      "Description": "Add up 2 numbers",
      "ReadOnly": false,
      "Idempotent": true
    }
  ]
},
{
}
```

```

"Name": "PublicREST",
"Domain": "UnitTest",
"Version": "0.9",
"Stage": "Live",
"Protocol": "REST",
"Description": "REST Call for the Public Registry",
"Endpoint": "http://mymachine.mynetwork.com:57774/csp/registry/docserver/public",
"ResponseStyle": "Sync",
"LastModified": "2022-03-05 16:15:33.38",
"Topics":
[
  "Public",
  "Search",
],
"Public": true,
"Attributes":
[
  {
    "Name": "Security",
    "Value": "Username and Password"
  }
],
"Files":
[
  {
    "Filename": "TestPlan",
    "MimeType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
    "CharEncoding": "UTF-8",
    "FileSize": "16.95 KB",
    "Contents": null
  }
],
"Actions":
[
  {
    "Name": "public",
    "Ref": "public",
    "Verb": "GET",
    "Description": "Returns REST endpoint for public registry",
    "RequestSchema":
    {
      "Type": "Notes",
      "Notes": "This is the REST endpoint to query the public Registry"
    },
    "ResponseSchema":
    {
      "Type": "REST Information"
    },
    "ReadOnly": true,
    "Idempotent": true
  }
]
}

```

以下に、サービスの記述に含まれる可能性のあるフィールドのリストと説明を示します。すべてのフィールドについて、ESB サイトには、ESB のサービス・レジストリのフィールドの使用に関する情報を提供するガイドラインと一般的な使用方法が記載されているはずです。

#### [名前]

[名前] は、サービスを特定します。サービスを特定するその他の属性は、[ドメイン] と [バージョン] です。[ドメイン] は通常、サービスのカテゴリを特定し、[バージョン] はバージョン番号を表す文字列です。[名前]、[ドメイン]、および [バージョン] の組み合わせは、サービス・レジストリ・エントリごとに固有です。例えば、ドメインが異なるか、バージョンが異なる限り、複数のサービスの名前が同じである可能性があります。

#### [ドメイン]

[ドメイン] は、サービスのカテゴリを指定します。

#### [バージョン]

[バージョン] は、サービスのバージョンを特定します。

## [段階]

[ライフサイクルの段階] は、サービスの開発状態を表します。これは、以下のいずれかの値を持つことができます。[概念]、[開発]、[テスト]、[ステージング]、[稼働中]、[廃止]、および [非アクティブ]。

## [プロトコル]

[サービス・プロトコル] は、クライアントがサービスへのアクセスに使用するプロトコルを表します。これは、以下のいずれかの値を持つことができます。[File]、[FTP]、[HL7]、[HTTP]、[REST]、[SOAP]、[SQL]、[X12]、および任意のカスタム値。

## [説明]

[説明] は、サービスの簡単な説明を提供します。

## [エンドポイント]

[エンドポイント] は、サービスの場所を指定します。これは、サービスの呼び出しに使用する場所です。

## [応答スタイル]

[応答スタイル] は、サービスが値を返す方法を表します。これは、以下のいずれかの値を持つことができます。[同期]、[コールバック]、[リモート・デポジット]、または空白の値。

## [最終更新]

レジストリ・エントリが ESB レジストリで前回変更された日時です。指定した日時以降に追加または更新されたサービスは、[変更以降のサービスの取得] を呼び出すことによって、検出できます。

## [トピック]

[トピック] は、[単語によるサービスの取得] 呼び出しを使用してサービスを検索する場合に役立つ一連の用語を提供します。

## [連絡先]

[連絡先] は、サービスをサポートする、またはサービスに関連付けられているユーザまたは組織を指定します。[連絡先] には、以下のサブフィールドを含めることができます。

- ・ **[ID]** – そのユーザまたは組織の、名前またはその他の ID を指定します。
- ・ **[タイプ]** – 連絡先がサービスの **[作成者]**、**[コンシューマ]**、**[プロバイダ]**、**[オペレータ]**、**[マネージャ]**、**[スポンサ]** のいずれであるかを指定します。
- ・ **[詳細]** – そのユーザまたは組織の詳細を指定します。
- ・ **[ビジネス・パートナー]** – サービス・レジストリで内部的に使用されるフィールドですが、パブリック API を使用するアプリケーションには有用ではありません。
- ・ **[備考]** – そのユーザまたは組織に関する補足情報を指定します。

## [スキーマ]

[スキーマ] は、サービス・メッセージの構造を表します。[スキーマ] には、構造の定義を表すサブフィールドや、構造の定義を含むサブフィールドが含まれます。[スキーマ] には、以下のサブフィールドを含めることができます。

- ・ **[タイプ]** – スキーマの記述に使用するスキーマ定義メカニズムの名前を指定します。これは、WSDL、XSD、HL7、SEF、X12、AST、EDF、またはカスタムのスキーマ定義メカニズムのいずれかです。
- ・ **[リファレンス]** – スキーマ定義の URL を指定します。

- ・ **[コンテンツ]** – スキーマ定義の完全なテキスト・コンテンツを表します。
- ・ **[備考]** – スキーマに関する補足情報を提供します。
- ・ **[サムネイル]** – スキーマの短い抜粋を提供します。

## [パブリック]

常に真の値を持ちます。サービス・レジストリで内部的に使用されます。

## [属性]

[属性] は、名前と値のペアのリストを指定します。名前と値のペアを使用することによって、ESB サービス・レジストリに、サービスに関する任意の情報を含めることができます。

## [ファイル]

[ファイル] は、レジストリ・エントリに 1 つ以上のファイルを含める方法を表します。例えば、1 つのファイルにサービスに関する説明を格納できます。1 つのレジストリ・エントリ内の各ファイルに一意的なファイル名を付ける必要があります。いずれのサービスの取得呼び出しを実行した場合でも、返される JSON メッセージにはファイルの内容は含まれません。ファイルの内容を取得するには、別の GetFileByID 呼び出しを実行します。[ファイル] の各要素は、以下のサブフィールドから成ります。

- ・ **[ファイル名]** – ファイルの名前を表します。通常、これにはファイル拡張子が含まれます。1 つのレジストリ・エントリ内に、同じファイル名を持つファイルを 2 つ含めることはできません。
- ・ **[ファイル拡張子]** – ファイルの形式と目的に関する情報を提供します。オプション。
- ・ **[MIME タイプ]** – ファイルにアクセスするために使用するアプリケーションに関する情報を提供します。オプション。
- ・ **[文字エンコード]** – 文字エンコーディングに関する情報を提供します。オプション。
- ・ **[ファイル・サイズ]** – ファイル・サイズを表します。オプション。
- ・ **[コンテンツ]** – JSON サービス記述では、このフィールドは常に NULL です。ファイルの内容を取得するには、[ID によるファイルの取得] 呼び出しを実行する必要があります。

## [アクション]

[アクション] は、指定した URL と共に使用できる SOAP アクションまたは REST HTTP 要求メソッドを指定します。SOAP アクションは、WSDL で定義されている情報の概要を提供します。[アクション] には、以下のサブフィールドを含めることができます。

- ・ **[名前]** – レジストリで対象のアクションの特定に使用する名前を指定します。
- ・ **[リファレンス]** – アクションの文字列識別子を指定します。
- ・ **[動詞]** – HTTP 要求メソッドを指定します。通常、これは GET、PUT、POST、または DELETE ですが、任意の HTTP 要求メソッドを指定できます。
- ・ **[説明]** – アクションのテキストによる説明を指定します。
- ・ **[要求スキーマ]** – 受信メッセージ本文の形式を指定します。以下のサブフィールドで構成されます。[タイプ]、[リファレンス]、[備考]、[サムネイル]、および [コンテンツ]。
- ・ **[応答スキーマ]** – 応答メッセージ本文の形式を指定します。以下のサブフィールドで構成されます。[タイプ]、[リファレンス]、[備考]、[サムネイル]、および [コンテンツ]。
- ・ **[読み取り専用]** – チェックを付けると、呼び出しによってサーバの状態は変更されないことが指定されます。



- ・ [べき等] – チェックを付けると、同じ呼び出しの複数回の実行が、1 回の呼び出しと同じ影響をサーバに与えることが指定されます。

## 2.3 [すべてのサービスの取得]

このサービスの取得 REST 呼び出しでは、表示する権限があるすべてのサービス・レジストリ・エントリが返されます。この呼び出しでは、ファイルの内容が省略されることを除いて、各サービスの完全なレジストリ・エントリが返されます。レジストリから、表示する権限があるすべてのサービスが返されます。この呼び出しの URL は次のとおりです。

GET /services

例えば、サーバと最上位ディレクトリを含む完全な URL は、次のようになります。

`https://esb.example.com/registry/services`

## 2.4 [ID によるサービスの取得]

[ID によるサービスの取得] 呼び出しは、識別されたサービスに関する情報を返します。サービスは、名前、ドメイン、およびバージョンで、一意に識別されます。この呼び出しの URL 構文は次のとおりです。

GET /services/id/name/domain/version

例えば、名前が `coffeemaker`、ドメインが `demo`、バージョンが `1.0` のサービスを取得するには、次の呼び出しを使用します。

`https://esb.example.com/registry/services/id/coffeemaker/demo/1.0`

[ID によるサービスの取得] は、指定したサービスについて、[すべてのサービスの取得] で返されるものと同じ情報を返します。そのため、[すべてのサービスの取得] を呼び出した場合、[ID によるサービスの取得] を呼び出す必要はありません。

**注釈** この呼び出しの目的は、1 つのサービスを返すことですが、URL 内の名前、ドメイン、およびバージョンの部分にワイルドカードを指定できます。ワイルドカードを使用すると、この呼び出しは、指定の内容を満たすすべてのサービスを返します。

## 2.5 選択したサービスの取得

レジストリからすべてのサービスを取得したり、ID によって 1 つのサービスを取得したりする代わりに、サービスのサブセットを選択することができます。以下のいずれかの選択条件を指定できます。

- ・ [プロトコル] – リスト内のいずれかのプロトコルを使用するサービスを選択します。プロトコルとしては、以下のいずれかを指定できます。[File]、[FTP]、[HL7]、[HTTP]、[REST]、[SOAP]、[SQL]、[X12]、および任意のカスタム値。
- ・ [段階] – リストで指定した段階にあるサービスを選択します。段階としては、以下のいずれかを指定できます。[概念]、[開発]、[テスト]、[ステージング]、[稼働中]、[廃止]、および [非アクティブ]。
- ・ [バージョン] – 指定したバージョンのサービスを選択します。
- ・ [テキスト検索] – 以下のいずれかのフィールドの、指定したテキストを含むサービスを選択します。[名前]、[ドメイン]、[説明]、[エンドポイント]、および [トピック]。テキストの比較では、大文字小文字は区別されません。

- ・ [変更日] – 指定した日付以降にエントリが変更されているサービスを選択します。

[プロトコルによるサービスの取得] 呼び出しは、リスト内のいずれかのプロトコルに一致するプロトコルを使用するサービスを選択します。この呼び出しの URL 構文は次のとおりです。

GET /services/protocols/protocol-list

以下の呼び出しは、REST プロトコルを使用するすべてのサービスを返します。

https://esb.example.com/registry/services/protocols/REST

以下の呼び出しは、REST プロトコルと SOAP プロトコルのいずれかを使用するすべてのサービスを返します。

https://esb.example.com/registry/services/protocols/REST,SOAP

[段階によるサービスの取得] 呼び出しは、リスト内のいずれかの段階に一致する段階にあるサービスを選択します。この呼び出しの URL 構文は次のとおりです。

GET /services/stages/stage-list

以下の呼び出しは、稼働中の段階にあるすべてのサービスを返します。

https://esb.example.com/registry/services/stages/Live

[バージョンによるサービスの取得] 呼び出しは、指定したバージョンに一致するバージョンのサービスを選択します。この呼び出しの URL 構文は次のとおりです。

GET /services/version/version

例えば、以下の呼び出しは、バージョンが 1.0 のすべてのサービスを返します。

https://esb.example.com/registry/services/version/1.0

[単語によるサービスの取得] 呼び出しは、以下のいずれかのフィールドに、指定の検索テキストが含まれるすべてのサービスを検出します。

- ・ [名前]
- ・ [ドメイン]
- ・ [説明]
- ・ [エンドポイント]
- ・ [トピック]

[単語によるサービスの取得] 呼び出しの URL 構文は次のとおりです。

GET /services/includesword/search-text

例えば、以下の呼び出しは、いずれかのフィールドに “accounts, payable” というテキスト文字列が含まれるすべてのサービスを返します。

https://esb.example.com/registry/services/includesword/accounts,%20payable

[変更以降のサービスの取得] 呼び出しは、特定の日時以降にレジストリ・エントリが変更されているすべてのサービスを返します。この呼び出しの URL 構文は次のとおりです。

GET /services/modifiedsince/date-time

例えば、次の呼び出しは、特定の日時以降にエントリが変更されているすべてのサービスを返します。

https://esb.example.com/registry/services/modifiedsince/2015-02-1%2011:30

日付を年-月-日の形式で指定し、URL スペース・コードの %20 を区切り文字として使用し、時刻を24時間制で時間:分の形式で指定します。

## 2.6 ファイルの内容の取得

[ID によるファイルの取得] 呼び出しは、指定したファイルの内容を返します。テキスト・ファイルとバイナリ・ファイルのいずれかで返すことができます。JSON メッセージやファイル・メタデータを返すことはありません。ファイルの内容だけを返します。この呼び出しの URL 構文は次のとおりです。

```
GET /services/id/name/domain/version/file/filename
```

ここで、name/domain/version は、サービスを特定し、filename は、そのサービス内のファイルの名前を指定します。

例えば、以下の呼び出しは、ファイルの内容を返します。

```
https://esb.example.com/registry/services/id/coffeemaker/demo/1.0/file/HowToCallCoffeemaker.docx
```

注釈 [ID によるファイルの取得] ではワイルドカードを使用することはできません。URL 内の name/domain/version の各部分にワイルドカードを指定することはできますが、指定に一致するサービスが複数ある場合のこの呼び出しの動作は未定義です。



# 3

## パブリック・サービス・レジストリおよび外部サービス・レジストリの管理

ここでは、パブリック・サービス・レジストリおよび外部サービス・レジストリを管理する方法を説明します。これらのレジストリは目的は異なりますが、構造とユーザ・インタフェースは似ています。

### 3.1 パブリック・サービス・レジストリの管理

パブリック・サービス・レジストリを管理するには、[Interoperability]→[構成する]→[パブリックサービス・レジストリ]の順に選択します。パブリック・サービス・レジストリは、行レベルのセキュリティで保護されているため、パブリック・サービス・レジストリにアクセスするには、%EnsRole\_RegistryManager ロールを有している必要があります。

The screenshot displays the 'Public-Service Registry' management interface. On the left, there are navigation buttons: 'New Service', 'Previous', and 'Next'. Below these are search and filter options, including 'Search', 'Reset', 'Page Size' (set to 50), 'Page' (set to 1), 'Sort By' (Name), and 'Descending' checkbox. There are also dropdowns for 'Version' and 'Visible', and a 'Required Role' dropdown. A 'Word Match' text box is present with a description: 'Text contained in a service's name, domain, description, endpoint, or topics list.' Below these are expandable sections for 'Service Protocols', 'Lifecycle Stages', 'Extended Criteria', and 'Additional Attributes to Display'.

The main area shows a table of services:

Name	Domain	Version	Service Protocol	Lifecycle Stage	Visible
MathServiceREST	UnitTest	1.0	REST	Live	True
MathServiceSOAP	UnitTest	0.9	SOAP	Live	True
PublicREST	UnitTest	3.9a	REST	Live	True

On the right, the 'Details' tab is active, showing the details for the selected service 'MathServiceREST'. The details include: Name: MathServiceREST, Domain: UnitTest, Version: 1.0, Last modified: 2015-06-28 23:08:35.913. There are dropdowns for 'Service Protocol' (REST) and 'Lifecycle Stage' (Live). The 'Endpoint' is 'https://jgm6457/supportrest/Demo.REST.MathService/sum'. The 'Description' is 'Add 2 Numbers'. There are also dropdowns for 'Topics' (Maths, Search) and 'Response Style' (Sync). Below these are expandable sections for 'Internal Information', 'Schema', 'Actions', 'Attributes', and 'Contacts'.

パブリック・レジストリのフィールドの詳細は、“パブリックと外部の両方のサービス・レジストリのフィールド” および “パブリック・サービス・レジストリの内部フィールド” を参照してください。サービス・レジストリのエントリの作成および保守管理

の詳細は、“[サービス・レジストリ・エントリの作成と保守管理](#)”を参照してください。サービス・レジストリの検索の詳細は、“[サービス・レジストリの検索と表示](#)”を参照してください。

## 3.2 外部サービス・レジストリの管理

外部サービス・レジストリを管理するには、[Interoperability]→[構成する]→[外部サービス・レジストリ]の順に選択します。

The screenshot displays the 'External-Service Registry' management interface. On the left, there are navigation buttons: 'New Service', 'Previous', and 'Next'. Below these are search and filter options, including 'Search', 'Reset', 'Page Size' (set to 50), 'Page' (set to 1), 'Sort By' (Name), and 'Descending' checkbox. There are also sections for 'Service Protocols', 'Lifecycle Stages', 'Extended Criteria', and 'Additional Attributes to Display'. The main area shows a table of services with columns: Name, Domain, Version, Service Protocol, and Lifecycle Stage. The table lists various services like Activity, Statistics, Directions, Location, Licenses, Software, MacWinNamespaces, Dev, Math, UT, Movies, Entertainment, Sky, Weather, and Temperature. On the right, the 'Details' panel for the selected service 'Activity' is shown, including fields for Name, Domain (Statistics), Version (0.9), Last modified (2015-03-08 14:31:29.662), Service Protocol (REST), Lifecycle Stage (Concept), Endpoint (https://tysonsmbpnone/csp/msgbank/activity/data), Description (Post base 64 encoded list of activity data. Not idempotent), Topics (Statistics), and Response Style (Sync). There are also sections for Schema, Actions, Attributes, Contacts, and Files.

Name	Domain	Version	Service Protocol	Lifecycle Stage
Activity	Statistics	0.9	REST	Concept
Activity	Statistics	1	SOAP	Concept
Directions	Location	1	SOAP	Concept
Directions	Location	1.1b	REST	Concept
Licenses	Software	ABC	REST	Concept
MacWinNamespaces	Dev	0.1	REST	Concept
Math	UT	1	REST	Concept
Math	UT	1.1	FTP	Concept
Math	UT	2	File	Concept
Movies	Entertainment	2		Concept
Sky	Weather	0.5	REST	Concept
Temperature	Weather	A	REST	Concept

外部レジストリのフィールドの詳細は、[パブリックと外部の両方のサービス・レジストリのフィールド](#)を参照してください。サービス・レジストリのエントリの作成および保守管理の詳細は、[サービス・レジストリ・エントリの作成と保守管理](#)を参照してください。サービス・レジストリの検索の詳細は、[サービス・レジストリの検索と表示](#)を参照してください。

## 3.3 パブリックと外部の両方のサービス・レジストリのフィールド

この節では、パブリック・サービス・レジストリと外部サービス・レジストリのフィールドについて説明します。パブリック・サービス・レジストリと外部サービス・レジストリではほとんどのフィールドが同じです。両方のレジストリに共通のフィールドに加えて、パブリック・サービス・レジストリにはプライベート・フィールドがいくつかあります。このようなフィールドには、レジストリ管理者のみがアクセスでき、パブリック REST API を使用してアクセスすることはできません。

### [名前]

[名前] は、サービスを特定します。サービスを特定するその他の属性は、[ドメイン]と[バージョン]です。それぞれのレジストリは、これらの属性の使用法に関する規則を採用する必要があります。[ドメイン]は通常、サービスのカテゴリを特定し、[バージョン]はバージョン番号を表す文字列です。[ドメイン]と[バージョン]が組み合わさった[名前]は、エントリごとに一意である必要があります。例えば、ドメインが異なるか、バージョンが異なる限り、複数のエントリの名前が同じである可能性があります。

### [バージョン]

[バージョン] は、サービスのバージョンを特定します。

**[ドメイン]**

[ドメイン] は、サービスのカテゴリを指定します。

**[ライフサイクルの段階]**

[ライフサイクルの段階] は、サービスの開発状態を表します。これは、以下のいずれかの値を持つことができます。[概念]、[開発]、[テスト]、[ステージング]、[稼働中]、[廃止]、および [非アクティブ]。

**[サービス・プロトコル]**

[サービス・プロトコル] は、クライアントがサービスへのアクセスに使用するプロトコルを表します。これは、以下のいずれかの値を持つことができます。[File]、[FTP]、[HL7]、[HTTP]、[REST]、[SOAP]、[SQL]、[X12]、および任意のカスタム値。外部サービス・レジストリでは、Service Protocol によって、ビジネス・オペレーションのプロパティの設定に Endpoint をどのように使用するかが決まります。サービス・プロトコルとして TCP を明示的に入力することができますが、ドロップダウンの選択肢としては一覧に含まれていません。

**[説明]**

[説明] は、サービスの簡単な説明を提供します。

**[エンドポイント]**

[エンドポイント] は、サービスの場所を指定します。パブリック・サービス・レジストリでは、エンドポイントは通常、ESB 上のビジネス・サービスの URL です。外部サービス・レジストリでは、エンドポイントは、データまたはサービスの場所に関する情報を提供し、ESB ホストのプロパティの設定に使用されます。

外部サービス・レジストリのエンドポイントの形式は、サービス・プロトコルによって異なります。それぞれのタイプのサービス・プロトコルでの Endpoint の使用方法の詳細は、["ESB ホストの構成での外部サービス・レジストリの使用"](#) を参照してください。

**[応答スタイル]**

[応答スタイル] は、サービスが値を返す方法を表します。これは、以下のいずれかの値を持つことができます。[同期]、[コールバック]、[リモート・デポジット]、または空白の値。

**[トピック]**

[トピック] では、管理者ページおよびパブリック API からレジストリ・エントリの検索に役立つ検索用語を定義できます。レジストリ・エントリの作成時に値を入力することはできませんが、[詳細] パネルで値を入力することによって追加できます。カンマで区切った用語のリストを入力するか、ドロップダウン・メニューから 1 つ以上のトピックにチェックを付けることによってトピックを指定できます。ドロップダウン・メニューには、レジストリ・エントリに既に定義されているトピックが一覧表示されます。

**[スキーマ]**

[スキーマ] は、サービス・メッセージの構造を表します。[スキーマ] には、構造の定義を表すサブフィールドや、構造の定義を含むサブフィールドが含まれます。レジストリ・エントリの作成時にスキーマを指定することはできませんが、[詳細] パネルで値を入力することによってスキーマを追加できます。[スキーマ] には、以下のサブフィールドが含まれます。

- ・ **[タイプ]** – スキーマの記述に使用するスキーマ定義メカニズムの名前を指定します。WSDL、XSD、HL7、SEF、X12、AST、または EDF を選択することも、その他のスキーマ定義メカニズムの名前を入力することもできます。
- ・ **[リファレンス]** – スキーマ定義の URL を指定します。
- ・ **[備考]** – スキーマに関する補足情報を提供します。
- ・ **[サムネイル]** – スキーマの短いテキスト抜粋を提供します。

- ・ **[コンテンツ]** – スキーマ定義の完全なテキスト・コンテンツを表します。

## [アクション]

[アクション] は、指定した URL と共に使用できる SOAP アクションまたは REST HTTP 要求メソッドを指定します。SOAP アクションは、WSDL で定義されている情報の概要を提供します。レジストリ・エントリの作成時にアクションを追加することはできませんが、[詳細] パネルで [アクション] のプラス記号を選択することによってアクションを追加できます。[アクション] には、以下のサブフィールドが含まれます。

- ・ **[名前]** – レジストリで対象のアクションの特定に使用する名前を指定します。
- ・ **[リファレンス]** – アクションまたは Web メソッドの識別子です。
- ・ **[動詞]** – HTTP 要求メソッドを指定します。通常、これは GET、PUT、POST、または DELETE ですが、任意の HTTP 要求メソッドを入力できます。
- ・ **[読み取り専用]** – チェックを付けると、呼び出しによってサーバの状態は変更されないことが指定されます。
- ・ **[べき等]** – チェックを付けると、同じ呼び出しの複数回の実行が、1 回の呼び出しと同じ影響をサーバに与えることが指定されます。
- ・ **[説明]** – アクションのテキストによる説明を指定します。
- ・ **[要求スキーマ]** – 受信メッセージ本文の形式を指定します。以下のサブフィールドで構成されます。[タイプ]、[リファレンス]、[備考]、[サムネイル]、および [コンテンツ]。
- ・ **[応答スキーマ]** – 応答メッセージ本文の形式を指定します。以下のサブフィールドで構成されます。[タイプ]、[リファレンス]、[備考]、[サムネイル]、および [コンテンツ]。

## [属性]

[属性] は、名前と値のペアのリストを指定します。名前と値のペアを使用して、レジストリ・エントリの任意のフィールドを指定できます。レジストリ・エントリの作成時に属性を追加することはできませんが、[詳細] パネルで [属性] の + (プラス記号) を選択することによって属性を追加できます。それぞれの属性は、名前と文字列の値で構成されます。属性を定義すると、値を更新することも、その属性を削除することもできますが、属性の名前を変更することはできません。

## [連絡先]

[連絡先] は、サービスをサポートまたは使用するユーザまたは組織を指定します。レジストリ・エントリの作成時に連絡先を追加することはできませんが、[詳細] パネルで [連絡先] のプラス記号を選択することによって連絡先を追加できます。[連絡先] には、以下のサブフィールドが含まれます。

- ・ **[ID]** – そのユーザまたは組織の、名前またはその他の ID を指定します。
- ・ **[連絡先タイプ]** – 連絡先がサービスの [作成者]、[コンシューマ]、[プロバイダ]、[オペレータ]、[マネージャ]、[スポンサ] のいずれであるかを指定します。
- ・ **[ビジネス・パートナー]** – ビジネス・パートナー・オブジェクトへのリンクを示します。このオブジェクトは、住所や電話番号などの連絡先情報を提供します。パブリック API はビジネス・パートナー名のみを示し、ビジネス・パートナー・オブジェクトからの連絡先情報は含みません。
- ・ **[詳細]** – 連絡先に関する詳細を指定します。例えば、電話番号、電子メール・アドレス、またはパブリック API からアクセス可能なその他の連絡先情報を入力できます。
- ・ **[備考]** – そのユーザまたは組織に関する補足情報を指定します。

連絡先を定義すると、[ID] を除くすべてのサブフィールドを更新できます。赤い X を選択することによって、連絡先を削除することもできます。



**[ファイル]**

[ファイル] は、レジストリ・エントリに任意のテキスト・ファイルまたはバイナリ・ファイルを格納する方法を表します。例えば、サービスに関する説明や、大きなサイズのスキーマ定義が含まれるファイルを格納できます。サービスの作成時にファイルを追加することはできませんが、**[詳細]** パネルで [ファイル] のプラス記号を選択することによってファイルを追加できます。ローカル・システムまたは InterSystems IRIS® サーバ上にあるファイルを追加できます。[ファイル] には、以下のサブフィールドが含まれます。

- ・ **[ファイル名]** – ファイルの名前を表します。レジストリ内のファイルの名前を、ローカル・ファイル・システムまたは InterSystems IRIS® サーバ上の元のファイルの名前と同じにする必要はありません。
- ・ **[ファイル拡張子]** – ファイルの形式と目的に関する情報を提供します。このフィールドは、ファイル名の最後の部分で設定されます。
- ・ **[MIME タイプ]** – ファイルにアクセスするために使用するアプリケーションに関する情報を提供します。
- ・ **[文字エンコード]** – 文字エンコーディングに関する情報を提供します。
- ・ **[ファイル・サイズ]** – ファイル・サイズを表します。このフィールドは、ファイルの内容から計算されます。
- ・ **[コンテンツ]** – ファイルの作成時に、このサブフィールドはファイルの内容に設定されます。

## 3.4 パブリック・サービス・レジストリの内部フィールド

以下に、内部属性を示します。これらはパブリック・サービス・レジストリでのみ使用されます。これらの内部属性には、管理ポータルを使用している管理者のみがアクセスできます。これらの属性には、サービス・レジストリからパブリック API を使用してアクセスすることはできません。

**[パブリック]**

このチェック・ボックスは、対象のサービスに関する情報をレジストリのパブリック API から取得できるかどうかを制御します。**[パブリック]** が真の場合 (ボックスにチェックが付いている場合)、**[必要なロール]** の設定に基づいてユーザが表示を許可されているなら、API は対象のサービスに関する情報を返します。**[パブリック]** が偽の場合、パブリック API から、対象のサービスに関する情報は何も返されません。

**[必要なロール]**

**[必要なロール]** は、レジストリ・エントリへのパブリック API アクセスを提供するロールのリストを指定します。ユーザがレジストリ・エントリに関する情報を取得するには、そのユーザには **[必要なロール]** のリストに含まれているいずれかのロールが必要です。ユーザは、InterSystems IRIS へのログインを通じて、あるいはロールを有した Web アプリケーションを通じて、指定のロールを取得します。これらのロールの定義方法の詳細は、["パブリック・サービス・レジストリに対するロールとユーザの定義"](#) を参照してください。

**[インスタンス]**

**[インスタンス]** は、指定されたエンドポイントでサービスを提供している InterSystems IRIS インスタンスを特定します。通常、これは、ESB レジストリおよびサービス・レジストリに使用される InterSystems IRIS インスタンスです。ビジネス・サービスからエンドポイント・サービスが提供されていない場合、このフィールドは空白のままになります。

#### [ネームスペース]

[ネームスペース]は、指定されたエンドポイントでサービスを提供している相互運用対応ネームスペースを特定します。通常、これは、ESB レジストリおよびサービス・レジストリに使用される相互運用対応ネームスペースです。ビジネス・サービスからエンドポイント・サービスが提供されていない場合、このフィールドは空白のままにします。

#### [プロダクション]

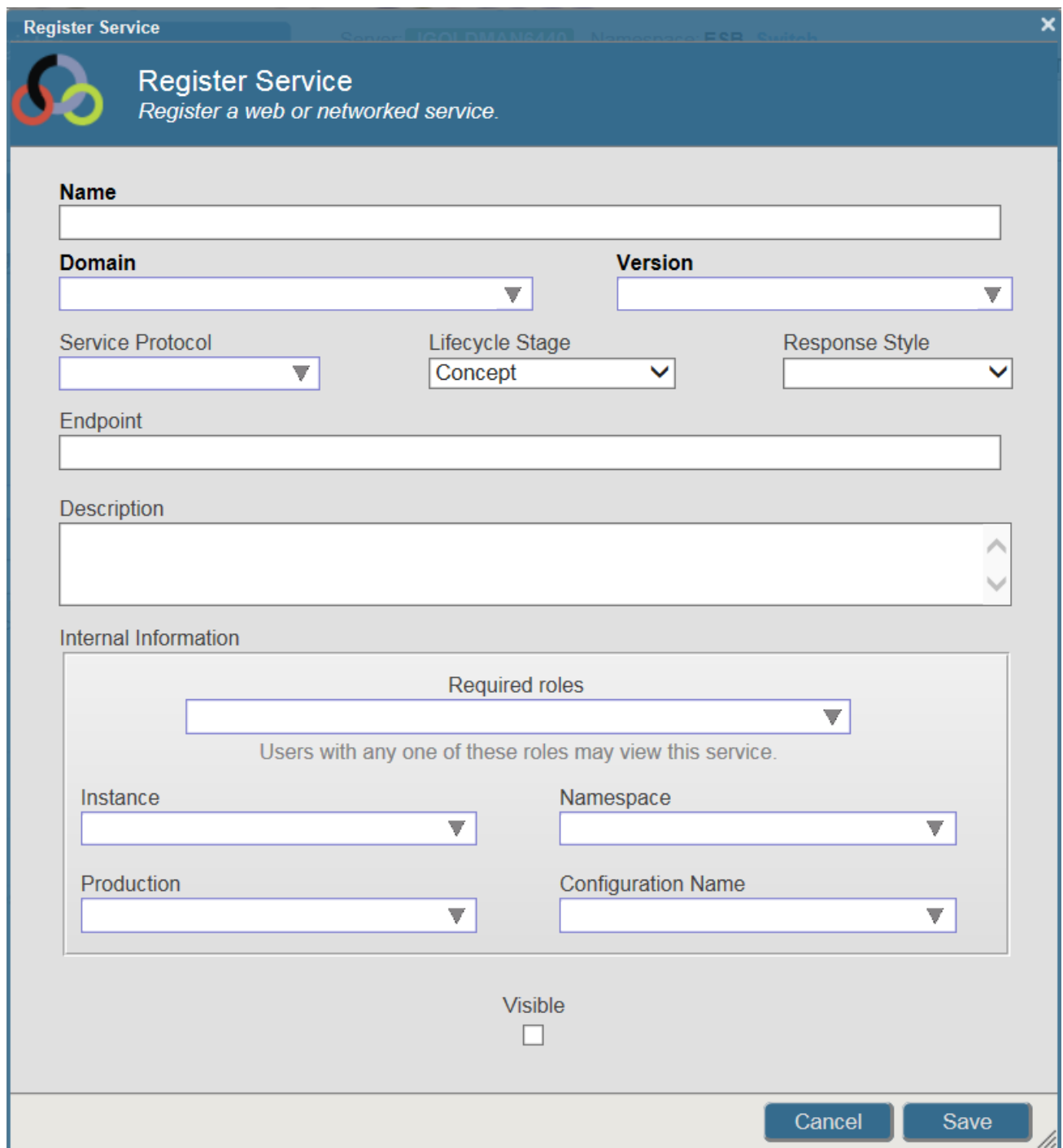
[プロダクション]は、指定されたエンドポイントでサービスを提供しているプロダクションを特定します。通常、これは ESB プロダクションです。ビジネス・サービスからエンドポイント・サービスが提供されていない場合、このフィールドは空白のままにします。

#### [構成名]

[構成名]は、指定されたエンドポイントでサービスを提供しているプロダクション構成項目を特定します。通常、この構成項目は、ESB プロダクション内のビジネス・サービスです。ビジネス・サービスからエンドポイント・サービスが提供されていない場合、このフィールドは空白のままにします。

## 3.5 サービス・レジストリ・エントリの作成と保守管理

パブリック・サービス・レジストリまたは外部サービス・レジストリの新しいエントリを作成するには、[新しいサービス]を選択します。新しいサービス・エントリをパブリック・サービス・レジストリに作成する場合、InterSystems IRIS から以下の [サービスの登録] フォームが表示されます。



The image shows a 'Register Service' dialog box with the following fields and controls:

- Name:** A text input field.
- Domain:** A dropdown menu.
- Version:** A dropdown menu.
- Service Protocol:** A dropdown menu.
- Lifecycle Stage:** A dropdown menu with 'Concept' selected.
- Response Style:** A dropdown menu.
- Endpoint:** A text input field.
- Description:** A text area with scrollbars.
- Internal Information:** A section containing:
  - Required roles:** A dropdown menu.
  - Users with any one of these roles may view this service.** (Text label)
  - Instance:** A dropdown menu.
  - Namespace:** A dropdown menu.
  - Production:** A dropdown menu.
  - Configuration Name:** A dropdown menu.
- Visible:** A checkbox.
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right.

新しいサービス・エントリを外部サービス・レジストリに作成する場合、[内部情報] のフィールドと [表示] フィールドがない、同様のフォームが InterSystems IRIS から表示されます。

新しいサービス・エントリを作成する場合、必須フィールドである [名前]、[ドメイン]、および [バージョン] に値を入力してから、[保存] ボタンを押します。これら 3 つのフィールドの値は、レジストリ・エントリの保存後に変更することはできません。これらのフィールドに新しい値を入力したレジストリ・エントリのコピーを作成することができ、そのためには、対象のレジストリ・エントリを選択してから、[アクション] パネルの [名前を付けて保存] を選択します。

その他のプロパティについて、[サービスの登録] フォームで値を入力できます。

- ・ [サービス・プロトコル]
- ・ [ライフサイクルの状態]

- ・ [応答スタイル]
- ・ [エンドポイント]
- ・ [説明]

パブリック・サービス・レジストリのエントリを作成する場合、[内部情報] のフィールドにも値を入力できます。

- ・ [必要なロール]
- ・ [インスタンス]
- ・ [ネームスペース]
- ・ [プロダクション]
- ・ [構成名]
- ・ [表示]

レジストリ・エントリを保存したら、リストからそのレジストリ・エントリを選択し、[詳細] パネルを使用して、3 つの必須フィールド以外のこれらいずれかのフィールドを変更することができます。[詳細] パネルでは、以下のフィールドの入力または変更を行うこともできます。

- ・ [トピック]
- ・ [スキーマ]
- ・ [アクション]
- ・ [属性]
- ・ [連絡先]
- ・ [ファイル]

[アクション]、[属性]、[連絡先]、および [ファイル] フィールドは複数の値を取ります。新しい項目を追加するには、[アクション追加]、[属性追加]、[連絡先追加]、または [ファイル追加] のプラス記号を選択します。フォームに入力して、[適用] を選択します。既存の項目を更新するには、その項目のクリップボードを選択します。項目を削除するには、その項目の赤い X を選択します。

[アクション] パネルを使用すると、レジストリ・エントリの削除、以下の 1 つ以上の必須フィールドに変更を加えたコピーの保存を実行できます。[名前]、[ドメイン]、および [バージョン]。

## 3.6 サービス・レジストリの検索と表示

[次] ボタンと[前] ボタン、および検索パネルを使用して、レジストリ・エントリを表示できます。1 ページに収まらない数のレジストリ・エントリがある場合は、[次] ボタンと[前] ボタンを使用して、エントリ全体をスクロールできます。検索パネルでは、クエリに一致するレジストリ・エントリを検索できます。

New Service Previous Next

Search
Reset
«

Page Size

Page

Sort By

☐ Descending

Version

Visible

Required Role

Word Match

Text contained in a service's name, domain, description, endpoint, or topics list.

▼ Service Protocols

☒ File

☒ HTTP

☒ SQL

☒ FTP

☒ REST

☒ X12

☒ HL7

☒ SOAP

☒ Other

▼ Lifecycle Stages

☒ Concept

☒ Live

☒ Defunct

☒ Staging

☒ Deprecated

☒ Test

☒ Development

▼ Extended Criteria

Add Criterion
 Add OR

▼ Additional Attributes to Display

Modify Display

検索パネルでは、以下の操作を実行できます。

- ・ [ページ・サイズ] を指定できます。ここには、1 ページに表示できるレジストリ・エントリの最大数を指定します。
- ・ 名前、ドメイン、バージョン、サービス・プロトコル、ライフサイクルの段階、および最終更新日時で、レジストリ・エントリをソートできます。昇順 (デフォルト) または降順でソートできます。
- ・ 以下のいずれかの条件を指定して、表示するレジストリ・エントリを選択できます。
  - － 指定したバージョン
  - － 表示または非表示のエントリ

- 指定した必要なロール
  - 以下のいずれかのフィールドの【単語一致】内のテキスト:[名前]、[ドメイン]、[説明]、[エンドポイント]、または [トピック]
  - 選択したいいずれかのサービス・プロトコル
  - 選択したいいずれかのライフサイクルの段階
  - 属性値に基づいて条件を指定する拡張条件
- ・ 【表示の変更】フィールドを使用して、サービスのリストに、特定の属性の値を表示することもできます。

# 4

## ESB の構成

ここでは、InterSystems IRIS® システムを ESB として構成する方法を説明します。

### 4.1 ESB としての InterSystems IRIS の構成の概要

この節では、InterSystems IRIS を ESB として構成するために必要なタスクの概要を示します。各タスクの詳細は、他の節で示しています。

- ・ ESB プロダクションおよびサービス・レジストリを格納するための相互運用対応ネームスペースを作成して、Web ゲートウェイを構成します。詳細は、“[InterSystems IRIS システムの構成とネームスペースの作成](#)”を参照してください。
- ・ ESB で使用する外部サービスのエンドポイントを定義する外部サービス・レジストリのエントリを作成します。詳細は、“[パブリック・サービス・レジストリおよび外部サービス・レジストリの管理](#)”を参照してください。
- ・ ESB プロダクションを作成し、ビジネス・サービス、およびそれらのサービスを提供するビジネス・オペレーションを追加して構成し、プロダクションを開始します。サービスとオペレーションの構成の詳細は、“[ESB のサービスおよびオペレーションの構成](#)”を参照してください。
- ・ クライアントでビジネス・サービスを使用できるようにするために必要な Web アプリケーションを作成します。詳細は、“[Web アプリケーションの構成](#)”を参照してください。
- ・ 管理ポータルからパブリック REST API を使用してパブリック・サービス・レジストリにアクセスするときに必要なロールとユーザを作成します。詳細は、“[パブリック・サービス・レジストリに対するロールとユーザの定義](#)”を参照してください。
- ・ Web アプリケーションを作成して、パブリック・サービス・レジストリに対して REST API を有効にします。詳細は、“[パブリック・サービス・レジストリ API への Web アプリケーションの構成](#)”を参照してください。
- ・ ESB からアクセス可能なサービスを記述するパブリック・サービス・レジストリのエントリを作成します。詳細は、“[パブリック・サービス・レジストリおよび外部サービス・レジストリの管理](#)”を参照してください。

### 4.2 パブリック・サービス・レジストリに対するロールとユーザの定義

レジストリ内の情報に誰もがアクセスできるように ESB を構成することも、個々のレジストリ・エントリによるアクセスを特定のアカウントに限定することもできます。

パブリック・サービス・レジストリの一部またはすべてのエントリへのアクセスが開かれるようにするには、以下の操作を行います。

1. 以下のいずれかの操作を実行して、ロールを作成します。
  - a. **[システム管理]**→**[セキュリティ]**→**[ロール]** の順に選択して、**[ロール]** ポータル・ページを表示します。
  - b. **[新規ロール作成]** ボタンを選択して、ロールに名前を付け (例 : ServReg\_Unauthenticated)、**[保存]** ボタンを選択します。
2. 以下の操作を実行して、パブリック・サービス・レジストリ向けの Web アプリケーションを編集または作成します。
  - a. **[許可された認証方法]** の **[認証なし]** チェックボックスにチェックを付けます。
  - b. 前の手順で定義したロールを **[アプリケーションロール]** に追加します。

追加の必要がある他のロールの追加の詳細は、“[パブリック・サービス・レジストリ API への Web アプリケーションの構成](#)” を参照してください。

3. 認証されていないユーザがアクセスできるようにするパブリック・サービス・レジストリのエントリごとに、手順 1 で定義したロールを **[必要なロール]** フィールドに追加します。これは、**[内部情報]** 内のカンマ区切りのリストです。

認証なしのアクセスを許可するには、ロールの定義だけで必要で、ユーザを定義する必要はありません。

パブリック・サービス・レジストリのエントリへのアクセスを 1 人以上のユーザに制限するには、以下の操作を行います。

1. 以下のいずれかの操作を実行して、ロールを作成します。
  - a. **[システム管理]**→**[セキュリティ]**→**[ロール]** の順に選択して、**[ロール]** ポータル・ページを表示します。
  - b. **[新規ロール作成]** ボタンを選択して、ロールに名前を付け (例 : ServReg\_IDServices)、**[保存]** ボタンを選択します。
2. 以下の操作を実行して、パブリック・サービス・レジストリに対して Web アプリケーションを編集または作成します。
  - a. **[許可された認証方法]** の **[パスワード]** チェックボックスにチェックを付けます。
  - b. 前の手順で定義したロールを **[アプリケーションロール]** に追加しないでください。

追加の必要がある他のロールの追加の詳細は、“[パブリック・サービス・レジストリ API への Web アプリケーションの構成](#)” を参照してください。

3. このロールを持つユーザがアクセスできるようにするパブリック・サービス・レジストリのエントリごとに、このロールを **[許可されたロール]** フィールドに追加します。これは、カンマ区切りのリストです。
4. 以下のように、ユーザ・アカウントを作成または編集して、ロールをユーザに割り当てます。
  - a. **[システム管理]**→**[セキュリティ]**→**[ユーザ]** の順に選択して、**[ユーザ]** ポータル・ページを表示します。
  - b. **[新規ユーザ作成]** ボタンを選択して、ユーザに名前を付け、パスワードなどの関連フィールドに入力し、**[保存]** ボタンを選択するか、既存のユーザを選択して編集します。
  - c. **[ロール]** タブで、作成したロールを 1 つまたは複数選択し、右矢印ボタンを選択して、**[割り当てる]** ボタンを選択します。

パブリック・サービス・レジストリ REST API にアクセスするために、ユーザにその他のロールを割り当てる必要はありません。サービス自体にアクセスできるように、追加のロールの割り当てが必要な場合があります。

**注釈** レジストリ・エントリへのアクセスの制限または許可は、そのレジストリ・エントリによって記述されるサービスへのアクセスの制限または許可とはまったく無関係です。サービス自体へのアクセスの制御は、そのサービスを起動する Web アプリケーションへのアクセスを制御するか、そのサービスでカスタム・コードを使用することによって実現できます。



## 4.3 パブリック・サービス・レジストリ REST API への Web アプリケーションの構成

この節では、パブリック・サービス・レジストリ REST API に Web アプリケーションを構成する方法を説明します。

1. Web ポートへの呼び出しを処理する Web アプリケーションを定義します。この Web アプリケーションの名前によって、サービスを呼び出す URL のルートが指定されます。単一の Web アプリケーションは複数のビジネス・サービスをサポートできますが、これらのビジネス・サービスはすべて、同じクラスを持っているか、その Web アプリケーションのディスパッチ・クラスのサブクラスを持っている必要があります。
  - a. [システム管理]→[セキュリティ]→[アプリケーション]→[Web アプリケーション] の順に選択して、[Web アプリケーション] ポータル・ページを表示します。[新規 Web アプリケーション作成] ボタンを選択します。
  - b. この Web アプリケーションに名前を付けます (例: /services)。名前の先頭には / (スラッシュ) を付加する必要があります。名前の中のアルファベット文字はすべて小文字にする必要があります。
  - c. [ネームスペース] で、当該サービス・レジストリが実行されているネームスペースを指定します (例: SERVICESNS)。[ネームスペースのデフォルト・アプリケーション] チェックボックスのチェックは外したままにします。
  - d. [アプリケーション]、[Web ファイル]、および [着信 Web サービス] の各チェック・ボックスにはチェックを付けてもかまいません。
  - e. [必須リソース] フィールドと [ID でグループ分け] フィールドは空白のままにします。
  - f. ログインするユーザがこのレジストリにアクセスできるようにするには、[許可された認証方法] の [パスワード] チェックボックスにチェックを付けます。ログオンなしにユーザがこのレジストリにアクセスできるようにするには、[認証なし] チェックボックスにチェックを付けます。
  - g. [ディスパッチ・クラス] を `Ens.ServiceRegistry.API.REST.Query` コンポーネント・クラスに設定します。
  - h. [保存] を選択します。
  - i. [アプリケーションロール] タブを選択します。
  - j. [利用可能] ボックスで、以下のロールを選択します。
    - ・ %EnsRole\_RegistrySelect
    - ・ ネームスペースのグローバル用データベースおよびルーチン用データベースに定義されているロール (例: %DB\_SERVICESNSG や %DB\_SERVICESNSR)
    - ・ ユーザがログインに使用するアカウントに関係なく、すべてのユーザがアクセスできるようにするパブリック・サービス・レジストリに定義されているロール。非認証アクセスを許可するよう Web アプリケーションを構成している場合は、これらのロールを少なくとも 1 つ指定する必要があります。

複数のロールを選択するには、Ctrl キーを押しながら選択します。

- k. 選択したロールがハイライト表示された状態で、右矢印ボタンを選択してこれらのロールを [選択済み] テキスト・ボックスに移動します。
- l. [割り当てる] ボタンを選択します。

**注釈** Web アプリケーションで非認証サービスと認証済みサービスの両方を許可する場合は、パスワードの入力が求められますが、パスワードを入力しないと、Web アプリケーションによって非認証アクセスが許可されます。

## 4.4 ESB ホストの構成での外部サービス・レジストリの使用

ESB プロダクションと外部サービス・レジストリが同じネームスペースにある場合、そのレジストリを使用して ESB ホストを構成できます。これによって、レジストリ内で外部サービスに関する情報を一元化できます。ほとんどの場合、このようなレジストリは、外部サービスにアクセスするビジネス・オペレーションのプロパティの設定に使用しますが、FTP サーバから受信メッセージを取得する FTP 受信アダプタに基づくサービスなどのビジネス・サービスにこのようなレジストリを使用することもできます。

外部サービス・レジストリを使用して、以下のいずれかのアダプタを実装するホストのプロパティを設定できます。

- ・ EnsLib.FTP.InboundAdapter
- ・ EnsLib.FTP.OutboundAdapter
- ・ EnsLib.File.InboundAdapter
- ・ EnsLib.File.OutboundAdapter
- ・ EnsLib.HTTP.OutboundAdapter
- ・ EnsLib.SOAP.OutboundAdapter
- ・ EnsLib.SQL.InboundAdapter
- ・ EnsLib.SQL.OutboundAdapter
- ・ EnsLib.TCP.OutboundAdapter

設定を外部サービス・レジストリから取得するようホストを構成するには、**[外部レジストリ ID]** フィールドでそのレジストリ・エントリのキー値を指定します。キー値は、サービスの名前、ドメイン、およびバージョンが、それぞれの要素を区切る || (2 本の縦棒) で連結されたものから成ります。例えば、名前が GetDaily、ドメインが Weather、バージョンが 1.0 の外部サービス・レジストリ・エントリからプロパティ値を取得するには、**[外部レジストリ ID]** プロパティを以下のように設定します。

GetDaily||Weather||1.0

次に、エンドポイント値を使用して、以下の各プロトコルにアダプタのプロパティを設定する方法を説明します。

- ・ HTTP および REST – エンドポイントで HTTPServer プロパティ、HTTPPort プロパティ、および URL プロパティが設定されます。例えば、エンドポイントが以下の場合

`http://newsrvcs.example.com:57781/csp/samples/docserver/namespaces`

ビジネス・オペレーションのプロパティは以下のように設定されます。

HTTPServer : newsrvcs.example.com

HTTPPort: 57781

URL: /csp/samples/docserver/namespaces

ビジネス・オペレーションがパススルー・ビジネス・オペレーションである場合、特殊文字 \* (アスタリスク)、^ (キャレット)、および | (縦棒) を使用して、受信 URL からエンドポイントを構築する方法を記述します。詳細は、“[パススルー・ビジネス・オペレーションの構成](#)” を参照してください。

- ・ FTP – エンドポイントで FTPServer プロパティ、FTPPort プロパティ、FilePath プロパティ、および SSLConfig プロパティが設定されます。例えば、エンドポイントが以下の場合

`ftp://newftp.example.com/filestore/docs/`

ビジネス・オペレーションのプロパティは以下のように設定されます。

FTPServer: newftp.example.com

FTPPort: 21

FilePath: /filestore/docs/

プロトコルが `sftp://` である場合、`SSLConfig` は `"!SFTP"` に設定され、`FTPPort` は 22 に設定されます。

- ・ SQL – エンドポイントは、`DSN` プロパティの設定に使用されます。
- ・ SOAP – エンドポイントは、`WebServiceURL` プロパティの設定に使用されます。このため、**EnsLib.SOAP.GenericOperation** および **EnsLib.SOAP.GenericOperationInProc** パススルー・オペレーションは、SOAP オペレーションではなく HTTP オペレーションと見なされます。
- ・ File – エンドポイントは、`FilePath` プロパティの設定に使用されます。
- ・ TCP – エンドポイントは、`IPAddress` プロパティおよび `Port` プロパティの設定に使用されます。例えば、エンドポイントが以下の場合

`102.23.10.1:4500`

ビジネス・オペレーションのプロパティは以下のように設定されます。

`IPAddress: 102.23.10.1`

`Port: 4500`

- ・ HL7 および X12 – エンドポイント文字列の先頭で、基盤となる以下のプロトコルが指定されます。File、TCP、FTP、または HTTP。File および TCP の場合、解析の前にエンドポイントの文字列から `file:` および `tcp:` が削除されます。

外部サービス・レジストリ・エントリから取得したビジネス・オペレーションの設定によって、システムのデフォルト設定、プロダクション定義、およびビジネス・ホスト・クラス定義のいずれの設定も上書きされます。



# 5

## ESB のサービスおよびオペレーションの構成

ここでは、ESB プロダクションでサービスとオペレーションを構成する方法、およびオペレーションが提供するサービスを使用する方法を説明します。

### 5.1 パススルー・ビジネス・サービスの構成

パススルー・ビジネス・サービスを追加するには、[プロダクション構成] ページで新規サービスを追加するための + 記号を選択してから、以下の手順を実行します。

1. 使用されているプロトコルの種類および SAML セキュリティが使用されているかどうかに基づいてクラスを選択します。以下のいずれかのクラスを選択します。
  - ・ `EnsLib.HTTP.GenericService`
  - ・ `EnsLib.REST.GenericService`
  - ・ `EnsLib.REST.SAMLGenericService`
  - ・ `EnsLib.SOAP.GenericService`
  - ・ `EnsLib.SOAP.SAMLGenericService`
2. このパススルー・サービスを呼び出すポートとして、Web ポートと専用ポートのいずれが使用されるかを設定します。ライブ・プロダクションの場合、Web ポートを堅牢な Web サーバ・ソフトウェアと共に使用して、高負荷を処理できる安全で完全に構成可能なシステムを構築できるようにします。(一方、専用ポートは、最小限の構成とセキュリティを提供する軽量リスナです。両方のポートで呼び出しを受け入れるようにサービスを構成することはできますが、そのような構成はお勧めしません。)

パススルー・サービスを Web ポートを通じて呼び出す場合は、以下の操作を実行します。

- ・ [ポート] フィールドを空白のままにします。
- ・ [設定] タブの [接続設定] フィールドで [標準リクエスト有効] チェックボックスにチェックを付けます。
- ・ [追加設定] セクションで、[プール・サイズ] を 0 に設定します。これにより、パススルー・サービスが専用ポートでリッスンしなくなります。この手順を省略して、[ポート] フィールドを空白のままにした場合は、InterSystems IRIS ではエラー・メッセージが表示されます。

パススルー・サービスを専用ポートから呼び出す場合は、以下の操作を実行します。

- ・ ポート番号を指定します。
- ・ [設定] タブの [接続設定] フィールドで [標準リクエスト有効] チェックボックスのチェックを外します。

### 3. [ターゲット] で、対象のパススルー・オペレーションを指定します。

Web ポートを使用するパススルー・サービスのパフォーマンスを最適化するために、パススルー・ビジネス・サービスが複数の呼び出しにまたがって接続を開いたままにするように構成できます。このように構成するには、[\[標準要求パーティションを維持する\]](#) チェック・ボックスにチェックを付けます。

SOAP と REST のパススルー・サービスで SAML 検証を使用する方法は、“[パススルー・サービスでの SAML 検証の使用法](#)” を参照してください。

Web ポートを使用してパススルー・ビジネス・サービスにアクセスするには、Web アプリケーションの定義も必要です。詳細は、“[Web アプリケーションの構成](#)” を参照してください。

## 5.2 パススルー・ビジネス・オペレーションの構成

パススルー・ビジネス・オペレーションを追加するには、[プロダクション構成] ページで新規オペレーションを追加するための + 記号を選択してから、以下の手順を実行します。

1. 使用するプロトコルおよびデータベースへのメッセージの格納を抑止するかどうかに基づいてクラスを選択します。詳細は、“[パススルー・サービスおよびパススルー・オペレーションでの永続メッセージの抑止](#)” を参照してください。以下のいずれかのクラスを選択します。
  - ・ `EnsLib.HTTP.GenericOperation`
  - ・ `EnsLib.HTTP.GenericOperationInProc` : メッセージの格納を抑止します。
  - ・ `EnsLib.REST.GenericOperation`
  - ・ `EnsLib.REST.GenericOperationInProc` : メッセージの格納を抑止します。
  - ・ `EnsLib.SOAP.GenericOperation`
2. パススルー・オペレーションの [HTTP サーバ]、[HTTP ポート]、および [URL] の各設定を構成します。これらの設定は、直接構成することも、外部サービス・レジストリを使用して構成することもできます。外部サービス・レジストリを使用して設定するには、[外部レジストリ ID] プロパティの値を設定します。[外部レジストリ ID] の設定の詳細は、“[ESB ホストの構成での外部サービス・レジストリの使用](#)” を参照してください。[外部レジストリ ID] の設定を適用したら、InterSystems IRIS では、レジストリから現在値が読み取られ、この値を使用して他のプロパティが設定され、値は読み取り専用としてマーキングされます。

URL を明示的に設定することも、汎用サービスに送信された受信 URL から派生させて設定することもできます。サービス・レジストリを使用して設定することも、URL プロパティで直接設定することもできます。受信 URL から派生させるには、以下のように、外部サービス・レジストリの [エンドポイント] フィールドの URL セグメントまたは [URL] プロパティを設定します。

- ・ 空の文字列 : GenericMessage 内の URL を使用します。通常はこの URL は汎用サービスに渡された URL です。一般に、この設定値を使用するのは、当該パススルー・サービスが専用ポートから呼び出される場合に、URL に Web アプリケーション名とサービス・コンポーネント名が含まれていない場合です。
- ・ | (縦棒) : GenericMessage 内の URL 値から Web アプリケーション名と構成名を除去して、その残り部分を URL として使用します。一般に、この設定値を使用するのは、GenericService が標準 Web ポートを通じて呼び出される場合です。ビジネス・サービスの Web アプリケーション名と構成名は、GenericService への呼び出しをルーティングするために必要ですが、外部サーバにとっては不要です。

URL は Web アプリケーション名および構成名と比較されます。このとき大文字小文字が区別されます。Web アプリケーション名のアルファベット文字はすべて小文字にする必要があり、構成名に対応する URL のセグメントは、プロダクションで定義されているとおりの構成名の大文字小文字に一致する必要があります。

- ・ `^` (キャレット) : GenericMessage 内の URL 値から Web アプリケーション名を除去して、その残り部分を URL として使用します。一般に、この設定値を使用するのは、GenericService が標準 Web ポートを使用して呼び出される場合に、パススルー・オペレーション・コンポーネントの名前が外部サーバで想定される URL の先頭部分と同じである場合のみです。URL 内の Web アプリケーション名のアルファベット文字はすべて小文字にする必要があります。

[URL] プロパティで空ではない文字列を指定し、縦棒もキャレットも含めない場合、受信 URL は送信 URL の生成には使用されません。

3. パススルー・オペレーションは通常パススルー・メッセージの内容を変更しませんが、`EnsLib.HTTP.GenericOperation` オペレーションと `EnsLib.REST.GenericOperation` オペレーションで文字セット変換を実行するよう指定できます。例えば、オペレーションに文字変換を実行させ、アクセント文字が Web ブラウザで正しく表示されるようにできます。文字セット変換を実行するように HTTP または REST の汎用オペレーションを設定するには、[追加設定] の [Read Raw モード] チェック・ボックスのチェックを外します。

注釈 GenericMessage クラスの Stream プロパティにデータが含まれる場合、content-length ヘッダが設定されていなくても、ストリームは `EnsLib.HTTP.GenericOperation` および `EnsLib.REST.GenericOperation` によって送信されます。

## 5.3 パススルー・サービスでの SAML 検証の使用法

パススルー・サービスは SAML トークンを検証しますが、SAML トークンによって外部サーバ上のリソースへのアクセスが可能になるかどうかは確認しません。外部サーバは、要求されたアクセスが SAML トークンによって許可されるかどうかを確認する必要があります。[検証] フィールドは、パススルー・ビジネス・サービスによって実行される SAML トークン検証のレベルを制御します。以下のフラグは、実行される検証の種類を指定します。

- ・ `t` - 'access\_token=' というキーを持つ Authorization ヘッダ SAML トークンが含まれている必要があります。
- ・ `a` - トークンにはアサーションが含まれている必要があります。
- ・ `r` - アサーションに NotBefore/NotOnOrAfter 時間条件が含まれている必要があります。
- ・ `v` - Trusted X.509 証明書と NotBefore/NotOnOrAfter 条件 (存在する場合) を使用してアサーション署名を検証します。
- ・ `o` - その他の署名付きノードを検証します (TimeStamp など)。

デフォルトでは検証の値は 1 であり、これは tarvo を指定することと同じです。0 という値を指定した場合は、パススルー・サービスは検証を実行しません。

NotBefore および NotOnOrAfter 時間条件をチェックする際は、デフォルトのクロック・スキュー許容差は 90 秒です。このスキュー許容差を変更するには、`^Ens.Config("SAML", "ClockSkew")` グローバルを設定します。すべてのコンポーネント向けのデフォルト・クロック・スキューを 180 秒に設定するには、次のコマンドを入力します。

```
Set ^Ens.Config("SAML", "ClockSkew")=180
```

特定のコンポーネント向けのスキュー許容差を 180 秒に変更するには、次のコマンドを入力します。

```
Set ^Ens.Config("SAML", "ClockSkew", component-name)=180
```



## 5.4 パススルー・サービスおよびパススルー・オペレーションでの永続メッセージの抑止

パススルー・サービスおよびパススルー・オペレーションを使用して効率性を最大にするために、永続メッセージの使用を抑止することができます。パススルー・サービスは、永続メッセージを作成することなく、直接パススルー・オペレーションに呼び出しを送信します。パススルー・オペレーションは、応答メッセージをパススルー・サービスに同じ方法で送信します。これによって、高レベルのスループットが実現され、メッセージをパージする必要がなくなりますが、以下の制限事項があります。

- ・ パススルー呼び出しの永続的な記録が保持されることはありません。メッセージ・ビューワにメッセージを表示したり、メッセージ・トレースを確認したりできません。これによって、問題のトラブルシューティングが難しくなります。
- ・ 再試行メカニズムが用意されていません。サーバへの通信の最初の試行に失敗した場合、この失敗は、パススルー・サービスを呼び出すアプリケーションに返されます。
- ・ このモードは、パススルー・サービスがパススルー・オペレーションを直接ターゲットとしている場合にのみ使用できます。メッセージが、ビジネス・プロセス・ルータなど他のプロダクション・コンポーネントを経由する場合、プロセス全体で永続メッセージが使用されます。

パススルー・サービスとパススルー・オペレーションのペアの場合に永続メッセージの使用を抑止するには、パススルー・オペレーションの専用クラスの 1 つを選択します。

- ・ `EnsLib.HTTP.GenericOperationInProc`
- ・ `EnsLib.REST.GenericOperationInProc`
- ・ `EnsLib.SOAP.GenericOperationInProc`

さらに、永続メッセージを抑止するために、パススルー・ビジネス・サービス構成で [永続メッセージを InProc で送信] チェック・ボックスのチェックを外す必要があります。

パススルー・サービスで標準の Web ポートを使用している場合、サービスが複数の呼び出しにまたがって TCP 接続を開いたままにするように構成することによって、効率性をさらに向上させることができます。そのためには、パススルー・サービス構成で、[標準要求パーティションを維持する] チェック・ボックスにチェックを付けます。

## 5.5 その他のビジネス・サービス、ビジネス・プロセス、およびビジネス・オペレーションの使用

パススルー・サービスとパススルー・オペレーションのみが設定されたプロダクションで最も単純な ESB システムを構成できますが、要件によっては、より複雑なプロダクション・コンポーネントでしか満たすことができません。例えば、ESB で以下のいずれかの処理を行う必要がある場合、他の種類のビジネス・サービスとビジネス・オペレーションが必要です。

- ・ 1 つの受信サービスからの呼び出しを、呼び出しの内容に応じて複数の外部サービスにルーティングする処理。
- ・ サービスに使用されるパラメータまたはプロトコルの変更。
- ・ 複数の外部サービスの機能を組み合わせて実装される 1 つのサービスの公開。
- ・ ESB へのサービスの直接の提供。
- ・ SAML トークンに基づいたセキュリティ以外のセキュリティの提供。



他のビジネス・サービス、ビジネス・プロセス、およびビジネス・オペレーションを使用して REST および SOAP サービス要求を処理する操作の詳細は、“[InterSystems IRIS での Web サービスおよび Web クライアントの作成](#)”を参照してください。

## 5.6 パススルー・サービスおよびパススルー・オペレーションのパフォーマンス統計の追跡

使用している ESB システムが、ユーザのニーズを満たし続けることができるように、ESB システムのパフォーマンスを監視して追跡することが重要です。この監視を定期的に行うことによって、作業負荷の増加を管理でき、負荷が増加した場合はリソースの追加で対処します。パススルー・サービスおよびパススルー・オペレーションで永続メッセージを使用している場合は、InterSystems IRIS の監視機能を使用して、パフォーマンスの追跡とレポートを行うことができますが、アクティビティ量の監視には、以下の場合に有用なパフォーマンス統計の概要を作成するメカニズムが用意されています。

- ・ システム全体のパフォーマンスの追跡
- ・ システムがサービス品質保証 (SLA) を遵守していることの確認
- ・ 潜在的な問題の特定

永続メッセージの保存を抑制している場合 (“[パススルー・サービスおよびパススルー・オペレーションでの永続メッセージの抑止](#)”を参照)、このような統計概要が、パフォーマンス追跡の主要ツールとなります。

この統計概要は、パススルー・サービスおよびパススルー・オペレーションごとの以下の情報で構成されます。

- ・ 決められた時間間隔 (10 秒) の間に完了したメッセージの合計数
- ・ これらのメッセージの処理の完了に要した合計時間

一般的にパフォーマンスの監視に当てはまることですが、パフォーマンス統計の定期的な監視および記録が重要です。これによって、傾向を検出したり、現在のパフォーマンスを基準値と比較したりできます。これは、パフォーマンス上の問題の原因が、負荷の増加、ネットワーク上の問題、サービスを提供しているサーバの問題、あるいは ESB のパフォーマンス上の問題のいずれであるかの判別に役立つ場合があります。

パススルー・サービスおよびパススルー・オペレーションの統計概要のメカニズムは、以下のコンポーネントで構成されます。

- ・ [構成] および [グローバル] の設定。パススルー・サービスおよびパススルー・オペレーションでの統計の生成を有効にします。
- ・ デーモン。パススルー・サービスおよびパススルー・オペレーションから統計を取得してデーモンに送信し、保存します。
- ・ ユーザ・インタフェース。統計の監視および調査に使用します。

詳細は、“[プロダクションの監視](#)”の“[アクティビティ量の監視](#)”を参照してください。



# A

## InterSystems IRIS システムの構成とネームスペースの作成

ここでは、InterSystems IRIS® を ESB として使用したり、パススルー REST および SOAP サービスおよびオペレーションと共に使用したりするように InterSystems IRIS を構成する方法を説明します。この付録は以下の節で構成されています。

ここでは、Web ポートを通じて HTTP および SOAP サービスを使用できるようにシステムを構成する方法を簡単に説明します。ここで紹介する情報は、これらのサービス用の開発システムやテスト・システムをセットアップするのに役立ちます。これらのトピックに関する包括的な情報は、ドキュメントに記載されています。詳細は、“Configuring System-Wide Settings” を参照してください。

開発システムまたはテスト・システムを HTTP または SOAP サービス向けにセットアップするには、以下の手順を実行します。

1. InterSystems IRIS をロックダウンされたインストール環境にインストールした場合は、`%Services_Bindings` サービスへのアクセスは無効になっています。管理ポータルを開いてサービスへのアクセスを有効にします。
  - a. InterSystems IRIS ランチャーから管理ポータルを開始します。管理ポータルにアクセスするには、`_system` ではなく Windows のログイン・ユーザ名を使用する必要があります。インストール時に指定したパスワードを入力します。
  - b. **[システム管理]**→**[セキュリティ]**→**[サービス]** の順に選択して、**[サービス]** ポータル・ページを表示します。
  - c. デフォルトでは、**[%Services\_Bindings]** サービスは無効になっています。このサービス名を選択し、**[サービス有効]** チェック・ボックスにチェックを付けて、設定を保存します。
2. 既存のネームスペースを使用しない場合は、新しいネームスペースを作成します。
  - a. **[システム管理]**→**[構成]**→**[システム構成]**→**[ネームスペース]** の順に選択して、**[ネームスペース]** ポータル・ページを表示します。
  - b. **[新規ネームスペース作成]** ボタンを選択して、ネームスペースの名前を指定します (例 : `SERVICESNS`)。
  - c. グローバル用データベースについて **[新規データベース作成]** ボタンを選択します。
  - d. データベース・ウィザードで、グローバル用データベースの名前を入力します (例 : `SERVICES_GDB`)。ウィザードはこの名前を使用して、このデータベース用のディレクトリを作成します。
  - e. **[次]** ボタンを 2 回選択して、**[データベースリソース]** フォームを表示します。**[新規リソース作成]** ラジオ・ボタンを選択します。ウィザードで **[新規リソース作成]** フォームが表示されます。表示されている名前 (`%DB_SERVICES_GDB` など) をそのまま使用して、**[パブリック許可]** の **[読み込み]** と **[書き込み]** というチェック・ボックスのチェックが外れていることを確認します。**[データベースリソース]** フォームの **[保存]** ボタンを選択して、**[データベースウィザード]** フォームの **[完了]** ボタンを選択します。

- f. ルーチン用データベースについて手順 c から e を繰り返します。
- g. **[保存]** ボタンを選択して、ネームスペースの作成を完了します。
- h. **[閉じる]** を選択してログを閉じます。

これで、システム構成が完了しました。

# B

## パススルー・ビジネス・サービスへの Web アプリケーションの構成

ここでは、パススルー・ビジネス・サービスに Web アプリケーションを構成する方法を説明します。パブリック・サービス・レジストリ REST API 用に Web アプリケーションを構成するには、“[パブリック・サービス・レジストリ API への Web アプリケーションの構成](#)” を参照してください。

1. 次のように、空のロールを作成して、このロールを不明ユーザに割り当てます。
  - a. [システム管理]→[セキュリティ]→[ロール] の順に選択して、[ロール] ポータル・ページを表示します。
  - b. [新規ロール作成] ボタンを選択して、ロールに名前を付け (例 : Services\_Role)、[保存] ボタンを選択します。
  - c. [メンバ] タブを選択して、[不明ユーザ] を選択し、右矢印を選択して、[割り当てる] ボタンを選択します。
2. Web ポートへの呼び出しを処理する Web アプリケーションを定義します。この Web アプリケーションの名前によって、サービス呼び出す URL のルートが指定されます。単一の Web アプリケーションは複数のビジネス・サービスをサポートできますが、これらのビジネス・サービスはすべて、同じクラスを持っているか、その Web アプリケーションのディスパッチ・クラスのサブクラスを持っている必要があります。
  - a. [システム管理]→[セキュリティ]→[アプリケーション]→[Web アプリケーション] の順に選択して、[Web アプリケーション] ポータル・ページを表示します。[新規 Web アプリケーション作成] ボタンを選択します。
  - b. この Web アプリケーションに名前を付けます (例 : /restpassthrough や /soappassthrough)。名前の先頭には / (スラッシュ) を付加する必要があります。名前の中のアルファベット文字はすべて小文字にする必要があります。
  - c. [ネームスペース] で、当該プロダクションが実行されているネームスペースを指定します (例 : SERVICESNS)。[ネームスペースのデフォルト・アプリケーション] チェックボックスのチェックは外したままにします。
  - d. [アプリケーション]、[Web ファイル]、および [着信 Web サービス] の各チェック・ボックスにはチェックを付けてもかまいません。
  - e. [必須リソース] フィールドと [ID でグループ分け] フィールドは空白のままにします。
  - f. [許可された認証方法] 行の [認証なし] チェック・ボックスにチェックを付けます。
  - g. [ディスパッチ・クラス] でコンポーネント・クラスを指定します (例 : EnsLib.REST.GenericService や EnsLib.SOAP.GenericService)。
  - h. [保存] を選択します。
  - i. [マッチングロール] タブを選択します。
  - j. [マッチングロールを選択:] フィールドで、前の手順で作成したロールを選択します。

- k. **[選択されたマッチングロールに対して追加するターゲットロールを選択]** フィールドで、当該ネームスペースのグローバルとルーチンに関連付けられた 1 つまたは複数のロールを選択します。これらのグローバルとルーチンは、同じデータベース内にあっても別々のデータベース内にあってもかまいません。お使いのサービスが別の InterSystems IRIS® データベースにアクセスする場合は、そのデータベースのロールも選択する必要があります。複数のロールを選択するには、Ctrl キーを押しながら選択します。

注釈 グローバル用データベースには、二次データベースおよび対応するロールも存在する場合があります (例: %DB\_GDBSECONDARY)。この二次データベースにはパスワードが保管されます。パススルー・サービスとパススルー・オペレーションのためにこのデータベースにアクセスする必要はありませんが、パスワード・アクセスを使用するカスタム Web サービスを作成する場合は、二次データベースのロールをターゲット・データベースに追加する必要もあります。

- l. 選択したロールがハイライト表示された状態で、右矢印ボタンを選択してこれらのロールを **[選択済み]** テキスト・ボックスに移動します。
- m. **[割り当てる]** ボタンを選択します。

注釈 Web アプリケーションで非認証サービスと認証済みサービスの両方を許可する場合は、パスワードの入力が求められますが、パスワードを入力しないと、Web アプリケーションによって非認証アクセスが許可されます。

# C

## パススルー・サービスとパススルー・オペレーションのワークスルー

このワークスルーでは、パススルー・サービスとパススルー・オペレーションの使用法を紹介します。

**注釈** このワークスルーでは、openweathermap.org サーバと www.webservice.net サーバによって提供されるいくつかの無料サービスを使用します。これらのサーバのいずれもインターシステムズとは関係なく、これらのサーバを常に使用できるとは限りません。これらのサーバを使用できない場合は、各自の環境で使用可能な他の任意の REST サーバまたは SOAP サーバを代わりに使用できます。openweathermap.org サーバは無料の天気データを提供します。このサーバをアプリケーション内で使用する予定の場合は、アプリケーション ID を取得する必要があります。このサービスは無料ですが、天気サーバに関する製品サポートが有料で提供されています。このサーバの詳細は、<http://openweathermap.com/> を参照してください。www.webservice.net サーバは、Cloud Computing Technologies 社から提供されています。

### C.1 REST パススルー・サービスおよびオペレーションの使用法

この節では、REST パススルー・サービスと REST パススルー・オペレーションを追加します。このサービスは Web ポート上でリッスンして、HTTP REST メッセージをパススルー・オペレーションに送信します。パススルー・オペレーションは、HTTP REST メッセージを [api.openweathermap.org](http://api.openweathermap.org) という外部サーバに送信して、この外部サーバは現在の天気情報を返します。

ネームスペースとプロダクションを選択または作成します。Web ポートを使用するには、そのネームスペース内で Web アプリケーションを定義して、ロールを定義する必要があります。ネームスペース、ロール、および Web アプリケーションの作成手順は、“[InterSystems IRIS システムの構成](#)”を参照してください。このワークスルーでは、Web アプリケーションに `/restpassthrough` (Web アプリケーション名は小文字の必要があります) という名前を付けて、Web アプリケーションの **[ディスパッチ・クラス]** を `EnsLib.REST.GenericService` に設定します。

1. **[プロダクション構成]** ページで、新しいオペレーションを追加するためのプラス記号を選択します。
  - a. オペレーション・クラス `EnsLib.REST.GenericOperation` を指定します。
  - b. オペレーションに名前を付けます (例 : `WeatherRESTPassthroughOp`)。
  - c. まだ有効にしないでください。
  - d. **[OK]** を選択します。
2. 作成したパススルー・オペレーションを選択して、**[設定]** タブを選択します。

- a. 外部サービス・レジストリを使用している場合は、[HTTP サーバ]、[HTTP ポート]、および [URL] の設定を指定するレジストリ・エントリを識別するための [外部レジストリ ID] を設定し、手順 3 に進みます。それ以外の場合は、次のように、これらの設定を直接指定します。
  - b. [HTTPサーバ] に「api.openweathermap.org」と入力します。
  - c. [URL] フィールドに「/」と入力します。
  - d. 他のフィールドはデフォルト値のままにします。
  - e. [有効] チェック・ボックスにチェックを付けます。
  - f. [適用] を選択します。
3. 新しいサービスを追加するためのプラス記号を選択します。
    - a. サービス・クラス **EnsLib.REST.GenericService** を指定します。
    - b. サービスに名前を付けます (例 : WeatherRESTPassthroughServ)。
    - c. まだ有効にしないでください。
    - d. [OK] を選択します。
  4. 作成したパススルー・サービスを選択して、[設定] タブを選択します。
    - a. [ポート] 番号が空白であることを確認します。
    - b. [ターゲット構成名] フィールドで、前の手順で追加したパススルー・オペレーションを選択します。
    - c. [接続設定] で、[標準リクエスト有効] チェック・ボックスにチェックを付けます。
    - d. [追加設定] で、[プール・サイズ] を 0 に設定します。これにより、パススルー・サービスが専用ポートでリスンしなくなります。プール・サイズを 0 に設定しない場合は、ポート番号を指定する必要があります。
    - e. [有効] チェック・ボックスにチェックを付けます。
    - f. [適用] を選択します。
  5. プロダクションを起動します。
  6. Web ポートを使用するパススルー・サービスの URL を入力します。この URL は以下の形式をとります。
    - ・ https://<baseURL>/ – <baseURL> は、IRIS インスタンスのベース URL です。
    - ・ 指定した web-application-name/。例 : restpassthrough/。"パススルー・ビジネス・サービスへの Web アプリケーションの構成" を参照してください。
    - ・ passthrough-service-name/ – 例 : WeatherRESTPassthroughServ/
    - ・ 外部サービスで要求される URL (data/2.5/weather?q=London&appid=asd9g87w42MyAPIKey など)。API キーを取得するには、OpenWeather アカウントに登録する必要があります。

例えば、OpenWeather から取得した API キーを含めて、次の URL を Web ブラウザに入力します。

`http://<baseURL>/restpassthrough/WeatherRESTPassthroughServ/data/2.5/weather?q=London&appid=asd9g87w42MyAPIKey`

Web アプリケーション名のアルファベット文字はすべて小文字にする必要があり、パススルー・サービス名は、構成項目名の太文字小文字に一致する必要があります。

すべて正常に機能している場合は、呼び出しによって現在のロンドンの天気が含まれた JSON メッセージが返されます。以下に、返される JSON メッセージの例を示します。



```
{
  "coord":{"lon":-0.13,"lat":51.51},
  "sys":{"message":0.0441,"country":"GB","sunrise":1399609017,"sunset":1399664214},
  "weather":[{"id":802,"main":"Clouds","description":"scattered clouds","icon":"03n"}],
  "base":"cmc stations",
  "main":{"temp":285.32,"pressure":1008,"temp_min":284.15,"temp_max":286.48,"humidity":91},
  "wind":{"speed":5.14,"gust":6.17,"deg":284},
  "rain":{"3h":0},
  "clouds":{"all":32},
  "dt":1399600508,
  "id":2643743,
  "name":"London",
  "cod":200
}
```

## C.1.1 パススルーの背景情報

パススルー・サービスへの呼び出しと外部サーバへの呼び出しの間で、URL がどのように変換されるのかに注目してください。パススルー・サービスに送信される URL の形式は以下のとおりです。〈baseURL〉はインスタンスのベース URL です。

`http://〈baseURL〉/Restpassthrough/WeatherRESTPassthroughServ/data/2.5/weather?q=London`

外部サーバに送信される URL は次のとおりです。

`http://api.openweathermap.org/data/2.5/weather?q=London`

data で始まる URL の末尾部分はどちらも同じです。[HTTPサーバ] フィールドでは、発信呼び出しでのサーバを設定します。[URL] フィールドで | という値を指定すると、オペレーションによって受信 URL から Web アプリケーション名と構成名が除去されて、その残り部分が送信 URL に組み込まれます。

**注釈** `EnsLib.REST.GenericService` クラスは、`EnsLib.HTTP.GenericService` クラスにはない追加機能を提供しません。`EnsLib.HTTP.GenericService` クラスを使用して REST 呼び出しをパススルーできますが、適切なサブクラスを使用する必要があります。

## C.1.2 パススルーのトラブルシューティング

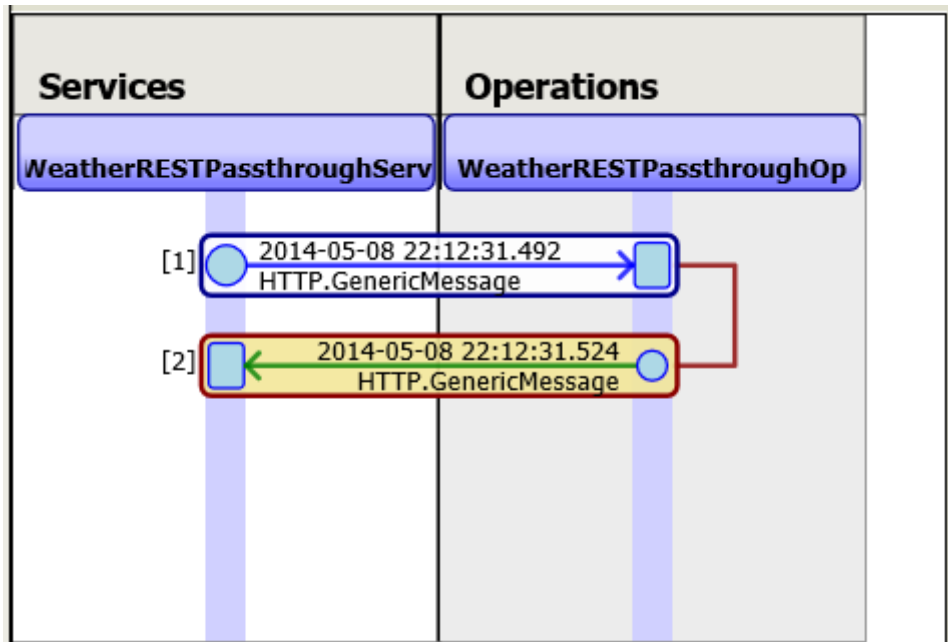
想定どおりの結果が得られない場合は、いくつかのトラブルシューティング手順を実行できます。

1. InterSystems IRIS を使用せずに、openweathermap の URL を Web ブラウザに直接入力します。次の URL を Web ブラウザに入力します。

`http://api.openweathermap.org/data/2.5/weather?q=London,uk`

これにより、天気サーバが正常に機能していることを確認します。

2. プロダクションのプロダクション・メッセージ・トレーシングを調べます。正常なメッセージ・トレースは次のように表示されます。



このトレースは、エラーの場所を特定するのに役立つことがあります。

3. TCPトレース・ユーティリティを実行します。リッスンするローカル・ポート、api.openweathermap.org というサーバ、およびポート 80 を指定します。パススルー・オペレーションの設定で、次のプロパティを設定します。

- ・ **[HTTPサーバ]** : localhost
- ・ **[HTTPポート]** : TCP トレースがリッスンしているポート

**[適用]** をクリックし、次のようにパススルー・サービスの URL を Web ブラウザに入力します。〈baseURL〉はインスタンスのベース URL です。

`http://<baseURL>/Restpassthrough/WeatherRESTPassthroughServ/data/2.5/weather?q=Boston,ma`

TCP トレースで、パススルー・オペレーションによって送出される HTTP 要求とサーバの応答が表示されることを確認します。

## C.2 SOAP パススルー・サービスおよびオペレーションの使用法

この節では、SOAP パススルー・サービスと SOAP パススルー・オペレーションを追加します。このサービスは Web ポート上でリッスンして、SOAP メッセージをパススルー・オペレーションに送信します。パススルー・オペレーションは、SOAP メッセージを外部サーバ `www.webservice.net` に送信します。この外部サーバが提供する各種サービスのうちの 1 つである通貨換算サービスによって、2 つの指定された通貨の為替レートが返されます。SOAP パススルー・サービスを呼び出すには、SOAPUI などの SOAP ツールキットを使用する必要があります。SOAPUI ツールキットの詳細は、<http://www.soapui.org/> を参照してください。

ネームスペースとプロダクションを選択または作成します。Web ポートを使用するには、そのネームスペース内で Web アプリケーションを定義して、ロールを定義する必要があります。ネームスペース、ロール、および Web アプリケーションの作成手順は、“[InterSystems IRIS システムの構成](#)”を参照してください。このウォークスルーでは、Web アプリケーションに `/SOAPpassthrough` という名前を付けて、Web アプリケーションの **[ディスパッチ・クラス]** を `EnsLib.SOAP.GenericService` に設定します。

## C.2.1 SOAP ツールキットからの外部サーバの呼び出し

パススルー・サービスとパススルー・オペレーションを作成する前に、SOAP ツールキットを使用してプロジェクトを作成して、外部サーバを直接呼び出してください。SOAP ツールキットを使用して、以下の手順を実行します。

- ・ 新しいプロジェクトを作成します。
- ・ WSDL ファイルの場所として以下を指定します。

```
http://www.webservicex.net/CurrencyConvertor.asmx?WSDL
```

- ・ 作成したプロジェクトに対して生成された要求を選択します。ツールキットによって SOAP 1.1 と 1.2 の要求が生成される場合は、SOAP バージョン 1.2 の要求を使用します。サーバに送信される SOAP メッセージがツールキットに表示されます。例えば、SOAPUI には以下が表示されます。

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:web="http://www.webserviceX.NET/">
  <soap:Header/>
  <soap:Body>
    <web:ConversionRate>
      <web:FromCurrency>?</web:FromCurrency>
      <web:ToCurrency>?</web:ToCurrency>
    </web:ConversionRate>
  </soap:Body>
</soap:Envelope>
```

- ・ 2 つの疑問符を通貨の略称に置換します。例えば、ユーロを米ドルに換算するには、1 つ目の疑問符を EUR に置換して、2 つ目の疑問符を USD に置換します。
- ・ 要求を実行します。SOAPUI を使用し、緑色の矢印を選択して要求を実行します。SOAP ツールキットに、次の例のような応答 SOAP メッセージが表示されます。

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
      <ConversionRateResult>1.3603</ConversionRateResult>
    </ConversionRateResponse>
  </soap:Body>
</soap:Envelope>
```

## C.2.2 SOAP パススルー・サービスおよびオペレーションの追加

この節では、SOAP パススルー・オペレーションと、Web ポートを使用する SOAP パススルー・サービスを追加します。

1. [プロダクション構成] ページに移動して、新しいオペレーションを追加するためのプラス記号を選択します。
  - a. オペレーション・クラス **EnsLib.SOAP.GenericOperation** を指定します。
  - b. オペレーションに名前を付けます (例 : ConvertCurrencySOAPCSPOp)。
  - c. まだ有効にしないでください。
  - d. [OK] を選択します。
2. 作成したパススルー・オペレーションを選択して、[設定] タブを選択します。
  - a. 外部サービス・レジストリを使用している場合は、[HTTP サーバ]、[HTTP ポート]、および [URL] の設定を指定するレジストリ・エントリを識別するための [外部レジストリ ID] を設定し、手順 3 に進みます。それ以外の場合は、次のように、これらの設定を直接指定します。
  - b. [HTTPサーバ] に「www.webservicex.net」と入力します。
  - c. [URL] フィールドに「|」(縦棒)を入力します。これにより、Web アプリケーションでは必要であるが外部サービスでは必要ない余分な部分が受信 URL から除去されます。

- d. [有効] チェック・ボックスにチェックを付けます。
  - e. [適用] を選択します。
3. 新しいサービスを追加するためのプラス記号を選択します。
    - a. オペレーション・クラス **EnsLib.SOAP.GenericService** を指定します。
    - b. オペレーションに名前を付けます (例 : ConvertCurrencySOAPCSPServ)。
    - c. まだ有効にしないでください。
    - d. [OK] を選択します。
  4. 作成したパススルー・サービスを選択して、[設定] タブを選択します。
    - a. [有効] チェック・ボックスにチェックを付けます。
    - b. [ポート] フィールドが空白であることを確認します。
    - c. [ターゲット構成名] で、前の手順で作成したパススルー・オペレーションを指定します。
    - d. [接続設定] で、[標準リクエスト有効] チェック・ボックスにチェックを付けます。
    - e. [追加設定] で、[プール・サイズ] を 0 に設定します。これにより、パススルー・サービスが専用ポートでリスンしなくなります。
    - f. [適用] を選択します。
  5. プロダクションを起動します。
  6. SOAP ツールキットで、Web ポートを使用するパススルー・サービスの URL を入力します。この URL は以下の形式をとります。
    - ・ `https://<baseURL>/` – <baseURL> は、IRIS インスタンスのベース URL です。
    - ・ 指定した web-application-name/。例 : `soappassthrough/`。"パススルー・ビジネス・サービスへの Web アプリケーションの構成" を参照してください。
    - ・ `passthrough-service-name/` – 例 : `ConvertCurrencySOAPCSPServ/`
    - ・ 外部サービスで要求される URL (`CurrencyConvertor.asmx` など)

例えば、サービス名を `soappassthrough`、パススルー・ビジネス・サービス名を `ConvertCurrencySOAPCSPServ` と指定する場合、Web ブラウザに以下の URL を入力します。

`http://<baseURL>/soappassthrough/ConvertCurrencySOAPCSPServ/CurrencyConvertor.asmx`

Web アプリケーション名のアルファベット文字はすべて小文字にする必要があります、パススルー・サービス名は、構成項目名の大文字小文字に一致する必要があります。

要求を実行します。SOAP ツールキットは、前に返したのと同じ XML メッセージを返します。

# サービス・レジストリ・パブリック API リファレンス

ここでは、サービス・レジストリ・パブリック API に関する参照情報を提供します。

## レジストリ・エントリを伴った JSON 戻りメッセージのサンプル

サービス・レジストリ・パブリック API 呼び出しのほとんどが、1 つ以上のレジストリ・エントリを返します。この節では、これらのエントリのフォーマットについて説明します。

### 複数のエントリを伴った JSON メッセージ

```
[
  {
    "Name": "MathServiceSOAP",
    "Domain": "UnitTest",
    "Version": "1.0",
    "Stage": "Live",
    "Protocol": "SOAP",
    "Description": "Add 2 Numbers",
    "Endpoint": "https://jgm6457/enslatest/csp/support/Demo.SOAP.MathService.cls",
    "ResponseStyle": "Sync",
    "LastModified": "2022-03-16 19:07:47.469",
    "Topics":
    [
      "Test",
      "Maths"
    ],
    "Contacts":
    [
      {
        "Identity": "QD Developer Moon",
        "Type": "Operator",
        "Details": "Details of contact",
        "BusinessPartner": "QD",
        "Notes": "This SOAP service is designed to have minimum moving parts"
      }
    ],
    "Schema":
    {
      "Type": "Notes",
      "Ref": "https://jgm6457/enslatest/csp/support/Demo.SOAP.MathService.cls?wsdl=1",
      "Content": "<definitions targetNamespace='http://tempuri.org'> ...</definitions>",
      "Notes": "Some WSDL"
    },
    "Public": true,
    "Attributes":
    [
      {
        "Name": "One",
        "Value": "1"
      }
    ],
    "Files":
    [
      {
        "Filename": "SOAPMathService.WSDL",
        "FileExtention": ".WSDL",
        "MIMETYPE": "text/text",
        "CharEncoding": "UTF-8",
        "FileSize": "1.44 KB",
        "Contents": null
      }
    ],
    "Actions":
    [
      {
        "Name": "Sum",
        "Ref": "Sum",
        "Verb": "POST",
        "Description": "Add up 2 numbers",
        "ReadOnly": false,
        "Idempotent": true
      }
    ]
  },
  {
    "Name": "PublicREST",
    "Domain": "UnitTest",
    "Version": "0.9",
    "Stage": "Live",
    "Protocol": "REST",
    "Description": "REST Call for the Public Registry",
    "Endpoint": "http://mymachine.mynetwork.com.com:57774/csp/registry/docserver/public",
```

```
"ResponseStyle": "Sync",
"LastModified": "2022-03-05 16:15:33.38",
"Topics":
[
  "Public",
  "Search",
],
"Public": true,
"Attributes":
[
  {
    "Name": "Security",
    "Value": "Username and Password"
  }
],
"Files":
[
  {
    "Filename": "TestPlan",
    "MimeType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
    "CharEncoding": "UTF-8",
    "FileSize": "16.95 KB",
    "Contents": null
  }
],
"Actions":
[
  {
    "Name": "public",
    "Ref": "public",
    "Verb": "GET",
    "Description": "Returns REST endpoint for public registry",
    "RequestSchema":
    {
      "Type": "Notes",
      "Notes": "This is the REST endpoint to query the public Registry"
    },
    "ResponseSchema":
    {
      "Type": "REST Information"
    },
    "ReadOnly": true,
    "Idempotent": true
  }
]
}
```

## 一般的な URL パラメータ

---

一般的な URL パラメータ `?format` は、ほとんどのサービス・レジストリ・パブリック API 呼び出しで使用可能です。このパラメータは、JSON 応答内の改行とインデントを制御します。このパラメータは、すべてのサービス・レジストリ・パブリック API 呼び出しで使用可能です。デフォルトでこれらの呼び出しは、各要素が 1 つの空白で区切られていて、改行のない JSON を返します。

### URL パラメータ

`?format=format-code`

返される JSON のフォーマットとインデントを行うオプションのパラメータ。

`format-code` には、以下のコードが 1 つ以上含まれます。

- ・ `n` – JSON 応答の各要素を新規行文字で終了します。
- ・ `w` – JSON 応答の各要素を新規行/改行で終了します。
- ・ `i` – JSON 応答の各レベルを 4 つの空白文字または 1 つのタブでインデントします。1 ～ 9 の数値も指定した場合、その数の空白またはタブでインデントされます。
- ・ 数値 1 ～ 9 – JSON 応答の各レベルを、指定した数の空白でインデントします。
- ・ `t` – インデントに空白文字の代わりにタブを使用します。
- ・ `u` – JSON 出力を内部フォーマットから UTF-8 に変換します。



## [取得]

Get About で、実行可能なパブリック API REST 呼び出しが一覧表示されます。

### 要求メソッドおよび URL

GET /about

### URL パラメータ

“一般的な URL パラメータ”を参照してください。

### 応答

```
[
  {
    "EndPoints": [
      {
        "GetAbout": "/about"
      },
      {
        "GetServices": "/services"
      },
      {
        "GetServicesByID": "/services/id/:DATA/:DATA/:DATA"
      },
      {
        "GetFileByID": "/services/id/:DATA/:DATA/:DATA/file/:DATA"
      },
      {
        "GetServicesByWord": "/services/includesword/:DATA"
      },
      {
        "GetServicesModifiedSince": "/services/modifiedsince/:DATA"
      },
      {
        "GetServicesByProtocols": "/services/protocols/:DATA"
      },
      {
        "GetServicesByStages": "/services/stages/:DATA"
      },
      {
        "GetServicesByVersion": "/services/version/:DATA"
      }
    ]
  },
  {
    "QueryParameters": [
      {
        "format":
          "1-9 = indent with this number of spaces (4 is the default with the 'i' format specifier);
          i - indent with 4 spaces unless 't' or 1-9;n - newline (lf);
          t - indent with tab character;
          u - output pre-converted to UTF-8 instead of in native internal format;
          w - Windows-style cr/lf newline"
      }
    ]
  }
]
```

### 呼び出しのサンプル

https://esb.example.com/registry/about

## [ID によるファイルの取得]

---

Get File By ID で、指定されたファイルが返されます。

### 要求メソッドおよび URL

GET `/services/id/name/domain/version/file/filename`

### URL パラメータ

なし。

### 応答

ファイルの内容。

### 呼び出しのサンプル

`https://esb.example.com/registry/services/id/weather/util/1.0/file/brief.txt`

## [サービスの取得]

---

Get Services で、ユーザがアクセス可能なすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

`https://esb.example.com/registry/services`

## [ID によるサービスの取得]

---

Get Services By ID で、ユーザからアクセス可能で、指定された名前、ドメイン、およびバージョンに一致するすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/id/name/domain/version

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

https://esb.example.com/registry/services/id/weather/util/1.0

## [プロトコルによるサービスの取得]

---

Get Services By Protocol で、ユーザからアクセス可能で、リスト内のいずれかのプロトコルに一致するすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/protocols/protocol-list

protocol-list は、カンマで区切られた、プロトコルのリストです。

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

https://esb.example.com/registry/services/protocols/REST,SOAP

### メモ

プロトコルとしては、以下のいずれかを指定できます。[File]、[FTP]、HL7、[HTTP]、[REST]、[SOAP]、[SQL]、[X12]、および任意のカスタム値。

## [段階によるサービスの取得]

---

Get Services By Stages で、ユーザからアクセス可能で、リスト内のいずれかの段階に一致するすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/stages/stage-list

stages-list は、カンマで区切られた、段階のリストです。

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

`https://esb.example.com/registry/services/stages/Development,Live`

### メモ

段階としては、以下のいずれかを指定できます。[概念]、[開発]、[テスト]、[ステージング]、[稼働中]、[廃止]、および [非アクティブ]。

## [バージョンによるサービスの取得]

---

Get Services By Version で、ユーザからアクセス可能で、指定されたバージョンに一致するすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/version/version

version は、バージョン番号です。

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

https://esb.example.com/registry/services/version/1.0

## [単語によるサービスの取得]

---

Get Services By Word で、ユーザからアクセス可能で、指定された検索テキストに一致するすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/includesword/search-text

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

https://esb.example.com/registry/services/includesword/patient

### メモ

指定されたテキストが、以下のフィールドで検索されます。

- ・ [名前]
- ・ [ドメイン]
- ・ [説明]
- ・ [エンドポイント]
- ・ [トピック]



## [変更以降のサービスの取得]

---

Get Services Modified Since で、ユーザからアクセス可能で、指定された日時以降にレジストリ・エントリが作成されたか更新されているすべてのサービスが表示されます。

### 要求メソッドおよび URL

GET /services/modifiedsince/date-time

### URL パラメータ

“一般的な URL パラメータ” を参照してください。

### 応答

“レジストリ・エントリを伴った JSON 戻りメッセージのサンプル” を参照してください。

### 呼び出しのサンプル

https://esb.example.com/registry/services/modifiedsince/2015-02-1%2011:30

