

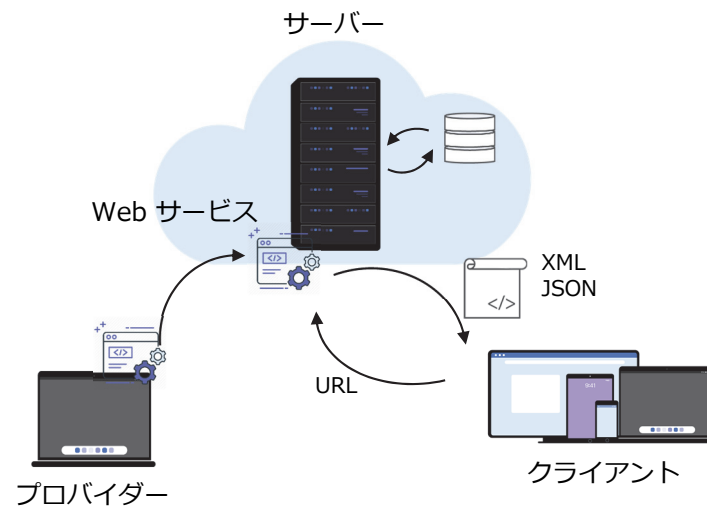
GeneXus の Web サービス

概要

GeneXus™

次に、Web サービスとは何か、GeneXus アプリケーションでどのように使用するのかについて説明します。

定義



Web サービスとは、ほかのプログラムに便利な機能を提供するプログラムのことです。Web サーバー上に配置され、通常はインターネットなどのネットワークを介して呼び出すことができます。

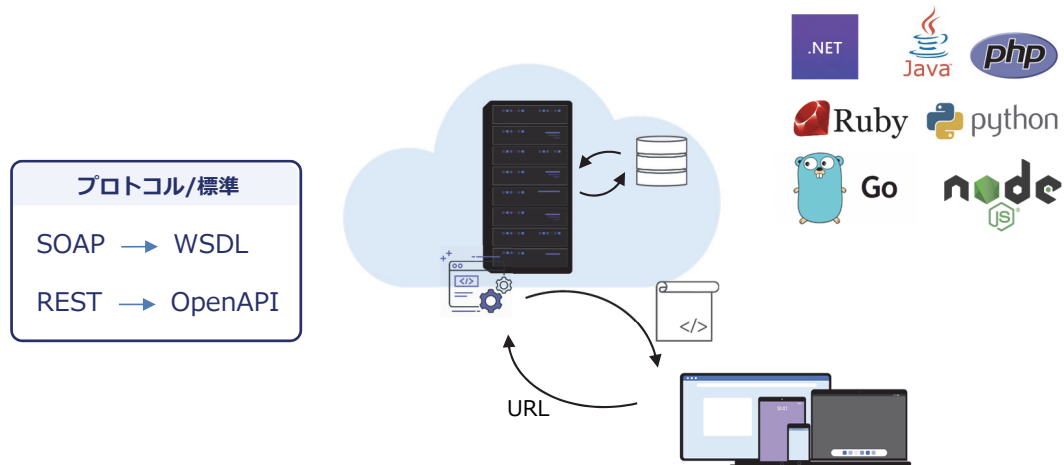
ほかのシステムからアクセスできるように、アプリケーションのバックエンドサービスを Web サーバーで公開すると、それが Web サービスとなります。こうしたサービスへのアクセスを容易にするために、やり取りの方法や受け取る情報の形式を定義する標準が使用されます。

Web サービスは、サービスプロバイダーがサーバー上で「公開」し、クライアントアプリケーションが「利用」します。

クライアントアプリケーションは、サービスにアクセスする際に場所 (URL) を指定して呼び出し、必要なパラメーターを渡します。

通常は、XML 形式または JSON 形式の構造で情報を受け取ります。

Web サービスのプロトコル



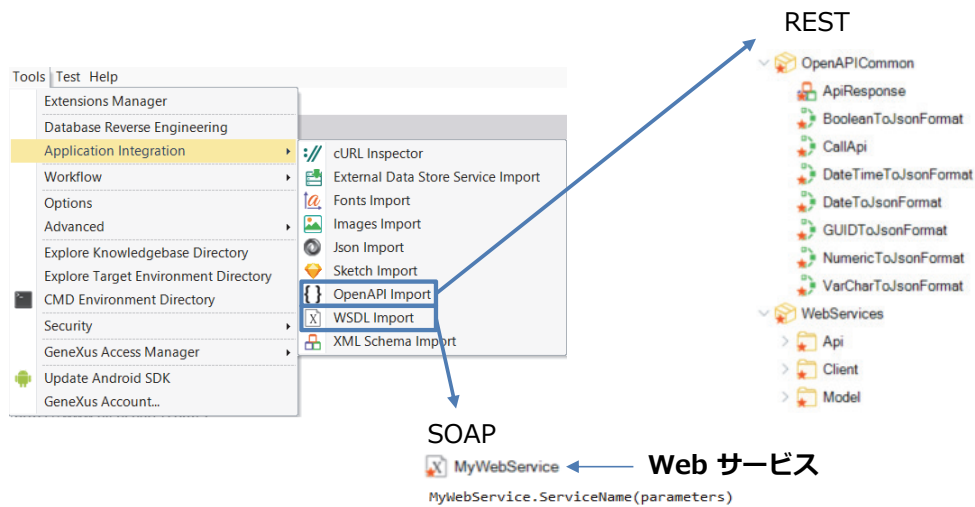
Web サービスはさまざまな標準に基づいて開発されます。業界で最もよく使用されているのは SOAP と REST です。各標準では、Web サービスで提供される機能に関する情報の公開方法が定義されています。

SOAP Web サービスの場合は WSDL (Webサービス記述言語) で記述される定義を使用しますが、REST サービスの場合は OpenAPI 標準を使用します。

公開されている Web サービスを利用するには、そのサービスの場所の情報が必要です。また、利用可能な機能にアクセスするためには、その Web サービスの定義をインポートする必要があります。

GeneXus では、SOAP または REST プロトコルを使用して、任意のプログラミングツールまたはプラットフォームで作成された Web サービスを利用できます。

GeneXus で Web サービスを利用する方法

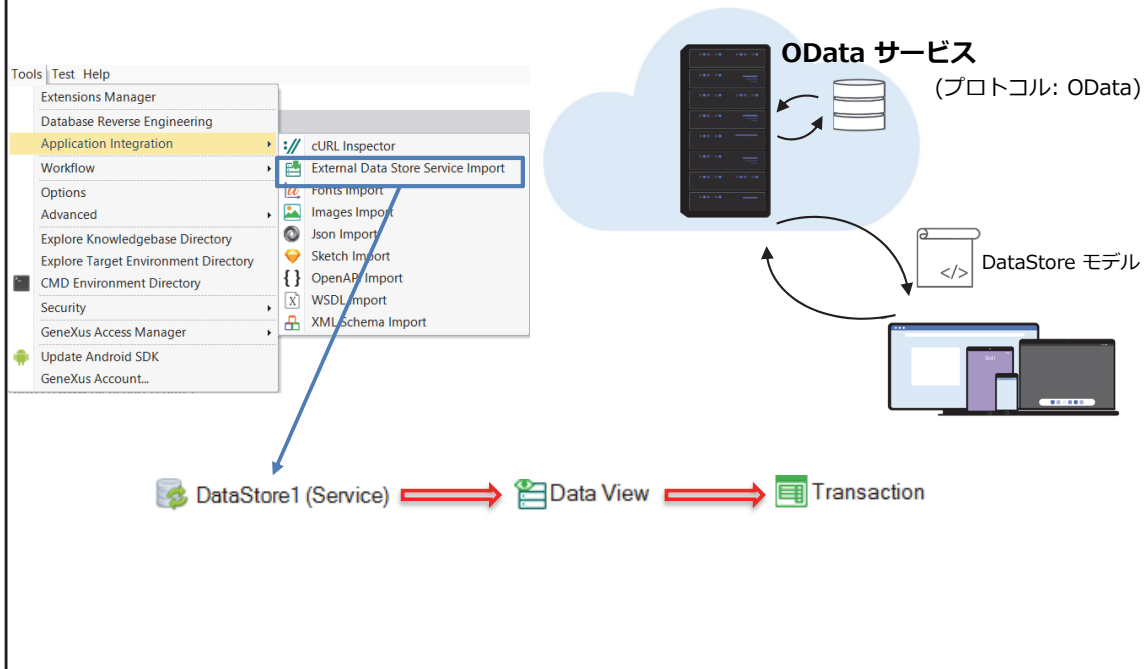


Web サービスを GeneXus アプリケーションに統合するには、まず [ツール] > [アプリケーションの統合] を選択します。次に、その Web サービスが SOAP プロトコルに基づく場合は [WSDL インポート] を、REST アーキテクチャの場合は [OpenAPI インポート] を選択します。

選択すると、Web サービスのタイプに応じたウィザードがトリガーされます。Web サービスが SOAP の場合は、外部オブジェクトが GeneXus で自動的に作成され、Web サービスに関連付けられます。また、そのデータの管理に必要な構造化データタイプも作成されます。

Web サービスが REST の場合は、いくつかの GeneXus オブジェクトが自動的に作成されます (それらのオブジェクトは、通常、単一のモジュール (WebServices など) に含まれます)。また、ナレッジベース内のこれらのオブジェクトを使用して、サービスを自動的に実行することもできます。ウィザードが完了すると、呼び出すプログラムが API フォルダ内に、データを管理するための SDT が Model フォルダ内に置かれます。

GeneXus で Web サービスを利用する方法 (続き)



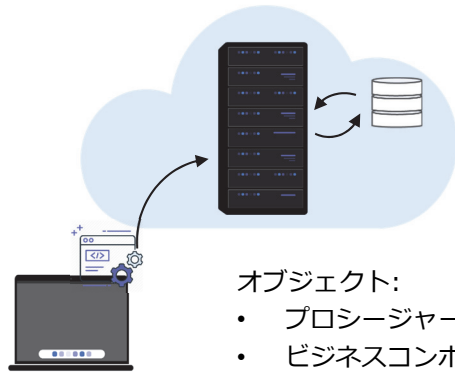
OData サービスは特殊なタイプの Web サービスです。これらのサービスは、Web サイトを介してデータベースのレコードを挿入、変更、または削除する操作を提供する、Open Data Protocol (OData) を使用します。

SOAP や REST Web サービスの場合は、まず定義をインポートする必要がありますでしたが、OData サービスを利用する場合は、最初にそのサービスに含まれているデータベースエンティティで構成されるモデルをインポートする必要があります。

そのためには、[ツール] > [アプリケーションの統合] を選択し、[外部のデータストアサービスをインポート] を選択します。このプロセスでは Service タイプのデータストアを作成して外部データストアに関連付けるため、この機能は GeneXus の製品版でのみ使用できます。

ウィザードの実行後、モデルに含まれていたエンティティの数だけトランザクションオブジェクトが作成され、アプリケーション内のほかのトランザクションと同じように操作できます。

GeneXus を使用して Web サービスを公開する方法



オブジェクト:

- プロシージャー
- ビジネスコンポーネント
- データプロバイダー

Properties	
Filter	
Procedure: MyWebService	
Name	MyWebService
Description	My Web Service
Module/Folder	Root Module
Main program	False
Call protocol	Internal
Execute in new LUW	False
Qualified Name	MyWebService
Object Visibility	Public
Interoperability	
Expose as Web Service	True
Web Service Protocol	
SOAP Protocol	True
REST Protocol	True
Generate OpenAPI interface	Use Environment property value

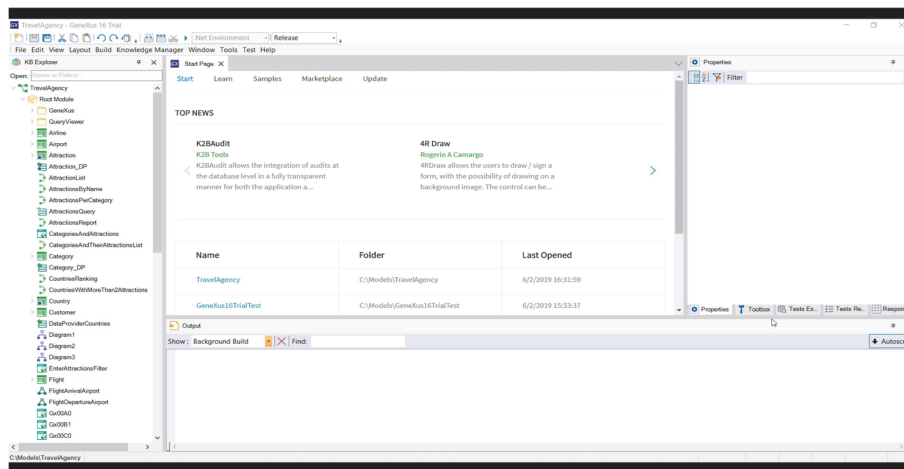
GeneXus では、Web サービスを作成して Web サーバー上で公開することもできます。

Web サービスとして公開できる GeneXus オブジェクトは、プロシージャー、ビジネスコンポーネント、およびデータプロバイダーです。

これらのオブジェクトのいずれかを Web サービスとして公開するには、[Expose as Web Service] プロパティを True に設定し、SOAP プロトコルを使用するか、REST を使用するか、両方を使用するかを指定します。

サービスでデータベースに対する CRUD 操作を提供したい場合は、OData プロトコルに従って必要な情報を生成することもできます。

REST Web サービスの使用例



[デモ: <https://youtu.be/bvmt0Gjcxpw>]

例として、新しい航空会社をデータベースに挿入する REST Web サービスを作成し、アプリケーションからそれを呼び出すとします。

そのためには、まず Airline トランザクションを開き、[Business Component] プロパティを True に設定します。これを保存します。

次に、航空会社のデータをパラメーターで受け取り、Airline テーブルに新しいレコードを挿入するプロシージャオブジェクトを作成します。

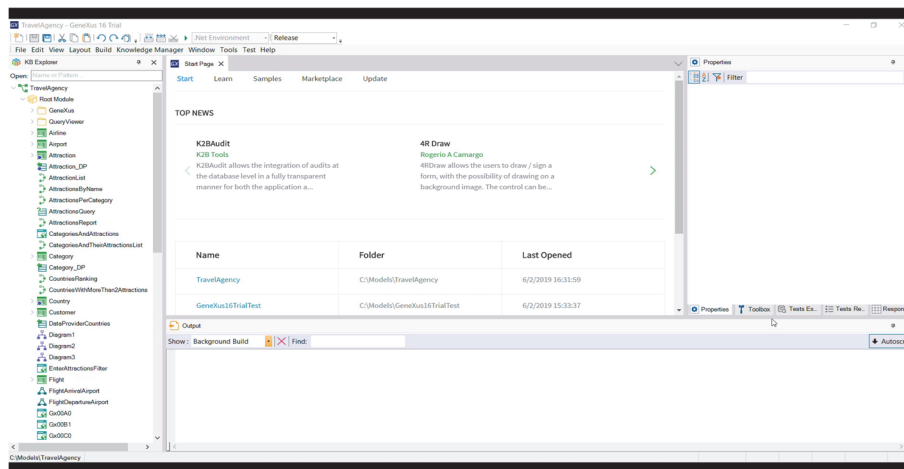
このプロシージャに CreateNewAirlineWS という名前を付け、2 つの入力変数 &AirlineName および &AirlineDiscountPercentage を使用して Parm ルールを作成します。[変数を追加] を使用してこの 2 つの変数を作成してから、変数のセクションで Airline という別の変数を作成します。この変数は Airline ビジネス コンポーネント タイプです。

ソースには、パラメーターで受け取ったデータを使用して航空会社を挿入するコードを記述します。AirlineId の値は自動採番されるため、入力は不要です。

次に、プロシージャオブジェクトのプロパティを開き、[Expose as Web Service] を True に、[SOAP Protocol] を False に設定します。また、[REST Protocol] は True のままにして、[Generate OpenAPI interface] は Yes にします。後者の設定により、Web サービスの定義を OpenAPI 標準で生成できます。プロシージャを右クリックして [これだけをビルド] を選択すると、このサービスが Web サーバー上で公開されます。ここではローカルマシンで公開されます。出力ウィンドウには、Web サービスの REST API のドキュメントが、その定義とともに生成されたことを確認するメッセージが表示されます。

この Web サービスをインポートする前に、すべてを格納する WebServices という新しいモジュールを作成します。

REST Web サービスの使用例



[デモ: <https://youtu.be/bvmt0Gjcxpw>]

Web サービスをインポートするには、[ツール] > [アプリケーションの統合] を選択し、[OpenAPI インポート] を選択します。[ファイルパス/URL] に次のように入力します: C:\Models\TravelAgency\CSharpModel\Web\default.yaml。これは、REST API ドキュメントが生成された場所です。また、WebServices モジュールをターゲットとして選択します。すべてが正しくインポートされたことを確認します。WebServices モジュールを開くと、インポートされたプロシーチャーが API フォルダ内に置かれています。また、Model フォルダ内には CreateNewAirlineWSInput という名前の SDT が置かれており、それを開くと、サービスに渡す必要があるパラメーターが含まれていることが分かります。

この Web サービスが適切に機能するかどうかをテストするため、CreateNewAirlineUsingWS という Web パネルが作成されています。フォームにボタンをドラッグし、Create airline イベントを呼び出します。[Variables] エレメントを開き、&CreateNewAirlineWSInput という変数を作成します。これは自動的に SDT タイプに設定されます。Web サービスの実行結果に関するフィードバックを受け取るため、&IsSuccess という変数と &HttpMessage という変数も作成します。タイプは既定で割り当てられます。

ボタンをダブルクリックし、イベントで SDT のメンバーをロードします。次に、SDT をパラメーターとして渡して Web サービスを呼び出します。最後に、画面にメッセージを表示するためのコードを記述します。

F5 キーを押します。入力した行が表示されます。作成したばかりの Web パネルを実行します。ボタンをクリックして、航空会社が正しく作成されたという通知を受け取ります。Airline トランザクションを選択して確定すると、必要な航空会社が実際に追加されたことを確認できます。

SOAP Web サービスの使用例

<https://training.genexus.com/en/learning/courses/genexus/genexus-16-course-analyst/web-services-introduction>

ここでは REST タイプの Web サービスの使用例をお見せしました。SOAP サービスの使用法に関する動画は、スライドに示すリンクから視聴できます。

OData Web サービスの使用例

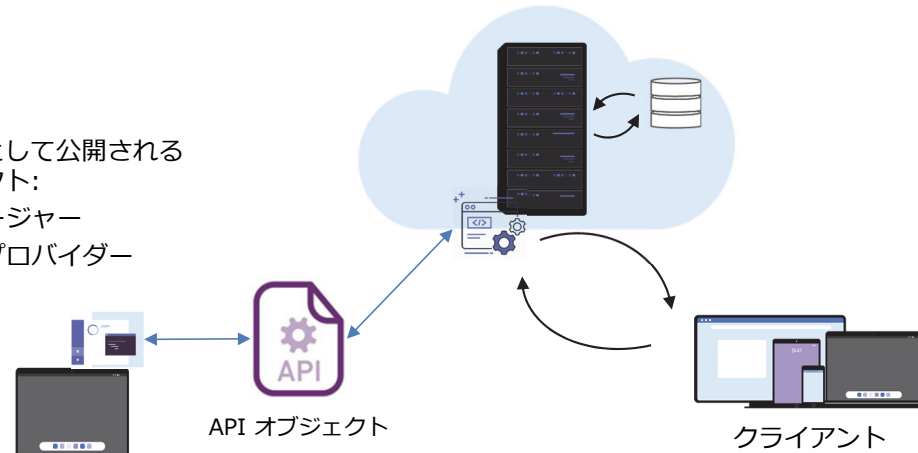
<https://training.genexus.com/en/learning/courses/genexus/genexus-16-course-analyst/web-services-that-access-the-providers-database>

OData プロトコルでの Web サービスの使用例については、スライドに示すドキュメントを参照してください。

API オブジェクト

サービスとして公開される
オブジェクト:

- プロシージャー
- データプロバイダー



次に、API オブジェクトを使用してオブジェクトをサービスとして公開する方法を見てみましょう。

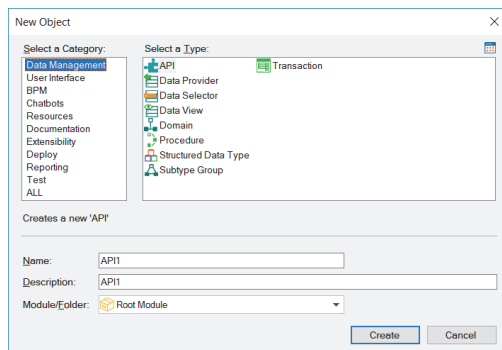
API (Application Programming Interface) オブジェクトは、アプリケーションのオブジェクト (プロシージャーやデータプロバイダーなど) へのアクセスインターフェースをプログラミングで定義するための GeneXus オブジェクトです。つまり、外部アプリケーションが、Web サービスとして公開されているこれらのオブジェクトにアクセスできるようにします。

API オブジェクトは、インターフェースを実装の詳細から分離する中間層を追加します。これにより、将来オブジェクトに対してプログラミング変更が行われても、外部アプリケーションによる呼び出しには影響が生じません。

たとえば、ナレッジベース内のオブジェクトの内部名と、サービスとして公開される名前をマッピングします。また、サービスの呼び出しには影響を与えずに、パラメーターのタイプや名前を変更したり、アクセスパスを変更したりできます。

この抽象化により、柔軟性が大幅に向上します。サービスを進化させても、それを利用するアプリケーションに変更に対応するためのコード変更を行う必要がないからです。

API オブジェクト (続き)



Properties	
Filter	
API: API1	
Name	API1
Description	API1
Main program	True
REST Protocol	True
gRPC Protocol	False
Generate OpenAPI interface	No
Services base path	API1
Module/Folder	Root Module
Qualified Name	API1
Object Visibility	Public
Miscellaneous	
Generate Object	True

API オブジェクトを作成するには、データ管理カテゴリに移動し、API タイプを選択します。開発環境においてプロパティを通じて、その名前、アドレス、使用するプロトコルなどを定義できます。

Google が開発した最新のオープン ソース フレームワークである gRPC プロトコルでは、高パフォーマンスかつ双方向でリモートプロシーチャーを呼び出すことができます。

API オブジェクト (続き)

The top screenshot shows the 'Service Source' tab with the following code:

```

1 AttractionsInfo{
2
3   RankingCountriesAttraction(out:&SDTCountries) => RankingCountriesWithAttractionsQty(out:&SDTCountries);
4
5   ListAttractionsByName(in:&NameFrom, in:&NameTo) => AttractionsByName(in:&NameFrom, in:&NameTo);
6
7 }

```

The middle screenshot shows the 'Events' tab with a dropdown menu set to 'After'.

The bottom screenshot shows the 'Variables' tab with the following variables defined:

Name	Type
Standard Variables	
PgmDesc	Character(256)
PgmName	Character(128)
RestCode	Numeric(3,0)
RestMethod	HttpMethod, GeneXus
RestMsg	Character(255)
Autodefined Variables	
SDTCountries	SDTCountries
NameFrom	Attribute:AttractionName
NameTo	Attribute:AttractionName

API オブジェクトには 3 つのエレメントがあります: [Service Source]、[Events]、および [Variables] です。

公開するサービスの名前は、[Service Source] エレメントで宣言構文を使用し、サービスとして公開するナレッジベース内の GeneXus オブジェクトの名前、および対応するパラメーターを使用して定義されます。

この例では、RankingCountriesWithAttractionsQty という名前で以前作成したデータプロバイダーが、RankingCountriesAttractions という名前でサービスとして公開されており、公開するパラメーターはオブジェクトのものと同じです。ListAttractionsByName という名前で公開する AttractionsByName リストについても同様です。

この定義により、将来、GeneXus オブジェクトの名前またはパラメーターを、そのオブジェクトの公開方法の定義を変えることなく変更できます。

[Events] エレメントには Before と After があります。これらのイベントを使用して、オブジェクトをサービスとして呼び出す前または後にアクションを実行できます。

[Variables] エレメントには、標準タイプのいくつかの変数を使用して、API オブジェクトの特性を実行時に定義または変更できます。

Web サービスに関する詳細情報

WSDL: <http://wiki.genexus.jp/hwikibypageid.aspx?6181>

OpenAPI: <http://wiki.genexus.jp/hwikibypageid.aspx?31864>

OData: <http://wiki.genexus.jp/hwikibypageid.aspx?40713>

API Object: <http://wiki.genexus.jp/hwikibypageid.aspx?46151>

GeneXus での Web サービスの詳細については、スライドに示すリンクから、各 Wiki を参照してください。