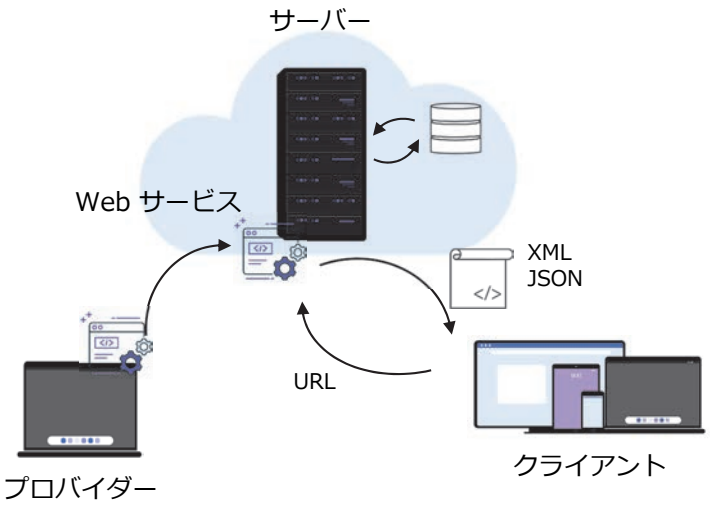
GeneXus の Web サービス

概要

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。

GeneXus

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。

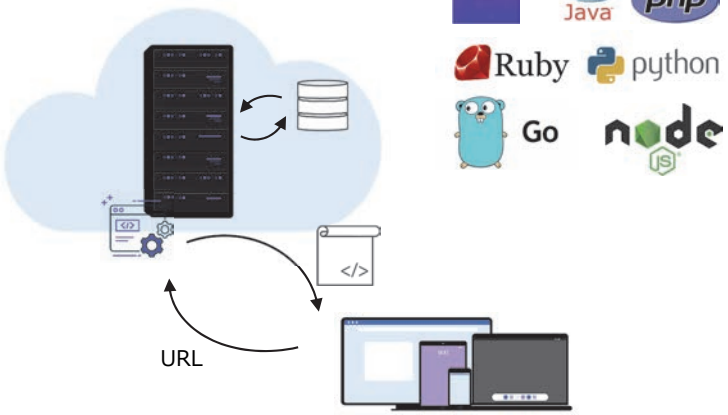
クライアントアプリケーションは、サービスにアクセスする際に場所 (URL) を指定して呼び出し、必要なパラメーターを渡します。

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。

GeneXus

Web サービスのプロトコル

プロトコル/標準	
SOAP	WSDL
REST	OpenAPI



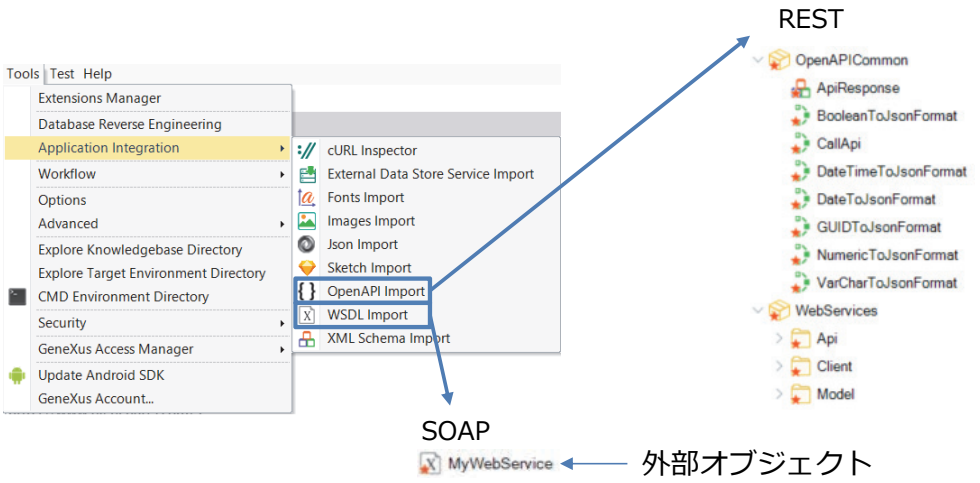
Web サービスはさまざまな標準に基づいて開発されます。業界で最もよく使用されているのは SOAP と REST です。各標準では、Web サービスで利用可能な機能に関する情報の公開方法が定義されています。

SOAP Web サービスは WSDL (Web サービス記述言語) で記述された定義を使用し、REST サービスは 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

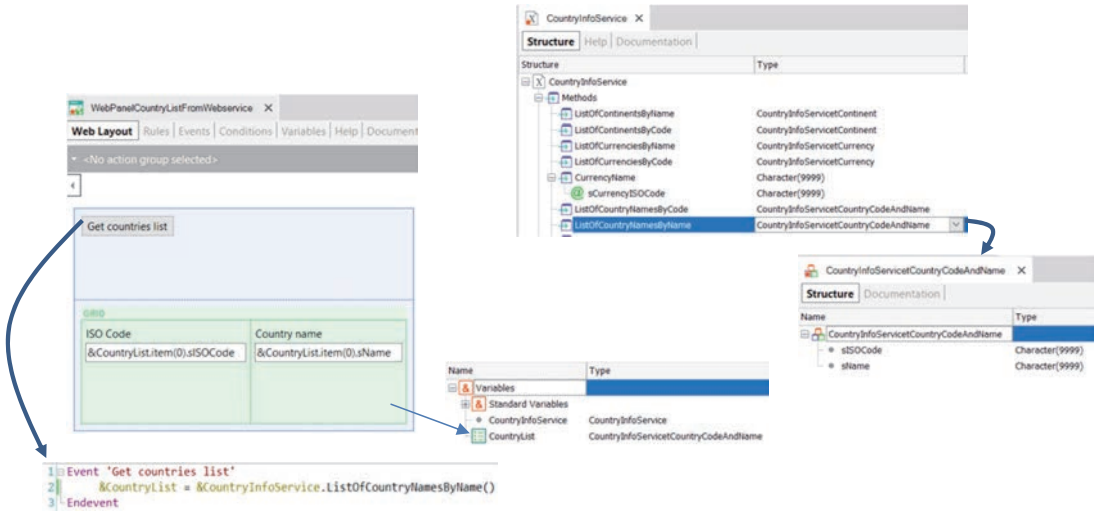


Web サービスを GeneXus アプリケーションに統合するには、まず [ツール] > [アプリケーションの統合] を選択します。次に、その Web サービスが SOAP プロトコルに基づく場合は [WSDL インポート] を、REST アーキテクチャの場合は [OpenAPI インポート] を選択します。

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 (通常、これらは WebServices などのモジュールに含めます)。また、ナレッジベース内で、これらのオブジェクトを介してサービスを自動的に実行できるようになります。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。

デモ: SOAP Web サービスへのアクセス



それでは、SOAP Web サービスからアプリケーションに国のリストをインポートしてみましょう。
これを行うためには、メニューオプションの [ツール] > [アプリケーションの統合] > [WSDL インポート] にアクセスし、画面に表示されている URL (http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL) を入力します。[次へ] をクリックします。
CountryInfoService というサービスが見つかります。インポートしたオブジェクトに順序を付けるため、SOAPWebService フォルダに保存します。推奨の接頭語はそのままで、[次へ] をクリックします。
サービス デスクリプション ノードで [+] をクリックします。Web サービスで提供される関数の一覧が開きます。国名のリストを取得する ListOfCountryNamesByName、その国の通貨の種類を取得する CountryCurrency、国旗のイメージを取得する CountryFlag などがあります。

[インポート] をクリックすると、GeneXus によって Web サービスの定義がインポートされます。出力ウィンドウに一連の構造化データタイプおよびその他のコンポーネントがインポートされていることが示されます。最後に、Knowledge Base Navigator に SOAPWebService フォルダがあることを確認します。この中には、外部オブジェクト CountryInfoService と、サービスの各メソッドから取得した情報を格納できる一連の SDT があります。

CountryInfoService をダブルクリックすると、CountryInfoService Web サービスで提供されるすべてのメソッドが外部オブジェクトに組み込まれていること、各メソッドに必要なパラメーターと、返されるデータタイプの詳細が示されていることが分かります。ここで、国のリストを名前順で返す ListOfCountryNamesByName メソッドを見てみましょう。

WebPanelCountryListFromWebService という名前の Web パネルを作成します。その変数に、外部オブジェクトのデータタイプを自動的に取得する CountryInfoService タイプの変数を作成します。
外部オブジェクトの定義に戻ると、LisfOfCountryNamesByName メソッドで返されるデータタイプが CountryInfoServicetCountryCodeAndName という SDT であることが確認できます。これを開くと、国の ISO コードと国名が格納されていることが分かります。
Web パネルに戻り、データタイプが SDT の CountryInfoServicetCountryCodeAndName である CountryList 変数を作成し、それをコレクションとして設定します。
次に、フォーム内にボタンをドラッグし、イベント名 [Get countries list] を付けます。これをダブルクリックして、イベント内に変数 &CountryList を挿入し、変数 &CountryInfoService を割り当てます。ピリオドを入力すると、Web サービスのすべてのメソッドにアクセスできます。ここでは ListOfCountryNamesByName を選択します。
フォームに戻り、SDT に基づく CountryList 変数をドラッグして、[OK] をクリックします。
これを実行します。

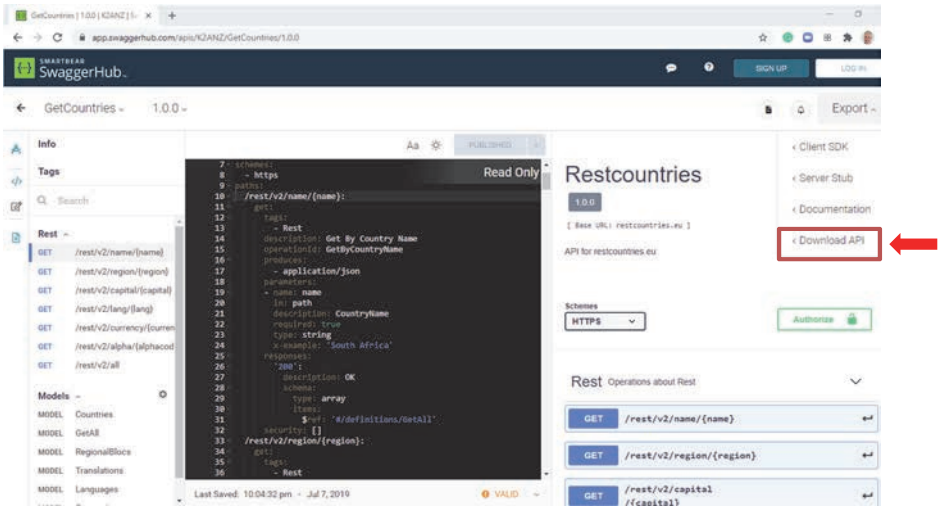
Web パネル WebPanelCountryListFromWebService を開き、[Get countries list] ボタンをクリックすると、想定したとおりに ISO コード付きでアルファベット順に並べられた国のリストが取得されます。

このデータは、旅行代理店のアプリケーションで、国でフィルタする際の選択リストとして使用することも、ほかの用途に使用することもできます。

GeneXus

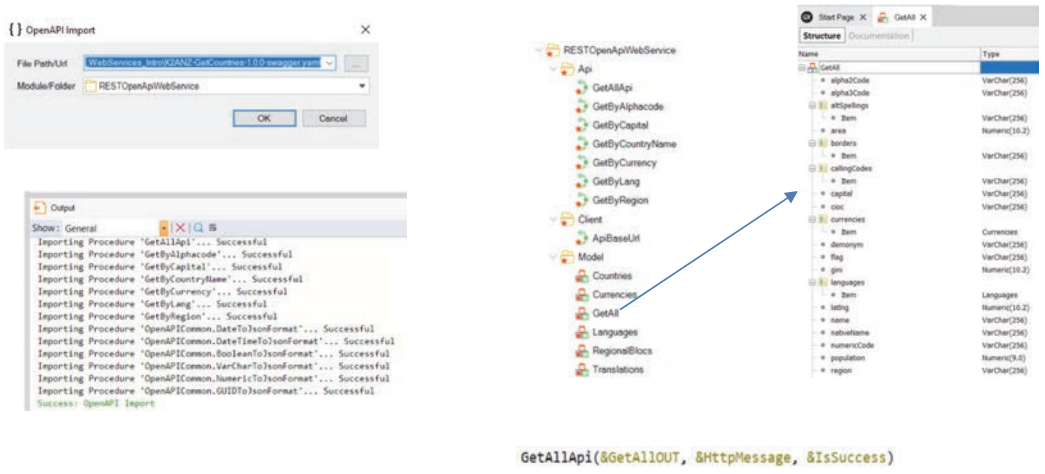
デモ: OpenAPI プロトコルを使用した REST Web サービスへのアクセス

現在、GeneXus では OpenAPI バージョン 2.0 をインポート可能



次に、OpenAPI Specification (Swagger Specification) に準拠した REST Web サービスをインポートするケースを見てみましょう。
Web ページ (https://app.swaggerhub.com) から GetCountries という API のサンプルを入手します。[Download API] オプションを使用して .yaml ファイルをダウンロードします。

デモ: OpenAPI プロトコルを使用した REST Web サービスへのアクセス

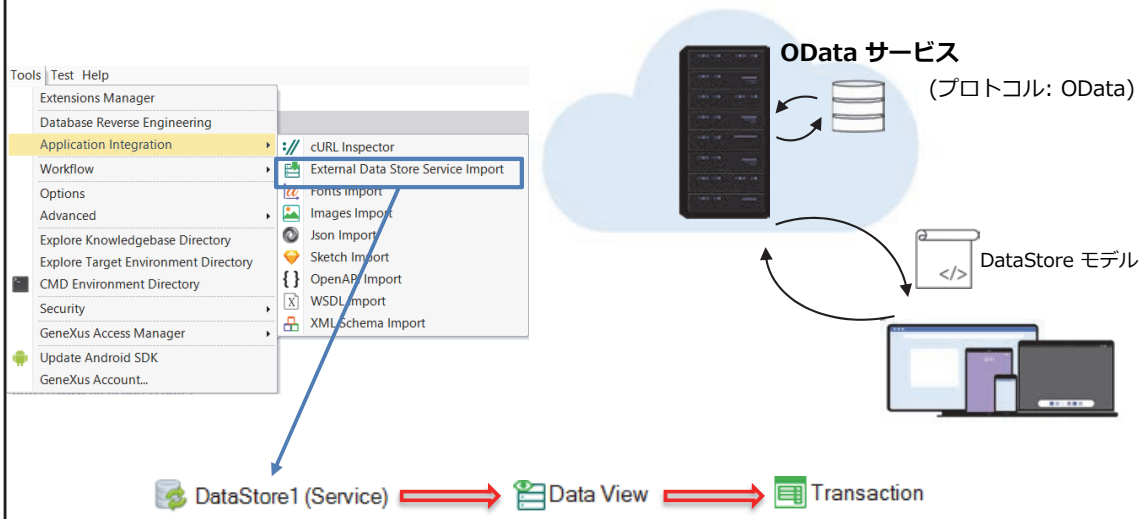


[ツール] > [アプリケーションの統合] > [OpenAPI インポート] を選択し、ダウンロードした .yaml ファイルを選択します。フォルダには RESTOpenApiWebService を選択して [OK] をクリックします。
出力ウィンドウを見ると、いくつかのエレメントがインポートされていることが分かります。インポートが完了したら、保存先のフォルダを開きます。呼び出し可能なメソッドが含まれている Api フォルダ、ApiBaseURL メソッドが含まれている Client フォルダ、および前のメソッドによって返されたデータタイプの SDT がいくつか含まれている Model フォルダが作成されたことが分かります。

GetAll という SDT を開いてみると、国から取得できるすべてのデータを確認できます。また、国のリストを取得したい場合は、GetAllApi メソッドを呼び出すことができます。GetAllApi メソッドは、変数 &GetAllOut (GetAll エレメントのコレクション)、&HttpMessage 、およびブール型の変数 &IsSuccess のデータを返します。

これで、&GetAllOut コレクションを参照して国のデータを取得できます。

GeneXus で Web サービスを利用する方法 (続き)



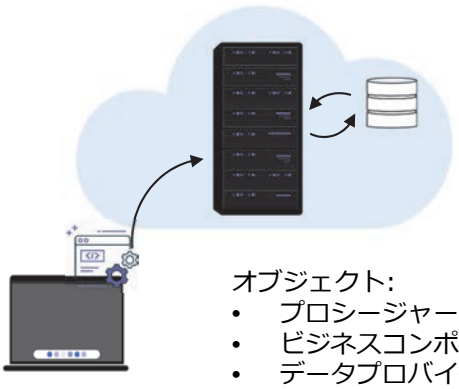
OData サービスは特殊なタイプの Web サービスです。これらのサービスは、Web サイトを介してデータベースのレコードを挿入、変更、または削除する操作を提供する、Open Data Protocol (OData) を使用します。

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 Transaction 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

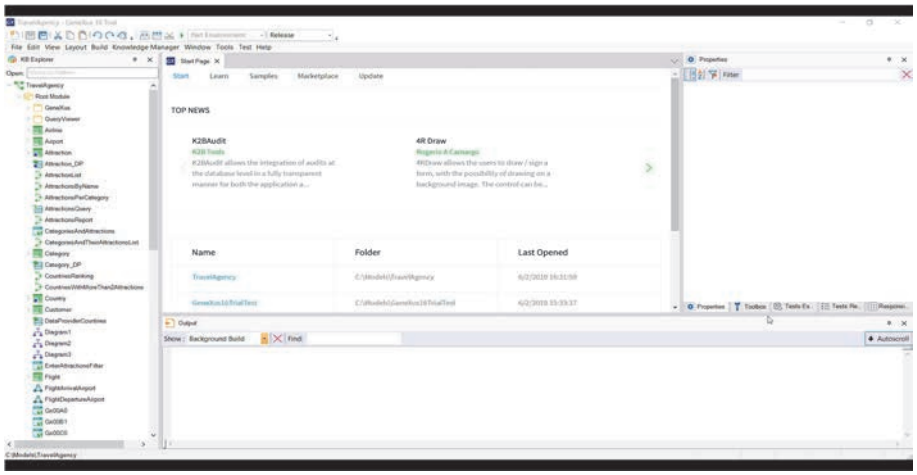
GeneXus では、Web サービスを作成して Web サーバー上で公開することもできます。

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 Procedure オブジェクトを作成します。

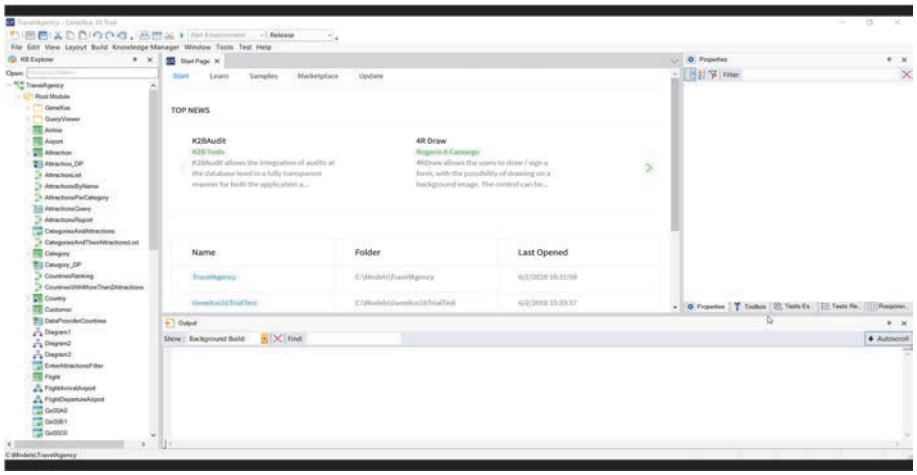
このプロシージャーに CreateNewAirlineWS という名前を付け、2 つの入力変数 &AirlineName および &AirlineDiscountPercentage を使用して Parm ルールを作成します。[変数を追加] を使用してこの 2 つの変数を作成してから、変数のセクションで Airline という別の変数を作成します。この変数は Airline ビジネス コンポーネント タイプです。

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、プロシージャー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。プロシージャーを右クリックして [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



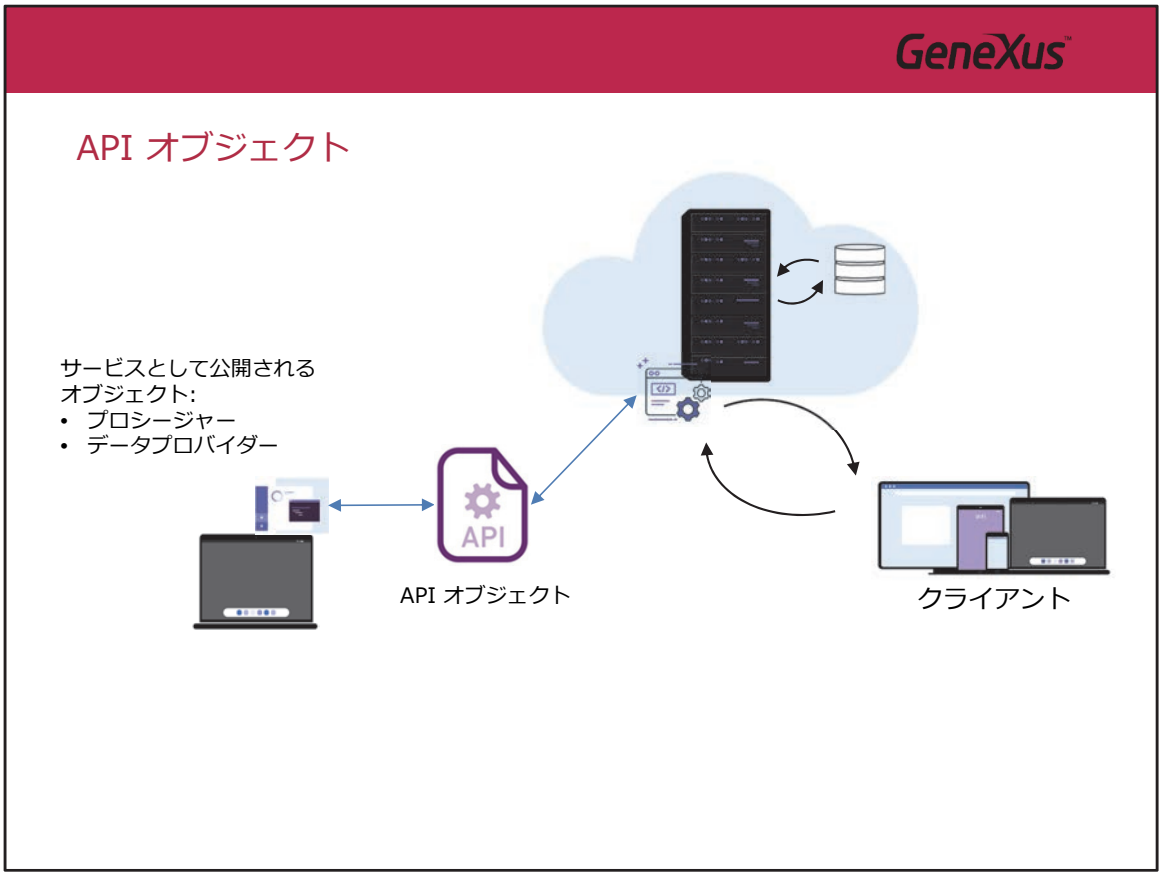
[デモ: https://youtu.be/bvmt0Gjcxpw]

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ジャーが 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。



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。

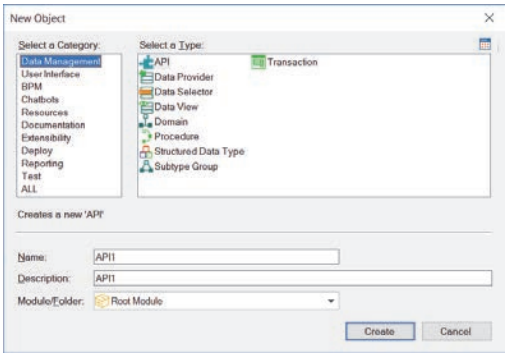
API オブジェクトは、インターフェースを実装の詳細から分離する中間層を追加します。これにより、将来オブジェクトに対してプログラミング変更が行われても、外部アプリケーションによる呼び出しには影響が生じません。

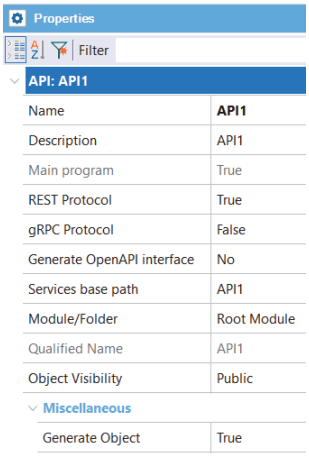
たとえば、ナレッジベース内のオブジェクトの内部名と、サービスとして公開される名前をマッピングします。また、サービスの呼び出しには影響を与えずに、パラメーターのタイプや名前を変更したり、アクセスパスを変更したりできます。

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。

GeneXus

API オブジェクト (続き)

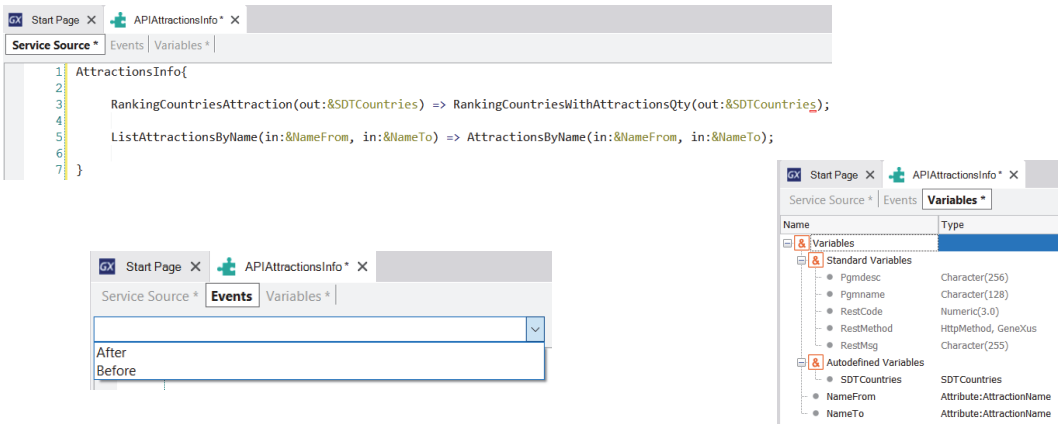




API オブジェクトを作成するには、データ管理カテゴリに移動し、API タイプを選択します。開発環境においてプロパティを通じて、その名前、アドレス、使用するプロトコルなどを定義できます。

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 gPRC 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ジャーを呼び出すことができます。

API オブジェクト (続き)



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。