

尾崎 浩司

株式会社ミガロ.

システム事業部 システム3課

Delphi/400を使用したWebサービスアプリケーション

インターネット技術を応用し、XML 処理を行う
というたいへん敷居が高く感じる。
実は、Delphi/400 を用いると
それらは容易に使用可能である。

- Webサービスとは
- SOAPとREST
- SOAPの使用方法
- RESTの使用方法
- 最後に



略歴
1973年8月16日生れ
1996年三重大学工学部卒
1999年10月株式会社ミガロ. 入社
1999年10月システム事業部配属

現在の仕事内容
ミガロ. 入社以来、主に Delphi/400
を利用した受託開発を担当している。

Web サービスとは

Web サービスとは、インターネットの技術を活用し、遠隔サイトにあるアプリケーションの機能をネットワークを通じて、自社アプリケーションから利用できるようにしたものである。

インターネットが発達したおかげで、現在では有用な情報が容易に取得できるようになった。だが、従来の Web アプリケーションでは、取得した情報をそのまま自社のアプリケーションに取り込むことができなかつたため、自社アプリケーション自体が有用な情報を直接活用するということは難しかった。しかし、Web サービスの普及により、企業の枠を超えた情報の連携および活用が可能になった。

一例を挙げると、従来であれば「乗換案内」のような路線情報を提供するサイトを使って、使用した路線の交通費をブラウザで確認し、その結果を自社の出張精算システムに手で入力するといった手順が必要であった。しかし、路線情報サ

イトの Web サービスを使用することで、直接、出張精算システムと連携することが可能になったのである。

つまり、使用した路線を指定すると、自動的に交通費を出張精算システムに登録するといったことができるわけだ。しかも、路線情報を提供するサイトは常に最新の運賃情報を提供しているため、自社アプリケーションでは、運賃マスターといった情報をまったく管理する必要がないというメリットもある。

このように Web サービスを使用することで、自社のアプリケーションの利便性を大きく向上させることができるのである。

では、Web サービスは、どのようにして、システム間の連携を可能にしているのだろうか？ 従来の Web システムでは、人がサイトにアクセスして処理結果をブラウザで見るのに対し、Web サービスでは、プログラムがサイトにアクセスして処理結果を XML 形式で受け取るというのが特徴である。

インターネットの技術を応用して、

XML で処理を行うというすごく敷居が高いような話にも思えるが、実は Delphi/400 を用いると、これらを容易に使用することが可能である。本稿では、具体例を挙げながら、Delphi/400 を使用した Web サービスの活用方法について触れていくこととする。

SOAPとREST

Web サービスを使用するにあたって、Web サービスを提供しているサイトを調査すると、おそらく使用方式に「SOAP」あるいは「REST」といった言葉がでてくるであろう。

SOAP とは、Simple Object Access Protocol の略で、SOAP メッセージという XML によってメッセージ交換を行う方法である。

対して、REST とは、Representational State Transfer の略で、HTTP の GET メソッドを使って指定された URL にアクセスすると、XML が返ってくるというものである。

図1

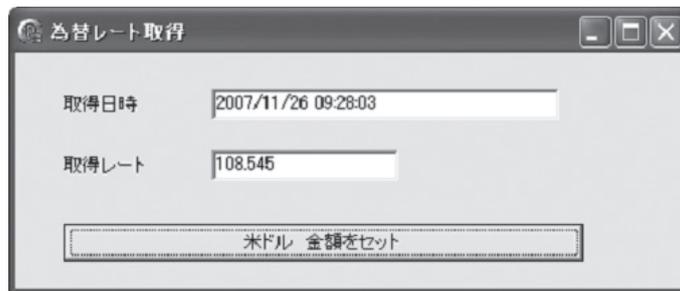


図2

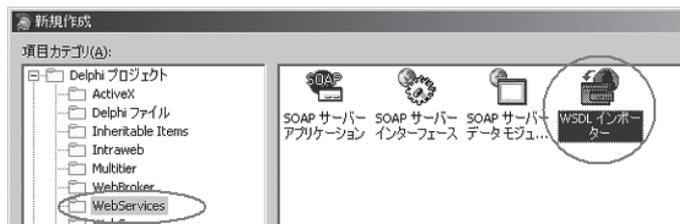
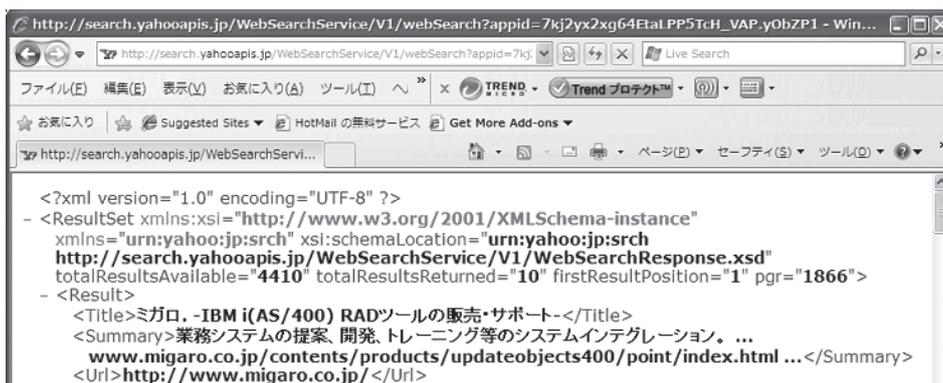


図3



図4



一般的には、SOAP は、XML メッセージのやり取りを定義する必要があるため難易度が高く、REST は、URL を指定するだけで情報が XML で受け取れるため難易度が低いといわれる。

しかし、どちらの手法も Delphi/400 では容易に使用可能である。それでは、それぞれの使用方法について具体例を挙げながら説明を進めていこう。

SOAP の使用方法

SOAP を使用するのに必要なのが「WSDL」と呼ばれるものだ。WSDL というのは、Web Services Description Language の略で、XML でメッセージ交換を行うためのルールが定義されたものである。(WSDL 自体が XML で記述されている)。

この WSDL を使用することで、Delphi/400 アプリケーションから容易に Web サービスが利用可能になるのだ。

今回、SOAP の具体的な使用例として「Webservicex.Net」(<http://www.webservicex.net/WCF/webServices.aspx>) と呼ばれるサイトで公開されている、「Currency Convertor」という為替レート情報取得 Web サービスを使用してみよう。

この Web サービスを使用したサンプルプログラムを実行したアプリケーションが、図 1 である。このアプリケーションは、ボタンをクリックすると、米ドルに対する日本円のレートを表示するというものだ。【図 1】

Currency Convertor のサイトを確認すると、WSDL ロケーションとして、以下のように記載されていることがわかる。この情報をもとに、Delphi/400 で使用可能なクラスを生成することが可能なのである。

<http://www.webservicex.net/CurrencyConvertor.asmx?WSDL>

作成手順は次の通りである。

- ①新規プロジェクトを作成
- ②メニューより [ファイル] → [新規作成] → [その他] を選択
- ③ [WebServices] から [WSDL インポート] を選択 【図 2】

④ WSDL ファイル URL を入力 【図 3】

⑤ [完了] ボタンを押下

上記ウィザードを実行すると、WSDL を解析し、Delphi/400 で使用可能なクラスユニットが自動生成されるので、これをアプリケーションから利用すればよい。

ソース 1 は、先程の図 1 のサンプルアプリケーションのソースコードである。【ソース 1】

自動生成された CurrencyConvertor ユニットの uses 節に含めることにより、為替レート取得のために定義されたクラスやメソッドが使用可能になったことがわかるであろう。

REST の使用方法

REST は、先程も述べた通り、Web サービスを提供するサーバーに対して URL を与えると結果として XML を受け取れるというものだ。

REST の具体例としては、「Yahoo! デベロッパーネットワーク」(<http://developer.yahoo.co.jp/>) で提供されている Web サービスを紹介しよう。それらの Web サービスは、Yahoo! JAPAN で提供されているさまざまな機能を Web サービス化したものとなる。(サービスを使用するには、アプリケーション ID が必要なので、事前に取得しておこう)。

例えば Yahoo! 検索 Web サービスを使用し、キーワード "migaro" で問い合わせを行うには、以下のような URL を使用する。

<http://search.yahooapis.jp/WEBSearchService/V1/webSearch?appid=<アプリケーション ID>&query=migaro>

これをブラウザに入力し実行すると、図 4 のように XML が取得できることがわかる。【図 4】

Delphi/400 から REST の使用

このように REST とは、HTTP リクエストに対して XML をレスポンスとして返すものなのである。では、これを Delphi/400 から使用するにはどうすれ

ばよいだろうか？

Delphi/400 で HTTP クライアントを実装するのは、TIdHTTP と呼ばれるコンポーネントである。URL を指定して結果を取得するには、Get メソッドを使用すればよい。

ここで具体的なサンプルアプリケーションを紹介しよう。図 5 のように TIdHTTP コンポーネントを貼り付けたフォームを用意し、ソース 2 のようにボタンクリックイベントを記述する。【図 5】 【ソース 2】

本サンプルは、Yahoo! ニュースのトピック用 Web サービスであるが、実行すると、メモコンポーネント内に、先程のブラウザの場合と同じように XML が表示されることがわかるだろう。

Delphi/400 から XML の扱い

これで Delphi/400 で XML が取得できることはご理解いただけたと思う。では、どうすればこの XML をプログラムからアクセスしやすくてできるのだろうか？

XML 文書というのは、HTML と同じマークアップ言語と呼ばれるものだが、HTML とは違い使用するタグ自体の定義を自由に行えるのが特徴である。その定義にあたるのが XML スキーマと呼ばれるものだ。実はこの XML スキーマを使用すると、Delphi/400 から XML の扱いが容易になるのである。

Yahoo! ニュースのトピック用 Web サービスのサイトを確認すると、レスポンス用の XML スキーマの定義 (拡張子 xsd) が記載されているのがわかる。これを Delphi/400 に取り込めばよいのだ。その手順は次の通りである。

- ① [新規作成] → [その他] を選択
- ② XML フォルダの [XML データバインディング] を選択
- ③ ソースコード欄に XSD ファイルを指定 【図 6】
- ④ ウィザードを進み、データバインディングの [設定を保存しない] に設定。【図 7】
- ⑤ [完了] ボタンを押下

上記ウィザードを実行すると、XML スキーマを解析してできたクラスユニットが自動生成される。このユニットを使

図5



図8

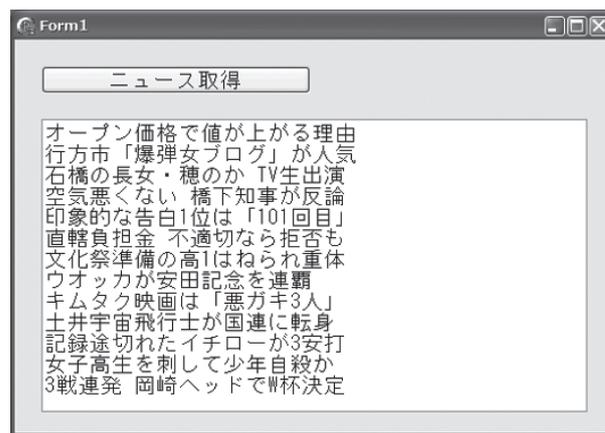


図6

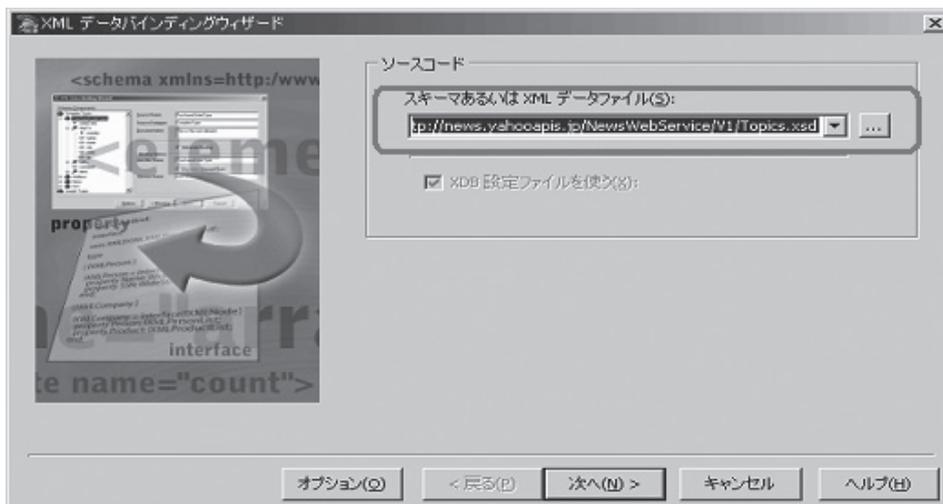
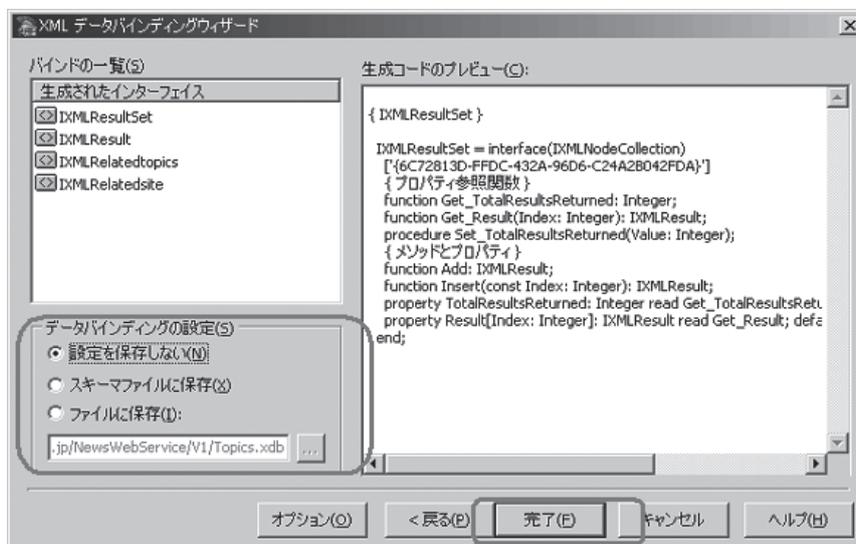


図7



用することで、容易に XML の各要素にアクセス可能になるのである。

なお、XML 文書を Delphi/400 から使用するには、TXMLDocument コンポーネントを使用するのであわせて覚えておこう。

では、先程のサンプルアプリケーションを改良してみよう。TXMLDocument コンポーネントを1つ追加した後に、XML スキーマをデータバインディングウィザードでユニットを作成し、ソース3のようにプログラムを修正する。【ソース3】

XML スキーマの定義にもとづき、ニューストップICKの結果セット集合体や各要素がプログラムから使用可能になっているのわかるだろう。

このサンプルプログラムを実行すると、図8のようにメモコンポーネントにトピックが一覧表示されるのである。また、このサンプルプログラムでは、ニュースの件数やトピック名等がクラス変数として取得できているのもわかる。【図8】

最後に

Web サービスを Delphi/400 から使用するのには、SOAP であっても REST であっても容易であることがおわかりいただけたであろう。

今回紹介した「Websvcx.Net」や「Yahoo! デベロッパーネットワーク」だけでなく、他にも多様な Web サービスが提供されているので、いろいろ試してみしてほしい。そして、自社のアプリケーションと Web サービスをどう組み合わせるか、いろいろ検討してほしい。

自社アプリケーションに Web サービスを活用することで、これまで以上の利便性をユーザーに提供できると思われるので、ぜひともチャレンジしていただきたい。

■

ソース1

```
25 var
26   Form1: TForm1;
27
28 implementation
29
30 uses CurrencyConverter;
31
32 {$R *.dfm}
33
34 procedure TForm1.Button1Click(Sender: TObject);
35 var
36   rw: CurrencyConverterSoap;
37   cRate: System.Currency; // ---- 通常のCurrency型はSystemユニットに定義
38   dDate: TDateTime;
39 begin
40   // 通貨情報を取得
41   rw := GetCurrencyConverterSoap();
42   cRate := rw.ConversionRate(USD, JPY);
43   dDate := Now;
44
45   // 取得結果を画面表示
46   edtDate.Text := FormatDateTime('YYYY/MM/DD HH:NN:SS', dDate);
47   edtRate.Text := CurrToStr(cRate);
48 end;
49
50 end.
```

ソース2

```
25 implementation
26
27 {$R *.dfm}
28
29 procedure TForm1.btnGetClick(Sender: TObject);
30 const
31   cCMD = 'http://news.yahooapis.jp/NewsWebService/V1/Topics'
32         + '?appid=<実際はアプリケーションIDを記述>';
33
34 var
35   sData: String;
36 begin
37   //URLを指定して実行結果を取得
38   sData := IdHTTP1.Get(cCMD);
39   //文字コードを変換 (UTF8->Ansi)
40   sData := Utf8ToAnsi(sData);
41   //取得結果をメモコンポーネントに表示
42   Memo1.Lines.Text := sData;
43 end;
44 end.
```

ソース3

```
28 uses newsyahooapisjpNewsWebServiceV1Topics;
29
30 {$R *.dfm}
31
32 procedure TForm1.btnGetClick(Sender: TObject);
33 const
34   cCMD = 'http://news.yahooapis.jp/NewsWebService/V1/Topics'
35         + '?appid=<実際はアプリケーションIDを記述>';
36
37 var
38   i: Integer;
39   sData: String;
40   AXMLResultSet: IXMLResultSet; //結果セット
41   AXMLResult: IXMLResult; //1件の結果 |
42 begin
43   //メモコンポーネントをクリアする
44   Memo1.Clear;
45
46   //URLを指定して実行結果を取得
47   sData := IdHTTP1.Get(cCMD);
48   //取得結果をXMLコンポーネントにセット
49   XMLDocument1.LoadFromXML(sData);
50
51   //XML結果セットの取得
52   AXMLResultSet := GetResultSet(XMLDocument1);
53
54   //結果セットより取得した件数分ループで情報を取得する
55   for i := 0 to AXMLResultSet.TotalResultsReturned - 1 do
56     begin
57       //1件の情報を取得する
58       AXMLResult := AXMLResultSet.Result[i];
59       //メモコンポーネントに件名をセットする。
60       Memo1.Lines.Add(AXMLResult.Title);
61     end;
62 end;
```