# Delphi/400

# IntraWebを使用したWeb開発のTips紹介

株式会社ミガロ. システム事業部 システム1課 福井 和彦



### 略歴

生年月日:1972年3月20日 最終学歴:1994年 大阪電気通信大学 工学部卒業 ミガロ入社年月:2001年04月 株式会社ミガロ.入社 社内経歴:2001年04月 システム事業部配属

### 現在の仕事内容:

主にDelphi/400を使用したシステムの受託開発を担当しており、 要件確認から納品・フォローに至るまで、システム開発全般に 携わっている。

# 株式会社ミガロ. システム事業部 システム1課 石山 智也



### 略歴

生年月日:1988年4月5日 最終学歴:2012年 近畿大学 経済学部卒業 ミガロ入社年月:2019年6月 株式会社ミガロ.入社 社内経歴:2019年6月 システム事業部配属

現在の仕事内容: 主にDelphi/400を利用したシステムの受託開発をメインに担当している。 開発スキルの向上を目指し、日々精進している。

### 1.はじめに

近年、テレワークの増加や業務効率化の観点から業務シ ステムのWeb化は注目の存在となっている。 Webシステムはクライアントサーバーシステムとは違 い、クライアントPCにアプリケーションをインストール や再配布を行う手間が要らず、ブラウザとインターネッ ト環境があればどこからでもシステムを使用できること が利点である。

Delphi/400で Web サーバー アプリケーションを構築 するツールとしてIntraWebがある。IntraWebを使用す ると、従来のGUIアプリケーションと同様にWebサーバ ーアプリケーションを構築することができる。本稿では IntraWeb 15を使用したWebアプリケーション開発の Tipsを紹介する。また実行結果は、MicrosoftのEdgeを 使用して確認する。

## 2.IntraWebでの新規プロジェクトの作成

### 2-1.IntraWeb 15のインストール

本題に入る前に、IntraWeb 15を使用できる開発環境を 整える必要がある。Delphi/400 10.2Tokyoを標準インス トールすると、IntraWebは古いバージョンのバンドル版 がインストールされている。IntraWeb 15を使用するに は、古いバンドル版をアンインストールした後、IntraWeb 15をインストールする必要がある。その手順を簡単にまと めておく。 ③ ②で取得したツール「IWBundledRemovalTool.exe」 を実行してバンドル版をアンインストールする。この時 「Delete IntraWeb package・・・」のチェックボックスは チェックしておく。 ④ ①で取得したIntraWeb 15をインストールする。

<手順>

 IntraWeb 15を次のサイトより取得する。 https://www.atozed.com/intraweb/download/v15/
 バンドル版をアンインストールするツールを次のサイト より取得する。

https://www.atozed.com/intraweb/bundled/remo val-tool/

### 2-2.新規プロジェクトの作成

過去のレポートでもIntraWebでの新規プロジェクトの作 新規プロジェクトの作成は「ファイル」→「新規作成」→「そ 成方法については掲載しているが、本稿では執筆時点での の他」の順に選択し、表示された新規作成画面で「Delphi 最新バージョンであるIntraWeb 15を例に解説するため プロジェクト」→「IntraWeb」→「IntraWeb Application 改めて記載する。 Wizard」をクリックする。【図1】

## 図1 IntraWeb新規プロジェクトの作成

77-	NUE I	411(E)	検索(5)	表示心	IJ,	ファクタ	リング(の)	プロジェクト(型)	実行(E)	コンポーネン
8	新規作成(N)		> E		VCL フォーム アプリケーション - Delphi(Y) マルチデバイス アプリケーション - Delphi(D) パッケージ - Delphi(E)			V iiD		
	パージョー開き直す	パージョン管理リボジトリから開く(2) 開き直す(B) トまき保存(S) Ctria		關(乙) Ctrl+S	,		VCLフォーム - Delphi( <u>L</u> ) マルチデバイスフォーム - Delphi( <u>P</u> ) コーット - Delphi(H)			
	名前を付けて保存(A) ブロシェクトに名前を付けて保存(E) すべて保存(V) Shift+Ctrl+C		杲存(E) hift+Ctrl+S			VCL フォーム アプリケーション - C++Builder(B) マルチデパイス アプリケーション - C++Builder(U) い パッケージ - C++Builder(U)				
	リアントを	ij Ijsod					VCL フォー マルチデバ	-ム - C++Builder (イスフォーム - C	r(E) ++Builder(	Ω
10	印刷(2)		<u>()</u>	Alt+F11		13 17	ユニット-・ その他(Q)	C++Builder(Z)		
×	終了(2)			contra			カスタマイン	<i>τ</i> ω		-

### 1.はじめに

2.IntraWebでの新規プロジェクトの作成

2-1.IntraWeb 15のインストール

2-2.新規プロジェクトの作成

3.明細形式のWeb画面を作成するテクニック

3-1.「TIWGrid」を使用した明細表示

3-2.明細表示のカスタマイズテクニック

4.CSVファイルのアップロード/ 取込を行うテクニック

4-1.CSVファイルのアップロード

4-2.CSVファイルの取込と 明細への反映方法

5.さいごに

016 MIGARO Technical Report

IntraWeb 15のライセンスキーを持っていない場合でも、 評価版として使用することができる。ライセンスキーが必 要な方は、ミガロ.営業担当まで問合せをしてほしい。 Delphi/400 畑中 侑

Delphi/400 福井和彦石山

Delphi/400 佐田 雄 —

围 S

а IF

rtPad4i 祐二

Vale 尾崎

w o

✓ C++Builder プロジェクト	₽検索		
ActiveX C Builder ファイル DataSnap Server IntraWeb RAD Server (EMS) Web サービス Web Broker Public Server (EMS) Delphi プロジェクト ActiveX Delphi プァイル IntraWeb RAD Server (EMS) Web サービス Web Broker Public Server (EMS) Web サービス Web Broker Public Server (EMS) Web サービス	IntraWeb Application Wizard	New DUnit Test Project	
	IntraWeb Applicat	ion Wizard	

表示された「IntraWeb Application Wizard」でWebアプリ ケーションの種類、プロジェクト名、ベースディレクトリを設 定する。【図2】

### **2** IntraWeb Application Wizard

IntraWeb Application Wizard		>
	IntraWeb Application	Wizard
•	Welcome to the IntraWeb Application Wizard application you want to create and any addit	I. Select the type of tional options
	Application Type	Options
U A	StandAlone Application (Indy)	Use <u>I</u> CL Stack Trace
	O StandAlone Application (Http.sys)	Pool Data Connections     Use ScaleMM2 Mem. Manager
	O ISAPI Extension	Use FastMM4 Mem. Manager
a	O IW Library	
inti	Project <u>N</u> ame Project 1	
	Base Directory	
	C:¥projects¥IntraWeb_Sample	Browse
	QK <u>C</u> ancel	

【図2】にあるWebアプリケーションの種類とオプションについて簡単に記載する。

<Webアプリケーションの種類>

① StandAlone Application (Indy)、
① StandAlone Application (Http.sys) (スタンドアローン)
WebサーバーもIntraWebが提供するモード。
② ISAPI Extension (アプリケーション)
WebサーバーはIISを用いて提供するモード。
③ IW Library (ASPX配置)
② と同様にWebサーバーはIISを用いるが、ASP.NETアプ リケーションとして配置可能なモード。

<オプションの種類> ① Use JCL Stack Trace JCL (JEDI Code Library)を使用してスタックトレースが可 能になる。 ② Pool Data Connections データベースへアクセスの都度、接続(コネクション)を確立 するのではなく、あらかじめ事前に一定数のコネクションを 確立しておき、それを使い回すことが出来るようになる。 ③ Use ScaleMM2 Mem.Manager、 Use FastMM4 Mem.Manager メモリリークを検出するためにScaleMM2、FastMM4を使 用することが可能になる。 【図2】の状態でOKボタンを押下すると新規プロジェクト
 が作成される。【図3】
 開発画面には、ServerController、Unit、UserSessionU
 nitの3つが自動で生成される。
 ServerControllerは、サーバー側の処理を扱うファイル
 で、例えばブラウザの戻るボタン押下時やアプリケーショ
 ン終了時の制御を行うといったことが出来る。
 Unitは実際に開発を行う画面で、このファイルに対してコンポーネントを貼り付けて開発を行う。
 UserSessionUnitはクライアントサーバー開発を行う時
 のDataModuleと同じ扱いになる。
 次の章では明細形式のWeb画面を作成するテクニックを
 紹介する。

ファイル 編む	L 検索	表示 リファ	クタリング	プロジェクト	実行	コンボーネント	ツール	りんこやり
1 1 - 8		000				E 22 8	ット Windows	~
構造		** 5	DINAL A	(-ジ  🔐 Unit1	UserSes	sionUnit 📝 Proje	t1 ドキュメン	hi i
🕀 🛅 Uses		2		•				
			: Epr	ogram Proje	ct1;			
			· US	es				
			1	FastMM4,				
				IWJcIStackT	race,			
			1	IWJclDebug,				
			1	Unit1 in 'U	nit1.pas	' [IWForm1:	TIWAppFor	m).
			10	ServerContr	oller in	ServerCont	roller.pa	s' [IWServ
			1	Usersession	Unit in	UserSession	Unit-pas'	[ImuserSe
			. {\$	R *.res}				
			- he	zin				
				TIWStart.Ex	ecute(Tr	ue);		
			Ler	10.				
+12-11 12-1	thb							
The second se	0	÷×.						
Miller Carrier T	Million Consider	•						
IWUserSession T	WUserSession							
WUserSession T	WUserSession							
IWUserSession T ゆ検索 プロパティーイベン	WUserSession							
NUSerSession T 検索 プロパティーイベン E LiveBinding デサ	WUserSession	7717						
NUSerSession T の検索 プロパティーイベン B LiveBinding デサ Name	WUserSession	ザイナ						
NUSerSession T の検索 ブロバティ イベン E LiveBinding デザ >> Name OldCreateOrder	WUserSession	ザイナ 1						
NUSerSession T の検索 ブロバティ イベン E LiveBinding デサ Name OldCreateOrder Tag	WUserSession K LiveBinding 77 IWUserSession □ False 0	ザイナ 						
NUserSession T ゆ 検索 プロパティ イベン E LiveBinding デサ Name OldCreateOrder Tag ClassGroup	WUserSession	ザイナ 1						
AJJAN TJAN IWUserSession TT )検索 「プロパティ イベン」 E LiveBinding デザ ン Name OldCreateOrder Tag ClassGroup	WUserSession	<del>9</del> 77						
AJJンドコンA WUserSession T )検索 ブロ(ティ イベン) E LiveBindingデザ Name OldCreateOrder Tag ClassGroup	WUserSession	<u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>						
AJ シンパーフバイ IWUserSession T り検索 E LiveBinding デザ > Name OldCreateOrder Tag ClassGroup	NUserSession	<u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>						
AJJンドコンA WUserSession T )検索 「フル(ティ イベン E LiveBinding デジ > Name OldCreateOrder Tag ClassGroup	WUserSession	<u>मिंग</u>						
NUSerSession TI WUserSession TI う 検索 う 大口/(ティーイベン) で LiveBinding デザ >> Name OldCreateOrder Tag ClassGroup	WUserSession	<u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>						
NUSerSession TI の検索 ブロパティ イベン で LiveBinding デザ Name OldCreateOrder Tag ClassGroup	WUserSession N I LiveBinding 77 MUserSession □ False 0	<u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>						
MUSerSession T の検索 プロパティイベン E LiveBinding デサ Name OldCreateOrder Tag ClassGroup	N CliveBinding 77 WUserSession D False 0	<u>भ</u> िने						



Delphi/400 俎中 侑 Delphi/400 福井 和彦 石山 智也 Delphi/400 佐田 雄一 SmartPad4i 國元祐二 Valence 尾崎 浩司



### 3.明細形式のWeb画面を作成するテクニック

この章では明細形式のWeb画面でよく見られる「件数が一 定数を超えた際に頁を分ける機能」や、「明細のタイトルをク リックしたら並び替えが行える機能」の実装方法について、 次の順番に従って説明を行う。

まず始めに本章以降で使用するプロジェクトを、プロジェク

ト名「IntraWebSample」として新規作成する。【図4】

3-1.「TIWGrid」を使用した明細表示

### 「TIWGrid」を使用した明細表示

- ② 明細表示のカスタマイズテクニック
- ・件数が一定数を超えた際に頁を分ける機能を実装
- 明細のタイトルをクリックしたら並び替えが行える機能 を実装

Veb Application Wizard			
	IntraWeb Application	Wizard	
•	Welcome to the IntraWeb Application Wizard application you want to create and any addit	I. Select the type of tional options	
<u> </u>	Application Type	Options	
<b>D</b>	• StandAlone Application (Indy)	Use <u>J</u> CL Stack Trace	
	O StandAlone Application (Http.sys)	Pool Data Connections     Use ScaleMM2 Mem. Manager	
	◯ ISAPI Extension	Use FastMM4 Mem. Manager	
a	O IW Library		
Ę		サンブルのプロジェクト名を 「IntraWebSample」とする	
<b>.</b>	Project <u>N</u> ame		
	Base <u>D</u> irectory 【	定 Browse	

作成されたプロジェクトのUnit1を開き、表示されたフォー
ムのNameプロパティに「frmList」と設定する。そして、【図
5】に従ってコンポーネントを配置し各プロパティを設定す
る。本稿では明細表示のコンポーネントとしてTIWGridを
使用する。配置と設定が終わったら、Unit1をファイル名
「ListFrm.pas」として名前を付けて保存を行う。



それでは、実行してブラウザにどのように表示されるか確認をしてみよう。実行方法は、メニューバーより「実行」→「実行」と選択するか、キーボードの「F9」を押下する。初回実行時、Windowsの警告が出る場合がある。

【図6】 警告が出たら「アクセスを許可する」をクリックして 先に進める。



実行すると始めに「IntraWeb Application」の画面が開く。 画面には各ブラウザのアイコンが並んでいる。その中から Edgeのアイコンをクリックする。そうするとEdgeが開き Web画面が表示される。【図7】

グラムでTClientDatasetへ作成して保持する。

【図8】に従ってTClientDatasetの各設定を行う。

UserSessionUnitを開き、画面にTClientDatasetを配置

する。Nameプロパティは「cdsSample」と設定する。続いて



🗑 IntraWeb Application — 🗆 🗙	
Ele Settings Jools Spipt Remeter Elegeのアイコンをクリック Server Log	
Ny transliteb Application (Driv) ~ x86) Registrated to Scalar O Sumary (Arrow 1997) Company: Migato Ca., Lid Developer Name: Kould O Sumary SN 424F-450T47-SIVED 3440 Free updates unit; 7 13, 2022 Editors: Unaview Unimate Edition	
License Number: 2021 Build Date: 7 25, 2021 IntraWeb Version: 15 🔲 🔮 127.0.0.1:8888 🗙 🗙 🕂	
32481 Server on 644 HTTP Server is astive $\leftarrow \rightarrow \mathbb{C}$ () 127.0.0.1:8888	îð
明細形式の表示	
明細表示	

この時点では明細にはまだ何も表示はされない。実行結果 が確認できたら、Edgeの「×」ボタンをクリックしてブラウザ を閉じ、IntraWeb Application画面も同様に「×」ボタンを クリックして終了し開発画面へ戻る。

続いて、明細に表示するデータの準備を行う。データはプロ

### 図 8 TClientDatasetの設定



022 MIGARO Technical Report

次に、UserSessionUnitのOnCreateイベントで cdsSampleのCreateDataSetメソッドを呼び出し、メモリ 上にデータセットを作成する。【ソース1】

### (ソース1)

### UserSessionUnitのOnCreateイベント

procedure TIWUserSession.IWUserSessionBaseCreate(Sender: TObject); begin // データセットを作成 cdsSample.CreateDataSet; end;

そして、データ作成の処理としてUserSessionUnitの Publicのプロシージャ「SampleDataCreate」を作成す る。実装部は【ソース2】に従って記述する。この処理で100 件のデータを作成する。UserSessionUnitの開発は以上 となる。

### ソース 2

UserSessionUnitのSampleDataCreateプロシー
<宣言部> public { Public declarations } procedure SampleDataCreate; // サンプルデータ作成 end; <実装部> procedure INWUcerSection SampleDataCreate;
var i: Integer; begin // cdsSampleをクリア cdsSample.EmptyDataSet;
for i := 1 to 100 do begin cdsSample.Append; cdsSampleSYCD.AsString := FormatFloat('000000', i cdsSampleSYNM.AsString := 'サンプル商品 (' + Forma cdsSampleSURY.AsInteger := i; cdsSampleTANK.AsInteger := i * 1000; cdsSampleKNGK.AsCurrency := i * (i * 1000); cdsSample.Post; end; end;



ここからは、frmListの明細へcdsSampleのデータを表示 する処理を実装していく。frmListを開き、メニューバーより 「ファイル」→「使用するユニット」を選び、ServerController を選択する。次にfrmListのPrivateプロシージャとして

「GridClear」「GridLayout」「ListDataSet」の3つを宣言す る。【ソース3】各プロシージャの実装部は【ソース4~7】に 従って記述する。

## ソース 3

frmListのPrivateプロシ	ージャ宣言
private { Private 宣言 } procedure GridClear; procedure GridLayout; procedure ListDataSet;	// 明細初期化 // 明細の項目幅調整 // 明細データセット



frmListのGridClearブロシージャ
procedure TfrmList.GridClear;
begin with gallist do
begin
Clear;
// 行数初期化
RowCount := I;
// タイトル設定
Cell[0, 0].Text := '商品コード';
Cell[0, 1].Text := '商品名';
Cell[0, 2].Text := '数量';
Lell[U, 3].lext := 単恤; Call[O4] Tayt := '今頗':
Cerrico, 4].Text 五領 ,
// 列表題Alignment
Cell[O, O].Alignment := taCenter;
Cell[O, 1].Alignment := taCenter;
Cell[U, 2].Alignment := talenter;
Cell[0, 3].Alignment := taCenter;
end;
end;

ソース 5

frmListのGridLayoutプロシージャ
procedure TfrmList.GridLayout; begin with gdList do begin // 列幅の初期設定 Cell[0, 0].Width := '100'; // [00] 商品コード Cell[0, 1].Width := '300'; // [01] 商品名 Cell[0, 2].Width := '100'; // [02] 数量 Cell[0, 3].Width := '170'; // [03] 単価 Cell[0, 4].Width := '180'; // [04] 金額
// 高さの初期設定 Cell[0, 0].Height := '35'; end; end;



続いてfrmListのOnCreateイベントに、【ソース8】に従って 明細の初期表示の処理を実装する。そして「明細表示」ボタン (btnListView)のOnClickイベントには、【ソース9】のように ここまで設定してきた各プロシージャの実行を記述する。



frmListのOnCreateイベント
procedure TfrmList.IWAppFormCreate(Sender: TObject); begin // 明細初期設定 gdList.RowCount := 1; gdList.ColumnCount := 5;
// 明細初期化 GridClear; 【ソース4】参照
// 明細の項目幅調整 GridLayout;【ソース5】参照 end;

# ソース9

frmList.btnListViewのOnClickイベント
<pre>procedure TfrmList.btnListViewClick(Sender: TObject);</pre>
begin // an/mtattal/
// 明釉初期化
GridClear;
// サンプルデータ作成
UserSession.SampleDataCreate;
// 明細データセット
ListDataSet;
end:

それでは実行して確認してみよう。今回は【図7】の時とは 違い明細のタイトルが表示されている。続いて画面の「明 細表示」ボタンをクリックすると、明細にデータが表示され る。しかし現時点では、1画面に全件を明細表示しただけと なる。【図9】

# 図9 実行結果イメージ(明細表示)



# Delphi/

12	27.0.0.1:88	888		×	+						-		>	<
	С	()	127.0.0.1:88	88		îô	۲	G	Ę	= (	è	۲		·
σ	)表示													^
														_
T	ŧ													I
7			商品名			数量		単価			金額	A	٦	
7	サンプル	レ商品	<u>商品名</u> (000010)			<b>数量</b>		単価 1	,000		金額	<b>A</b>	00	
7	サンプル	レ商品	<b>商品名</b> (000010) (000020)			数量 1 2		<b>単価</b> 1 2	,000		金額	A 1,00 4,00	00	
7	サンプルサンプル	レ商品レ商品	商品名 (000010) (000020) (000030)			数量 1 2 3		<b>単価</b> 1 2 3	,000		金額	<b>A</b> 1,00 4,00 9,00	00	
	サンプル サンプル サンプル サンプル	レ商品レ商品レ商品	商品名 (000010) (000020) (000030) (000040)			数量 1 2 3 4		単価 1 2 3 4	,000		金額	<b>ğ</b> 1,00 4,00 9,00 16,00	00 00 00	
	サンプル サンプル サンプル サンプル	レ商品レートの商品レートの商品	商品名 (000010) (000020) (000030) (000040) (000050)			数量 1 2 3 4 5		単価 1 2 3 4 5	,000,000,000		金精	1,00 4,00 9,00 16,00 25,00	00 00 00 00	
	サンプバ サンプバ サンプバ サンプバ サンプバ	レ商商品品の高品のである。	商品名 (000010) (000020) (000030) (000040) (000050) (000060)			数量 1 2 3 4 5 6		<b>単価</b> 1 2 3 4 5 6	,000 ,000 ,000 ,000		金額	1,00 4,00 9,00 16,00 25,00 36,00	000000000000000000000000000000000000000	

サンプル商品 (000040)	4	4,000	16,000
サンブル商品 (000050)	5	5,000	25,000
サンプル商品 (000060)	6	6,000	36,000
サンブル商品 (000070)	7	7,000	49,000
サンプル商品(000080)	8	8,000	64,000
サンプル商品(000090)	9	9,000	81,000
サンプル商品 (000100)	10	10,000	100,000
サンプル商品 (000110)	11	11,000	121,000
サンブル商品 (000120)	12	12,000	144,000
サンプル商品 (000130)	13	13,000	169,000
サンブル商品 (000140)	14	14,000	196,000
サンプル商品 (000150)	15	15,000	225,000
サンプル商品 (000160)	16	16,000	256,000
サンブル商品 (000170)	17	17,000	289,000
Ash min man (asses)			



Delphi/400 俎中 侑



### 3-2.明細表示のカスタマイズテクニック

この節では前節で作成した明細表示の画面に対して、機能 追加の実装を進めていく。実装には外部ライブラリを使用す る。IntraWebにおいても、画面をカスタマイズするために外 部ライブラリを使用することが可能である。前節で画面に配 置したTIWGridは、HTMLにはtableタグとして出力され る。そのtableタグの表示内容をカスタマイズできる、 「DataTables」という外部のjQueryプラグインライブラリを 使用して機能追加の実装を行う。

(公式サイト:https://www.datatables.net/)

まず始めにfrmListのOnCreateイベントへ【ソース10】に 従って追記を行う。ここではCDN (Content Delivery Network)にて公開されている、jquery DataTablesのスタ

### ソース 10

イルシートとJavascriptの参照設定を追記する。参照先は URLで記述しており、その中にバージョン情報も含まれてい る。最新のバージョンについては、https://cdn.datatables. netを参照いただきたい。

続いてfrmListのOnRenderイベントへスクリプトの追加を 記述する。【ソース11】

記述した中に有る"#TBLGDLIST"の"GDLIST"は、画面の TIWGridのNameプロパティの設定値を大文字にしたもの となる。もし【図5】の設定時にTIWGridのNameプロパティ を異なる設定にしている場合は、その設定に合わせていただ きたい。



### ソース 11

### frmListのOnRenderイベント

procedure TfrmList.IWAppFormRender(Sender: TObject); begin // HTMLスクリプトを追加 AddToInitProc('\$("#TBLGDLIST").DataTable();'); end;

「Previous」「1···10」「Next」が表示されている。これは、 それでは実行して確認してみよう。Web画面が開くと【図 9】の時点から更に、いくつか項目や表記が追加されてお 10件毎に10頁に分かれている各頁間を移動する機能に り、明細のタイトル部にも▲▼が表示されている。ただ全て なっている。それぞれクリックすると明細の内容が変わっ 英語表記となっている。明細を表示して追加された各機能 ているのと、「Showing (X) to (Y) of 100 entries」の を見ていこう。明細を表示すると最初の10件のみが明細に (X)と(Y)の値が変わっているのが分かる。これで件数が 表示される。明細の下には「Showing 1 to 10 of 100 一定数を超えた際に頁を分ける機能が実装されたことに entries」と表記されており、100件中の1件目から10件目 なる。【図10】

件数が一定数を超えた際に頁を分ける機能 図 10 127.0.0.1:8888 × + ÷ ← → ♂ ① 127.0.0.1:8888 ය 🤌 ଓ 👍 🏽 😩 明細 明細形式の表示 最初の10件のみが表示されている 明細表示 ow 10 👻 entri 商品コード 商品名 数量 単価 金額 000010 サンプル商品 (000010) 1,000 1.000 1 サンプル商品(000020) 2 2,000 4,000 000020 サンプル商品 (000030) 9,000 000030 3 3,000 000040 サンプル商品(000040) 4 4.000 16.000 サンプル商品 (000050) 000050 5 5,000 25,000 000060 サンプル商品(000060) 6 6.000 36,000 49,000 000070 シブル意見 000080 100件中の1件目から10件目迄を 64,000 表示していることを表している 000090 81.000 サンプル商品(000100) 000100 100,000 Showing 1 to 10 of 1 2 3 10 Nex 10件毎に10頁に分かれている

迄を表示していることを表している。更にその下には、

028 MIGARO Technical Report

127.0.0	1:8888 × +			-
⇒ c	127.0.0.1:8888		ි 🤣	3 1≏ @
副形式の表	<del>.</del>			
明細表示	]			
w 10 👻 er	tries	s	earch:	
5品コード ⇔	商品名 ▲	数量 🔶	単価 🔶	金額 ≑
000110	サンプル商品 (000110)	11	11,000	121,000
000120	サンブル商品(000120)	12	12,000	144,000
000130	サンプル商品 (000130)	13	13,000	169,000
000140		14	14,000	196,000
000150	明細の内容が変わる	5 15	15,000	225,000
000160	サンブル商品 00160)	16	16,000	256,000
000170	サンプル商品 (000170)	17	17,000	289,000
000180	サンブル商品 (000180)	18	18,000	324,000
000190	サンブル 表記が変材	っる	19,000	361,000
000200	サンブルまの		20,000	400,000
	Showing 11 to 20	) of 100 entri	25	
	Previous 1 2 3	4 5 …	10 Next	
	2頁目をクリッ	クする		



次に明細のタイトルをクリックしてみると、クリックした項目 の昇順で明細が並び変わる。しかも頁内ではなく、100件全 体での並び替えとなっている。更に同じ項目をクリックする と今度は降順に切り替わる。これで明細のタイトルをクリッ クしたら並び替えが行える機能も実装されたことになる。 【図11】

明細の上に追加された項目についても見てみる。左上のコン ボBOX「Show」は1頁当りの件数を示している。初期値は

### 明細の並び替え機能

127.0.0.1:8888



「10」でコンボBOXリストには既定の値がセットされている。

コンボBOXの値を変える毎に、選択した値の件数が1頁当り

また右上の「Search」は検索項目になる。Search項目に

の件数となって、明細の表示内容が変わる。

### ← → C (i) 127.0.0.1:8888 明細形式の表示 明細表示 how 10 🗸 entrie 商品コード 商品名 000010 サンプル商品 (000010) 000020 サンプル商品 (000020) 000030 サンプル商品 (000030) 3,000 9,000 000980 サンプル商品 (000980) 98 98,000 9,604,000 000040 サンプル商品(000040) 4,000 16,000 000970 サンプル商品(000970) 97 97,000 9,409,000 000050 サンプル商品(000050) 25,000 9,216,000 5,000 000960 サンプル商品 (000960) 96 96,000 000060 サンプル商品(000060) 6.000 36.000 000950 サンプル意品(000950) 9.025.000 95 95.000 000070 サンプル商品 (000070) 7.000 49.000 000940 サンプル商品 (000940) 94 94.000 8.836.000 000080 サンプル商品(000080) 8,000 64,000 000930 サンプル商品 (000930) 93.000 8,649,000 93 サンプル商品 (000090) 81,000 000920 サンプル商品 (000920) 8,464,000 9,000 92,000 100,000 サンプル商品 (000910) 8,281,000 000100 サンプル商品 (000100) 10,000 000910 91,000 91 Showing 1 to 10 of 100 entries Showing 1 to 10 of 100 entries Previous 1 2 3 4 5 ... 10 Next Previous 1 2 3 4 5 ... 10 Next

### 図 12 追加機能「Show」「Search」



ここまでの機能を僅か3行のソースコードを追加するだけ この節の最後に、追加された表記が英語になっているもの で実現することができる。また、ShowコンボBOXや を日本語に変更する方法を説明する。変更箇所はfrmList Search項目の非表示や、1頁当りの件数の初期値を変更 のOnRenderイベントで、ソースコードを【ソース13】に従 することも可能である。frmListのOnRenderイベントのソ って変更することで日本語表記に変わる。ここでもCDNに ースコードを変更することで実装できる。詳しくは【ソース て公開されているJSON(JavaScript Object Notation) 12】を参照し、実装して結果を確認してもらいたい。 を参照する。

## ソース 12

### frmListのOnRenderイベント(変更) procedure TfrmList.IWAppFormRender(Sender: TObject); var sScriptStr: string; begin // HTMLスクリプトを追加 // AddToInitProc('\$("#TBLGDLIST").DataTable();'); sScriptStr := '\$("#TBLGDLIST").DataTable({'; sScriptStr := sScriptStr + 'displayLength: 20'; sScriptStr := sScriptStr + ', lengthChange: false ';< sScriptStr := sScriptStr + ', searching: false '; sScriptStr := sScriptStr + '});'; //HTMLスクリプトを追加 AddToInitProc(sScriptStr); end;

### ソース 13

procedure TfrmList.IWAppFormRender(Sender: TObject); var sScriptStr: string; begin // HTMLスクリプトを追加 // AddToInitProc('\$("#TBLGDLIST").DataTable();');
sScriptStr := '\$.extend( \$.fn.dataTable.defaults, ['; sScriptStr := sScriptStr + 'language: ['; sScriptStr := sScriptStr + 'url: "http://cdn.datatables.net/plug- sScriptStr := sScriptStr + ']'; sScriptStr := sScriptStr + ']; sScriptStr := sScriptStr + '\$("#TBLGDLIST").DataTable(); //HTMLスクリプトを追加 AddToInitProc(sScriptStr); end;

[displayLength:20] 1頁当りの件数の初期値を設定 「lengthChange: false」 ShowコンボBOXを非表示にする [searching: false] Search項目を非表示にする



Delnhi/400	Delnhi/400	Delphi/400	SmartPad4i	Valence
Jeipni/4uu	Jeipni/400	Deipni/400	SmartPad4I	Valence
田中 宿	福井和彦 石山智也	佐田 雄一	國元 祐二	第二部 目

実行して確認してみよう。【図10】では英語表記になっていた 箇所が日本語表記に変わっている。【図13】

以上で「明細形式のWeb画面を作成するテクニック」の説明 は終了となる。次の章ではCSVファイルを取扱うテクニック を紹介する。

## 図 13 日本語表記対応



### 4. CSVファイルのアップロード/ 取込を行うテクニック

この章では「CSVファイルをアップロード」し「アップロード したCSVファイルのデータを取り込み画面明細へ表示する| 方法について説明を行う。

### 4-1.CSVファイルのアップロード

最初はCSVファイルをアップロードする機能を実装してい く。新しくフォームを追加して実装するため、メニューバーよ り「ファイル」→「新規作成」→「その他」の順に選択し、表示 された新規作成画面で「Delphiプロジェクト」→ 「IntraWeb |→「New Form |をクリックする。【図14】 新しく追加された「Unit1」のNameプロパティを 「frmCSVUpload」と設定する。【図15】に従ってコンポーネ ントを配置し各プロパティを設定する。この時 「TIWFileUploader」のTextStringsプロパティの設定も合 わせて行う。【図16】

この画面は「TIWFileUploader」を使用してアップロードし たCSVファイルを読み込み、そのデータを「TIWGrid」に表 示をする。そして「更新」ボタンをクリックしたら、前章で設定 したUserSessionUnitのcdsSampleへ更新を行い、当画面 を閉じてfrmListの明細へ反映させる動きとなる。 配置と設定が終わったら、Unit1をファイル名 「CSVUploadFrm.pas」として「名前を付けて保存」を行う。





### 図 16 TIWFileUploaderのTextStringsプロパティの設定

T	extStrings	(TIWFileUploaderTextStrings)
	CancelButtonText	キャンセル
	DragText	ここにファイルをドロップしてください。
	EmptyErrorText	{file}は空です。ファイルを再度選択してください。
	MinSizeErrorText	{file} is too small, minimum file size is {minSizeLimit
	MultipleFileDropNotAllowedText	ドロップできるのは1つのファイルだけです。
	NoFilesErrorText	アップロードするファイルがありません。
	OfTotalText	of
	OnLeaveWarningText	ファイルがアップロードされていますが、今暮れるとアッ
	RemoveButtonText	除去
	SizeErrorText	{file} is too large, maximum file size is {sizeLimit}.
	TypeErrorText	{file}の拡張子が正しくありません。{extensions}ファー
	UploadButtonText	ファイルアップロード
	UploadErrorText	アップロードエラー
Ĩ		

1	<b>(</b>	<b>E</b>	٢	1
2b ti	New DUnit Test Case	New DUnit Test Project	New Form	ч
me				1
				1
				1
				1
				-1
				1
		ОК	キャンセル	^







始めに、「TIWFileUploader」を使用してアップロードした CSVファイルを読み込み、そのデータを「TIWGrid」に表示 するところまでを実装していく。前章でfrmListに実装した のと同様に、frmCSVUploadのPrivateプロシージャとして 「GridClear」「GridLayout」の2つを宣言する。合わせてプ ライベート変数「FUploadFile」もString型で宣言しておく。 【ソース14】各プロシージャの実装部については、GridCle arは【ソース4】、GridLayoutは【ソース5】と同様の処理を実 装する。

## ソース 14

### frmCSVUploadのPrivateプロシージャと変数の宣言

private

{ Private 宣言 } FUploadFile: String;

procedure GridClear; procedure GridLayout;

// 明細初期化 // 明細の項目幅調整

// CSVファイル保管場所

そしてfrmCSVUploadのOnCreateイベントにも、前章で frmListに実装したのと同様に明細の初期表示の処理を実 装する。【ソース8】

続いて、TIWFileUploaderの各イベントに処理を実装して いく。まずはOnAsyncUploadCompletedイベントに【ソー ス15】に従って処理を実装する。ここでは、CSVファイルの アップロードが終わった時点で、アップロードしたCSVファ イルのフルパスをFUploadFileに退避する。 次にOnAsyncUploadSuccessイベントに【ソース16~17】 に従って処理を実装する。ここでは、OnAsyncUploadCom pletedイベントで取得したCSVファイルのフルパス: FUploadFileを使用して、アップロードしたCSVファイルを 開き、画面明細へ展開する処理となっている。

## ソース 15

### frmCSVUpload.fuCSVUploadのOnAsyncUploadCompletedイベント

procedure TfrmCSVUpload.fuCSVUploadAsyncUploadCompleted(Sender: TObject; var DestPath, FileName: string; var SaveFile, Overwrite: Boolean); begin

//アップロードしたファイルのパスとファイル名を取得 FUploadFile := DestPath + FileName; end; ソース 16

# frmCSVUpload.fuCSVUploadのOnAsyncUplo

```
procedure TfrmCSVUpload.fuCSVUploadAsyncUploadSuccess(Sen
 EventParams: TStringList);
var
 i: Integer;
 sStrList: TStringList;
 sRowList: TStringList;
begin
 // 明細初期化
 GridClear;
 //CSVデータを保持するリストの作成
 sStrList := TStringList.Create;
 trv
   //行データを保持するリストの作成
   sRowList := TStringList.Create;
   try
     sStrList.LoadFromFile(FUploadFile);
     //先頭行は、タイトルの為無視して、2行目から取得
     for i := 1 to sStrList.Count - 1 do
     begin
      //行のテキスト(カンマ区切り文字列)をsRowListにt
      sRowList.Clear;
       sRowList.CommaText := sStrList[i];
      //sRowListの要素より明細へ値をセット
       gdList.RowCount := gdList.RowCount + 1;
                                           レー
```

### ソース 17

### frmCSVUpload.fuCSVUpload@OnAsyncUplo // Alignmentの設定 gdList.Cell[gdList.RowCount - 1, 0].Alignment := gdList.Cell[gdList.RowCount - 1, 1].Alignment := gdList.Cell[gdList.RowCount - 1, 2].Alignment := gdList.Cell[gdList.RowCount - 1, 3].Alignment := gdList.Cell[gdList.RowCount - 1, 4].Alignment := // サンブルデータ gdList.Cell[gdList.RowCount - 1, 0].Text := sRowL gdList.Cell[gdList.RowCount - 1, 1].Text := sRowL gdList.Cell[gdList.RowCount - 1, 2].Text := Forma gdList.Cell[gdList.RowCount - 1, 3].Text := Forma gdList.Cell[gdList.RowCount - 1, 4].Text := Forma end; finally sRowList.Free; end; finally sStrList.Free; end; // 明細の項目幅調整 GridLayout; end;

adSuccessイベント	Delphi/400 屆中
der: TObject;	
	Delphi/400 福井 和彦 石山 智也
zyh	Delphi/400 佐田 雄一
ス17]へ続く	SmartPad4i 國元 祐二
oadSuccessイベント(続き)	
<pre>taCenter; taLeftJustify; taRightJustify; taRightJustify; taRightJustify; ist[0]; ist[1]; tFloat('#,0', StrToIntDef(sRowList[2], 0)); tFloat('#,0', StrToIntDef(sRowList[3], 0)); tFloat('#,0', StrToCurrDef(sRowList[4], 0));</pre>	Valence 尾崎 浩司

ここまで実装できたら、frmListからfrmCSVUploadを呼 び出すように、frmListへ処理を追加する。frmListを開き 【図17】に従ってTIWButtonを追加する。そして追加した 「CSVデータ取込」ボタン(btnCSVUpload)のOnClickイ ベントには【ソース18】に従って処理を実装する。合わせて frmListのuses節に「CSVUploadFrm」を追加し、プライ ベート変数「frmCSVUpload」の宣言を行う。

### frmListに「CSVデータ取込ボタン」を追加 図 17



### ソース 18

### frmListのbtnCSVUploadのOnClickイベント <usesに「CSVUploadFrm」を追加> uses Classes, SysUtils, IWAppForm, IWApplication, IWColor, IWTypes, IWVCLBaseControl, IWBaseControl, IWBaseHTMLControl, IWControl, IWCompLabel, Vcl.Controls, Vcl.Forms, IWVCLBaseContainer, IWContainer, IWHTMLContainer, IWHTML40Container, IWRegion, IWCompGrids, IWCompButton, CSVUploadFrm; <private変数を宣言> private 宮글 ] frmCSVUpload: TfrmCSVUpload; procedure GridClear; // 明細初期化 procedure GridLayout; // 明細の項目幅調整 procedure ListDataSet; // 明細データセット <OnClickイベント> procedure TfrmList.btnCSVUploadClick(Sender: TObject); begin frmCSVUpload := TfrmCSVUpload.Create(WebApplication); frmCSVUpload.Show; end;

実行する前にアップロード用のCSVファイルを準備して たCSVファイルのデータが確認できる。【図20】 おく。CSVファイルのデータイメージは【図18】を参照いた では、CSVファイル以外をアップロードしたらどうなるの か確認してみよう。PDFファイルを用意しアップロードを だきたい。それでは実行して確認してみよう。今節で追加 したfrmListの「CSVデータ取込 |ボタンをクリックする 行うと、【図21】のようにエラーが発生する。これは、【図 と、frmCSVUploadへ遷移する。画面上部には背景色が **15**]でTIWFileUploaderのAllowedExtensionsプロパ ティに"CSV"を設定したことで、拡張子を制限しているた 赤色の「ファイルアップロード」ボタンが表示されている。 CSVファイルをアップロードする方法は以下の2通りがあ めである。 る。【図19】 TIWFileUploaderは先程の2通りのアップロードの方法 ① [ファイルアップロード]ボタンをクリックして、ファイル を既に実装しており、またWebサーバーで実行した場合 選択。ダイアログを開き、ダイアログよりCSVファイルを選 には、アップロードしたファイルをWebサーバー内のフォ ルダに保管する機能も実装している。そのため 択する。 ②CSVファイルを直接「ファイルアップロード」ボタンヘド TIWFileUploaderを使用することで、Webアプリケー ラッグ&ドロップする。 ションでのファイルアップロード機能を簡単に実装するこ どちらかの方法でCSVファイルのアップロードを行う。そ とが可能となる。

うすると画面の明細にデータが表示され、アップロードし

X	18	CSV	/ファイ	ルの	データ	イメー	ジ
	1		.110		210		310.
1	商品	1-1	、, 商品	品名,	数量,	単価,	金額
2	<i>"</i> 100	010″,	″サン	プル	·商品	(1000	)10)
3	<i>"</i> 100	020″,	″サン	プル	商品	(1000	)20)
4	<i>"</i> 100	030″,	″サン	プル	商品	(1000	)30)
5	<i>"</i> 100	040″,	″サン	プル	商品	(1000	)40)
6	<i>"</i> 100	050″,	″サン	プル	商品	(1000	)50)
7	<i>"</i> 100	060″,	″サン	プル	商品	(1000	)60)
8	<i>"</i> 100	070″,	″サン	プル	·商品	(1000	)70)
9	<i>"</i> 100	080″,	″サン	プル	·商品	(1000	(080
10	<i>"</i> 100	090″,	″サン	プル	商品	(1000	)90)
11	<i>"</i> 100	100″,	″サン	プル	商品	(1001	00)
12	<i>"</i> 100	)110″,	″サン	プル	商品	(1001	10)
13	<i>"</i> 100	120″,	″サン	プル	商品	(1001	20)
14	<i>"</i> 100	130″,	″サン	プル	商品	(1001	30)
15	<i>"</i> 100	140″,	″サン	プル	·商品	(1001	40)
16	<i>"</i> 100	150″,	″サン	プル	商品	(1001	50)

. . . . . . . . . . . . . . . . . . 90, 1000, 90000 ″, 91, 1000, 91000 , 92, 1000, 92000 ″. 93. 1000. 93000 . 94, 1000, 94000 ". 95, 1000, 95000<sub>4</sub> , 96, 1000, 96000 ". 97, 1000, 97000 98, 1000, 98000 ″, 99, 1000, 99000↓ , 80, 1000, 80000 ″. 81. 1000. 81000 , 82, 1000, 82000 . 83, 1000, 83000 , 84, 1000, 84000



### 図 19 実行結果イメージ(CSVアップロード)





j 🔮 12	7.0.0.1:8888 × +					-		×
$\leftarrow \rightarrow$	C (i) 127.0.0.1:8888		ි 🤣	G	£≣	œ	۲	
csvデー	夕取込							
ファイル	レアップロード							
🗸 UploadSar	nple.csv 0.8kB							
Was						=	2	
更初						<i>1</i> , <del>2</del> ,1	9	
商品コード	商品名	数量	単価	Γ	金額	I.		
100010	サンプル商品 (100010)	90	1,000			90,0	00	
100020	サンプル商品(100020)	91	1,000	0		91,0	00	
100030	サンプル商品(100030)	92	1,000			92,0	00	
100040	サンプル商品(100040)	93	1,000	D		93,0	00	
100050	サンプル商品 (100050)	94	1,000	D		94,0	00	
100060	サンプル商品 (100060)	95	1,000	D		95,0	00	
100070	サンプル商品 (100070)	96	1,000	0		96,0	00	
100080	サンプル商品 (100080)	97	1,000	D		97,0	00	
100090	サンプル商品 (100090)	98	1,000	0		98,0	00	
100100	サンプル商品 (100100)	99	1,000	D		99,0	00	
100110	サンブル商品 (100110)	80	1,000	D		80,0	00	
100120	サンプル商品(100120)	81	1,000	D		81,0	00	
100130	サンブル商品 (100130)	82	1,000	D		82,0	00	
	サンプル商品 (100140)	83	1,000	D		83,0	00	
100140				-				

## 図 21 CSVファイル以外をアップロードした場合



### 4-2.CSVファイルの取込と明細への反映方法

前節では、CSVファイルをアップロードしてデータを画面
 可細に取り込むところを実装してきた。この節ではアップ
 ロードしたデータを、「更新」ボタンをクリックすることで
 UserSessionUnitのcdsSampleへ更新し、当画面を閉じ
 てfrmListの明細へ反映させる機能を実装していく。
 始めにUserSessionUnitのcdsSampleへ更新し、当画面を閉じ
 てfrmListの明細へ反映させる機能を実装していく。
 いるの明細へ反映させる機能を実装していく。
 いるの明細へ反映させる機能を実装していく。

### ソース 19

UserSessionUnitのプライベート変数とProperty
private [Private declarations] FDataDspMode: Boolean;
public { Public declarations } procedure SampleDataCreate; // サンブルデータ作成
// 「明細形式の表示」のデータ表示モード (True:データ表示) property DataDspMode: Boolean read FDataDspMode write FDataDspM
énd,

次にfrmCSVUploadを開き、メニューバーより「ファイル」 →「使用するユニット」を選び、ServerControllerを選択 する。そして、「更新」ボタン(btnSave)のOnClickイベント に【ソース20~21】に従って処理を記述する。ここでは、最 初にcdsSampleのデータを全て削除した後で、CSVファ イルのデータをcdsSampleへ追加している。その後で UserSessionUnitのDataDspModeプロパティを 「True」にしている。DataDspModeプロパティに設定した 値は、frmListに戻った際に使用する。最後に 「Self.Release」で画面を閉じている。 また、「戻る」ボタン(btnClose)のOnClickイベントにも 【ソース22】に従って処理を記述する。frmCSVUploadに 対する機能の実装は以上となる。

### ソース 20

frmCSVUpload.btnSaveのOnClickイベント
procedure TfrmCSVUpload.btnSaveClick(Sender: TObject); var i: Integer; sStrList: TStringList; sRowList: TStringList; begin // UserSessionのcdsSampleをクリア UserSession.cdsSample.EmptyDataSet;
//CSVデータを保持するリストの作成 sStrList := TStringList.Create; try //行データを保持するリストの作成 sRowList := TStringList.Create; try
sStrList.LoadFromFile(FUploadFile); //先頭行は、タイトルの為無視して、2行目から取得 for i := 1 to sStrList.Count - 1 do begin //行のテキスト (カンマ区切り文字列) をsRowListにセット sRowList.Clear; sRowList.CommaText := sStrList[i];
【ソース21】へ続く





Delphi/400 俎中 侑 Delphi/400 福井 和彦 石山 智也 Delphi/400 佐田 雄一 SmartPad4i 國元 祐二 Valence 尾崎 浩司

### ソース 21

frmCSVUpload.btnSaveのOnClickイベント(続き)			
// サンブルデータを追加 UserSession.cdsSample.Append; UserSession.cdsSampleSVD.AsString UserSession.cdsSampleSVTM.AsString UserSession.cdsSampleSVTM.AsInteger UserSession.cdsSampleTANK.AsInteger UserSession.cdsSampleKNGK.AsOurrency UserSession.cdsSample.Post; end; finally sRowList.Free; end; finally sStrList.Free; end;	:= sRowList[0]; := sRowList[1]; := StrToIntDef(sRowList[2], 0); := StrToIntDef(sRowList[3], 0); := StrToCurrDef(sRowList[4], 0);	// 商品コード // 商品名 // 数量 // 金額	
// データ表示モードセット UserSession.DataDspMode := True;			
// 閉じる Self.Release; end;			



frmCSVUpload.btnCloseのOnClickイベント procedure TfrmCSVUpload.btnCloseClick(Sender: TObject); begin // 閉じる Self.Release; end;

続いてfrmListを開き、OnRenderイベントを【ソース23】に 従って変更する。ここではUserSessionUnitの DataDspModeプロパティが「True」の場合に、明細をクリ アしてからcdsSampleのデータを明細に表示している。また OnRenderイベントはfrmCSVUploadから戻ってきた際に も発生するため、frmCSVUploadの更新ボタンをクリック すれば、DataDspModeプロパティが「True」になっている ため、CSVファイルのデータで洗い替えられたcdsSample のデータがfrmListの明細に表示されることになる。



frmListのOnRenderイベント(変更)		
procedure Tfrmlist.IWAppFormRender(Sender: TObject):		
var		
sScriptStr: string:		
begin		
sScriptStr := '\$ extend( \$ fp.dataTable.defaults. [':		
sScriptStr := sScriptStr + 'language: [':		
sscriptStr := sscriptStr		
+ 'url: "http://cdp.datatables.net/plug-ins/9dcbecd42ad/i18p/Japanese.ison":		
s&criptStr := s&criptStr + ']':		
scriptStr := scriptStr + '):':		
sScriptStr := sScriptStr + '\$("#TBLGDLIST").DataTable();';		
//11版人ソウノアドで通知/		
Addioinitr rocissor iptotr);		
// CSVデータ取込両面から戻ってきたときに明細を再表示する		
if (learSection DataDorMode) then		
herin		
UserSession DataDonMode := False:		
oscioeston.batabspmode - Tarse,		
// 8日4四477首日/ビ		
77 Shauthata		
// 旧細データセット		
and		
end'		
chu,		

040 MIGARO Technical Report

それでは実行して確認してみよう。frmCSVUploadで 以上で「CSVファイルのアップロード/取込を行うテク CSVファイルをアップロードした後「更新」ボタンをクリッ ニック」の説明は終了となる。

クすると、frmListの明細にアップロードしたCSVファイル のデータが表示されることが確認できる。【図22】



### 5.さいごに

本稿ではIntraWebを使用したWeb開発のTipsとして、 Web画面での明細形式の表示を行う際のテクニックと、 CSVファイルをアップロードして取り込み画面に表示する 方法について紹介した。これらの機能は、コンポーネント の設定と、CDNで公開されているライブラリの利用、そし て少しのソースコードを実装することで実現することがで きた。また、本稿を執筆するにあたり参考にした、 IntraWebの開発元であるAtozed社が公開しているデモ プログラムには、IntraWebの活用方法が他にも沢山紹介 されている。こちらも是非参考にしていただきたい。 (URL:https://github.com/Atozed/IntraWeb) 本稿がIntraWebを使用してWeb開発を行う際の参考と なれば幸いである。

Delphi/400 俎中 侑 Delphi/400 福井和彦 石山 踏け Delphi/400 佐田 雄一 SmartPad4i 國元祐二 Valence 尾崎 浩司