## Delphi/400 上級

吉原泰介

RAD事業部 技術支援課

# [Delphi/400] RAD Serverを使った新しい 多層アプリケーション構築

はじめに
 RAD Server の特徴
 RAD Server の実装手順
 サーバーアプリケーション構築手順
 クライアントアプリケーション構築手順
 RAD Server の管理分析機能
 補足:他言語からの活用
 おわりに



1978年3月26日生まれ 2001年3月龍谷大学法学部卒業 2005年7月株式会社ミガロ、入社 2005年7月システム事業部配属 2007年4月RAD事業部配属

現在の仕事内容 Delphi/400を中心に製品試験およ び月 100件に及ぶ問い合せサポー トやセミナー講師などを担当してい

# 1.はじめに

ここ数年で Delphi/400 によるモバイ ルアプリケーションの業務開発も多く なってきた。そうしたモバイルアプリ ケーション開発において、IBM i を利用 するために重要となるのが中間サーバー を使った多層構成の仕組みである。【図 1】

Delphi/400 でこうした多層アプリ ケーションを開発する場合、サーバーも クライアントも Delphi/400 だけですべ て開発することができる強みがある。こ の中間サーバーのアプリケーション開発 では、DataSnap Server という技術が 使われているが、Delphi/400 の最新版 である 10.2 Tokyo では、新しく RAD Server というサーバー構築技術が利用 できるようになっている。

本稿ではこの新しい RAD Server に ついての特徴や実装方法、そして RAD Server を活用した他言語アプリケー ションとの連携テクニックなどの応用に ついて検証した内容を説明する。 なお本稿では RAD Server が利用で きる Delphi/400 10.2Tokyo の環境を 前提としている。

# 2.RAD Serverの特徴

新しく利用できるようになった RAD Server について、主な特徴を確認する。

まず RAD Server は、DataSnap Server と同様に、多層アプリケーショ ンの中間サーバー部分を実装する技術で ある。多層アプリケーションとは、複数 層で構成されたアプリケーションのこと で、たとえば Web アプリケーションや モバイルアプリケーションで IBM iに アクセスする場合には、中間層のサー バーを経由する 3 層構成となる。この中 間サーバーのアプリケーションを実装で き る の が DataSnap Server や RAD Server である。

ではこの2つの技術にどのような特 徴の違いがあるかをまとめてみる。

まず、従来のバージョンでも使用可能

な DataSnap Server は、次のような特 徴を持つ。

- DataSnap Server
- ・多層アプリケーションの開発を可能に する SDK
- ・サーバー機能はプログラムで開発する 必要がある
- ・開発での実装となるため、プログラム の自由度が高い
- ・TCP/IP、HTTP(S)、REST、 JSON、COM などの 標準技術をサポート

次に RAD Server は次のような特徴 を持つ。

## • RAD Server

- ・多層アプリケーションの REST API
   を公開するサーバー
- ・サーバーで必要となる高度な機能がい くつも提供されている
- ・ユーザー管理機能、認証機能、分析機 能などの標準機能を豊富に搭載



# 図2 DataSnap ServerとRAD Serverの違い

	DataSnap Server	RAD Server
機能開発	全て開発で実装が必要	必要な部分のみ開発
標準通信	TCPIP/HTTP(S)	HTTP(S) (REST形式固定)
DBエンジン	FireDAC、 dbExpress	FireDAC
モバイル対応機能	開発が必要	<b>Push</b> 通知、デバイス認証等が 標準機能
管理ツール	開発が必要	標準で付属(分析も可能)
ライセンス	開発ライセンスに含まれる (Enterprise以上)	開発ライセンスに 1サイトライセンス付属 ( <b>10. 2 Tokyo Enterprise</b> 以降)

# 図3 RAD Serverを使った多層アプリケーションの実装例



95

・HTTP (S)、REST、JSON などの標 準技術をサポート

細かい比較については【図 2】にまと めている。

比較するとわかるが、最も大きな違い は DataSnap Server はサーバーアプリ ケーションを細かく自由に開発すること を目的とした技術で、RAD Server は 大枠が完成されたサーバーアプリケー ションを、カスタマイズをして使う技術 ということである。

RAD Server は標準の機能がいろい ろ揃っている分、DataSnap Server と 違ってアプリケーションの仕組みは REST 形式で固定となっている。REST とは Representational State Transfer の略で、そのサービスの URL が持つメ ソッドにアクセスすることでデータの送 受信をステートレスで行う技術である。 汎用性が高く、Web やモバイルのアプ リケーションで広く使用されている。そ のため、Delphi/400 のアプリケーショ ンだけに限定されずさまざまなアプリ ケーションから活用することもできる (これについては後述する)。

どちらが優れているかは開発するア プリケーションの要件によっても異なる が、シンプルな機能の中間サーバーであ れば RAD Server の標準機能が強みを 発揮できる。

この2つの技術は、中間サーバーで 担っている役割は似ているが、実際の実 装手順は異なる部分も多い。そのため、 次章では新しい RAD Server の基本的 な実装手順について確認していく。

# 3.RAD Serverの 実装手順

RAD Server で構築するアプリケー ションは、DataSnap Server と同様に、 サーバーアプリケーションとそれを利用 するクライアントアプリケーションの2 種類で構成される。それぞれの実装手順 を2つのステップに分けて説明してい く。

・3-1.サーバーアプリケーションの構築
・3-2.クライアントアプリケーションの 構築 なお本稿では、【図3】に示すように PC やモバイルのクライアントアプリ ケーションから IBM iのデータを取得 する基本的な多層アプリケーションを題 材とする。

3-1.サーバーアプリケーション構築 手順

初めに中間サーバーに実装するアプ リケーションを構築する。構築は RAD Server にウィザードが用意されている ため、それほど難しいものではない。

## 【手順①プロジェクトの作成】

新規作成より、RAD Server (EMS) パッケージを選択してウィザードを起動 する。【図 4】

EMS という名称は、RAD Server 関 連の機能を意味する。

RAD Server が当初 EMS Server と いう製品名であったため、機能やコン ポーネント名で使われていることが多 い。

## 【手順②リソースの指定】

次にウィザードに従って設定を進め ることになるが、【図5】のように「リソー スを含むパッケージを作成する」を選択 して、そのリソース名を任意で命名する。

RAD Server は REST 形式になるた め、URL でアクセスする際に利用する 機能をリソースとして実装する。ここで 設定しておくと、指定したリソース名の ソースが作成される。たとえば【図5】 のように「CUST」というリソース名を 指定すると、実行時に下記のような REST サービスとして呼び出すことがで きる。

## http (s) :// サーバー /CUST

## 【手順③機能の指定】

②で指定したリソースに対して、どう いった機能のテンプレートを作成するか を【図6】で設定する。デフォルトは基 本機能としてデータを取得する Get、 GetItem が選択されているが、Put、 PutItem、DeleteItem といった更新系 の機能も用意されている。なお名称に Item と付く機能は、パラメータを渡し て処理できることを意味している。 【手順④自動生成されるソース】

ここまでのウィザード操作が完了す ると、【図7】のように中間サーバーア プリケーションのソース一式が自動生成 される。

次の手順からはこのソースに必要な プログラムを実装していく。

【手順5)コンポーネントの配置と設定】

自動生成されたリソースに機能を実 装するにあたって、まずは必要なコン ポーネントを配置する。RAD Server で使えるデータベースエンジンは最新の FireDAC のみに限定されている。その ため、ここではデータ取得(Get)の仕 組みを作ることを前提に FireDAC の主 要なコンポーネントを配置する。【図 8】

 $\cdot \text{ TFDConnection}$ 

- TFDPhysCO400DriverLink
- TFDTable
- TFDSchemaAdapter
- $\cdot \ {\rm TFDStanStorageJSONLink}$
- TFDGUIxWaitCursor

TFDConnection ~ TFDTable およ びTFDGUIxWaitCursor については、 一般的なFireDACのアプリケーション で実装する内容である。本稿では FireDACの詳しい使い方は割愛するが、 【図9】のようにTFDConnection や TFDTable を設定する。

RAD Server の構築でポイントにな るのはTFDSchemaAdapterと TFDStanStorageJSONLinkコンポーネ ントである。これはRAD Serverが REST形式として動作することが前提と なるため、TFDTableで取得したデー タをJSON形式に変換する機能の実装 に使用する。【図9】でTFDTableの SchemaAdapter プロパティに TFDSchemaAdapterを設定している のは、そのためである。

# 【手順⑥データ取得の機能を実装】

次に RAD Server がリクエストに応 じてデータを返す機能をプログラムで実 装する。ウィザードでリソースの Get 機能を指定しておけば、自動的に Get メソッドが作成されているので、ここで 必要な処理だけを数行コーディングす る。【ソース1】



処理内容としては TFDTable でデー タを開き、そのデータを TFDSchema Adapter で JSON 形式のストリームに 変換するだけである。この時、JSON 形 式 を 指 定 す る た め に TFDStanStorageJSONLink が必要と なっている。また図 6 で Item を使った 機能を実装する場合は、パラメータであ る ARequest を 使 っ て ARequest. Params.Values ['XXXX'] という形で 簡単に利用することができる。

プログラムの実装はこれで一通り完 了である。

#### 【手順⑦コンパイルと実行】

最後に作成した RAD Server の中間 サーバーアプリケーションをコンパイル して実行する。ただし初回のコンパイル 時にはコンポーネントのパッケージ開発 時と同様 (RAD Server もパッケージ 方式) にパッケージの参照確認や InterBase の設定確認のダイアログが表 示されるので応答が必要となる。【図 10】

InterBase は意識して使用するわけで はないが、RAD Server が標準で備え る管理分析機能等で内部的に使用してい る。なお InterBase のライセンスは RAD Server に含まれているので別途 購入の必要はない。

コンパイルが完了してアプリケー ションを実行すると、【図 11】のような 画面が起動して完成である。この画面で は、自動的にアクセスログが出力される ようになっており、クライアントからリ クエストがあるとこの画面にログが表示 される。

この起動画面はあくまでRAD Server に用意された標準の exe アプリケーショ ンである。DataSnap Server と違い、 作成したプログラム自体がサーバーとし て起動しているわけではない。

コンパイルしたリソースは bpl として 作成されて RAD Server に読み込まれ ている。運用環境の構築やリソースの配 布をする際には、エンバカデロ社の下記 オンラインヘルプに詳細情報が公開され ているため、参考いただきたい。

http://docwiki.embarcadero.com/ RADStudio/Tokyo/ja/

3-2.クライアントアプリケーション 構築手順

次に、前節で作成した RAD Server アプリケーションに接続してデータを取 得するクライアントアプリケーションを 構築する。

# 【手順①プロジェクトの作成】

新規作成よりマルチデバイスアプリ ケーションを選択して、FireMonkey の新規プロジェクトを作成する。【図 12】

今回はモバイルでの動作を見るため に FireMonkey で作成しているが、デ スクトップアプリケーションとして VCL で作成する場合も実装内容は同じ である。

【手順②コンポーネントの配置・設定】 新規作成したアプリケーションに必 要なコンポーネントを配置する。【図 13】

#### • TEMSProvider

- TEMSFireDACClient
- TFDSchemaAdapter
- TFDTableAdapter
- $\cdot \text{ TFDMemTable}$
- TFDGUIxWaitCursor

このアプリケーションでも FireDAC コンポーネントを使用するが、直接 IBM i に接続するわけではなく、前節で 作成した RAD Server アプリケーショ ンに接続してデータを取得する仕組みと なる。そのため、TFDConnection では な く、TEMSProvider お よ び TEMSFireDACClient の専用コンポー ネ ン ト を 使 用 す る (EMS は RAD Server 関連を意味する)。

また RAD Server より取得するデー タは JSON 形式で送られてくるため、 それを Delphi/400 のデータセット形式 に変換して戻すために TFDSchemaAdapter を使用する。そ れぞれのプロパティ設定は【図 14】に 示す。

ポイントとしては TEMSProvider で 接続する RAD Server の IP やポートを 指定し、TEMSFireDACClient で呼び 出すリソース情報を設定する点である。 リソース名は RAD Server のウィザー

#### ドで指定したものを設定する。

なおデータの画面表示はFireMonkey なので、【図 15】のようにライブバイン ディングを使って実装できる。VCLの 場合は TDBGrid を使うこともできる。

# 【手順③ RAD Server の呼び出し機能の 実装】

最後に RAD Server からデータ取得 の Get 機能を REST で呼び出す処理を プログラムで実装する。RAD Server はほとんどのアクセス制御を前手順のコ ンポーネントで処理してくれるため、 コードは1行で済む。【ソース2】

これでクライアントアプリケーショ ンのデータアクセスのプログラムが完成 したことになる。RAD Serverと DataSnap Serverの開発は似ているが、 RAD Serverでは標準機能がコンポー ネントなどで備わっている分、簡単に実 装ができる部分も多い。

#### 【手順④コンパイル・実行】

完成したプログラムを目的のプラッ トフォームを指定してコンパイルして実 行すると、Windows や iOS、あるいは Android 向けにアプリケーションを利 用することができる。【図 16】

全体的な仕組みは冒頭で説明した【図 3】のとおりである。

# 4.RAD Serverの 管理分析機能

前章で RAD Server の基本的な実装 手順を説明したが、RAD Server が持 つ機能について少しだけ補足を加える。 3-1. で実装した RAD Server のアプ リケーションは、起動画面でアクセスロ グが確認できることを説明したが、利用 できる機能はそれだけではない。

RAD Server は管理機能を標準で備 えているため、たとえばサーバーに対す るアクセス分析などの機能は開発しなく とも自動で実装されている。使い方とし ては起動画面メニューの「コンソールを 開く」ボタンをクリックする。【図17】

そうすると【図 18】のようなコンソー ルのログイン画面がブラウザで起動す る。このログインはユーザー名が consoleuser、パスワードが consolepass がデフォルトになっている。



このアカウント情報は RAD Server 上の下記iniファイルに設定されており、 テキスト編集で変更が可能である。

C:¥Users¥Public¥Documents¥Emba rcadero¥EMS¥emsserver.ini

ログインすると、【図 19】のように RAD Server にアクセスするクライア ントの状況や作成した REST リソース の使用頻度について、グラフを使ったビ ジュアル分析が行える。これによってど の機能がよく使われているか、どんな時 間帯に処理が集中しているか、などの把 握も容易である。

# 5.補足:他言語からの 利用

ここまで RAD Server を使った基本 的なアプリケーションの実装内容や機能 についてまとめてきたが、最後に RAD Server をさらに活用するための使い方 を考察していく。

RAD Server が REST 形式になるこ とは冒頭で説明したが、この REST 形 式はここまでの実装でも説明したとお り、JSON を利用できる。この REST/ JSON の組み合わせは、非常によく使用 される通信方式であるため、実は Delphi/400 以外の他言語アプリケー ションでも利用可能である。

本稿では他言語の例として、Delphi 言語と同じ開発元であるエンバカデロ・ テクノロジーズ社が提供する Sencha Architect を題材として RAD Server の利用を説明する。Sencha Architect とは、表、グラフ、ツリーなどの多彩な コンポーネントをドラッグ&ドロップし て簡単に HTML5 対応の Web アプリ ケーションを構築できる開発ツールであ る。【図 20】

こ の Sencha Architect は REST/ JSON データと連携してデータを扱うこ ともできるため、RAD Server とも容 易に連携ができる。

## ● RAD Server の実装調整

3章で実装した GET 機能の JSON は Delphi/400 のコンポーネントに特化し た作り方となっているため、もっと一般 的な JSON 形式に調整する。【ソース 3】 ● Sencha Architect からの利用

本稿では Sencha Architect の詳し い開発手順については割愛し、ポイント になる部分を説明する。

Sencha Architect に も Delphi/400 と同じようなコンポーネント(部品)が 用意されているため、RAD Serverか ら取得したデータを表示するために Grid Panelというコンポーネントをプ ログラムに配置する。【図 21】

この Grid Panel は Delphi/400の TDBGrid と扱いが似ている。

次に Grid Panel のデータ参照元を 指定する。

これは Grid Panel に Grid Builder と いうウィザード機能が用意されており、 【図 22】のように JSON WebService 形 式で RAD Server の URL が参照元とな るように設定する。これによってフィー ルド情報なども自動で取り込んで定義で きる。

このデータを Grid Panel の参照元に セットすれば、Delphi/400 と同じよう に Sencha Architect 上で RAD Server からの情報を表示することができる。【図 23】

このプログラムを Sencha Architect でコンパイルして実行するとブラウザで Web アプリケーションとして実行され、 RAD Server の情報を表示することが 可能である。【図 24】

Sencha Architect の細かい設定手順 は省略しているが、RAD Server の REST/JSON 形式は非常に汎用性が高 く、別の言語でも簡単に扱えることがわ かる。

もちろん Sencha Architect に限ら ず、他の言語でも REST/JSON が扱え れば同じように使用が可能である。これ を理解していれば RAD Server をより 広い範囲で活用することができる。

# 6.おわりに

本稿では、新しい多層アプリケーショ ンの構築技術として RAD Server の特 徴や実装手順について説明した。従来の DataSnap Server と 違 っ て、RAD Server は開発形式がある程度限定され ているものの、あらかじめ専用コンポー ネントやサーバーの管理機能を備えてい るため、効率よくアプリケーションを実 装するには効果的である。

また5章で述べたように、この REST サービスは非常に汎用性が高く、 Delphi/400 以外の言語でも利用するこ とができる。

今後、クラウドやソリューションを含 めて多様化していくシステム連携の中心 となるサーバーとして活用が期待でき る。

 $\mathbf{M}$ 

$-\phi$	
図10 初回コンパイル時の応答 パンクージの参照 「いたいたうでの参照 「「いたいた」」」を登るす: Control のはののですが、 「いたいた」」」を登るす: Control のはののですが、 「いたいた」」」を登るす: Control のはのの。 「いたいた」」」を登るす: Control のはのの。 「いたいた」」」を登るす: Control のはのの。 「いたいた」」」を登るす: Control のはのの。 「いたいた」」」」を登るす: Control のはのの。 「いたいた」」」」を登るす: Control のはのの。 「いたいた」」」」」」	
<pre>ソース1 データ取得の機能を実装 procedure TCUSTResource1.Get(const AContext: TEndpointContext; const ARequest: TEndpointRequest; const AResponse: TEndpointResponse); var oStr: TMemoryStream; begin oStr := TMemoryStream.Create; // クエリの実行結果をスキーマアダプタから // メモリストリーム経由で返す FDTable1.Open; FDSchemaAdapter1.SaveToStream(oStr,TFDStorageFormat.sfJSON); AResponse.Body.SetStream(oStr, 'application/json', True); FDTable1.Close; end;</pre>	
図11 RAD Serverの起動 ADD Server定動画面面	
	101

 $\oplus$ 



第 173/201-1/324/15         10000001	層 オブジェクト インスペクタ — □ X MemTable1 TFDMemTable  ▼	
Bind Source DB           Differ         Colspan="2">Contrained Cognet Contrained Contred Contred Contrained Contrained Contrained Contrained Contrat	MemTable1 TFDMemTable ● 検索	
修業 <b>プロボマイ 472トト</b> Form1 - デフォルトレイヤ           Advector         Bind SourceDB1         StringGid1           Advector         Diskeday         Form1 - デフォルトレイヤ           Changeduction         Form2 - String Gid1         String Gid1           Changeduction         Form3 - String Gid2         String Gid2           Changeduction         Form3 - String Gid2         String Gid2           Changeduction         Form3 - String Gid2         String Gid2           Max         Changeduction	)検索	Livebilding 7.947
20/27(-2/2-/->         AdviceSurvedUsage       (xxDosignTime, xxDosignTime, xxDosignTime		Form1 - デフォルトレイヤ
Proce         Proce         Proce         Proceedings         String Grid           Adverse         Formation		
Alapper         FDTAblokAppter1           hypergenerative         False           Aurocalafieds         False           CachedUpdate         False           Constraints         COneckConstraints)           27-UK 157/09.2016         Constraints           Constraints         COneckConstraints)           27-UK 157/09.2016         Constraints           Constraints         Constraints           Proceedure TForm1.Button1Click (Sender: Tobject);           begin         //BREU-TUNAUP-ZOGetDataEoFIFOHDT           //BREU-TUNAUP-ZOGetDataEoFIFOHDT           EMSFireDACClient1.GetData;           end;	Active Lass ActiveStoredUcage [auDesignTime,auRunTime]	A RindSourceDB1 StringGrid1
Public action         Constraints         Public True           Constraints         Faste           Propedure TForm1.ButtonlClick (Sender: Tobject);           begin         // 設定しているリンースのGetDataを呼び出す           // 設定しているリンースのGetDataを呼び出す         Faste           EMSFireDACClient1.GetData;         Faste           121         D7/974/D2020-20-105           3444         f4/2004-2444           f4/2004-2444         f4/2004-2444           f31         Publick           1231         PD7974/20420-20-1	Adapter FDTableAdapter1	
AutoCulfields ChangeAlterity ChangeAlterity Constraints (CheckConstraints)       False ChangeAlterity Constraints (CheckConstraints)         2-bit 17:0-0       Dynamic 10, 004 (CheckConstraints)         2-constraints (TCDeckConstraints)       Dynam	Aggregates (11 D'Aggregates) AggregatesActive	E FDMemTable1 *
Cohengkletkare       False         Changkletkare       (TheckConstaints)         Cohengkletkare       (TheckConstaints)         Proceedure TForm1.ButtonlClick(Sender: Tobject);       (Segin         //設定しているリソースのGetDataを呼び出す       (Sender: Tobject);         EMSFireDACClient1.GetData;       (Sender: Tobject);         I21       1000000000000000000000000000000000000	AutoCalcFields 🛛 True	
Contraints       (TherekConstraints)         Constraints       (TherekConstraints)         Proceedure TForm1.Button1Click (Sender: Tobject);       (Seconstraints)         //設定しているリソースのGetDataを呼び出す       (Seconstraints)         EMSFire DACClient1.GetData;       (Seconstraints)         end;       (Seconstraints)         1221       127-y79702/920-y2-         1333       9/1/0297-Nas         1331       9/1/0297-Nas         1331       9/1/0297-10-10-10         1384       MeM9/1(-7297)         13	CachedUpdates ChangeAlerter	
Constraints       (TCheckConstraints)       ▼         YWit Style, JP-Aublesk       Y         Y-Cougab/装売されています       Y         Y-Cougab/装売されています       Y         Y-Cougab/装売されています       Y         Y-Cougab/装売されています       Y         Y-Cougab/装売されています       Y         Y-Cougab/装売されています       Y         Pocodure TForm1.ButtonlClick (Sender: Tobject);       Y         Begin       //設たているリソースのGetDataを呼び出す         EMSFireDACClient1.GetData;       end;         6       プラットフォーム別にコンパイル・実行         Image: Second Se	ChangeAlertName	BBB Sart
27-0FX5763 27-0FX5763 27-0FX5763 27-0FX5775 27-0FX55 27-0FX55 27-0F	Constraints (TCheckConstraints)	
<sup>12</sup> 3-7.20月前援表示されています <b></b>	'ールド エディタ ファイルから読み込み ータセットの割り当て データセットの編集	
<ul> <li> </li></ul>	3- アルトパインド カベルカ毎年 べての項目が表示されています	
<ul> <li>ス2 データ取得の機能を実装</li> <li>procedure TForm1.Button1Click(Sender: TObject); begin //設定しているリンースのGetDataを呼び出す EMSFireDACClient1.GetData; end;</li> <li>6 プラットフォーム別にコンパイル・実行</li> <li></li></ul>		-
NLL 3     アイストックスパトックスARE C XX       procedure TForm1.Button1Click(Sender: TObject);       begin       //設定しているリソースのGetDataを呼び出す       EMSFireDACClient1.GetData;       end;   6 プラットフォーム別にコンパイル・実行        ID     NAME       1221     コフナッツマリンショップ2       1513     ダイブパウスタートル5       3444     ダイビングペース新井8       1231     アクアダイビングセンター       1351     ヘレビングセンター       1351     ヘレビングセンター       1364     MHMダイパースクラブ       1384     ADVENTURE UNDERSEA       2113     グリーンスポーツクラブ       1384     ADVENTURE UNDERSEA       2113     グリーンスポーツクラブ       1314     グリーンスポーツクラブ       1315     グリーンスポーツクラブ       1316     ワンプボーツクラブ       1317     グリーンスポーツクラブ       1318     グリーンスポーツクラブ       1319     グリーンスポーツクラブ       1310     グリーンスポーツクラブ	2 データ取得の機能を実装	
procedure TForm1.Button1Click(Sender: TObject);         begin         //設定しているリソースのGetDataを呼び出す         EMSFireDACClient1.GetData;         end;         6       プラットフォーム別にコンパイル・実行		
begin //設定しているリソースのGetDataを呼び出す EMSFireDACClient1.GetData; end; 6 プラットフォーム別にコンパイル・実行 <sup>●</sup> Form1 - □ × <sup>10</sup> NAME <sup>1221</sup> ココナッツマリンショップ2 <sup>1513</sup> ダイバハウスタートル5 <sup>3444</sup> ダイビングペース新井8 <sup>1231</sup> P27ダイビングセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブセンター <sup>1351</sup> 竜山ダイブマンター <sup>1350</sup> ダイブショップブルーリーフ <sup>1364</sup> MHMダイバーズクラブ <sup>1364</sup> ADVENTURE UNDERSEA <sup>1964</sup> ADVENTURE UNDERSEA <sup>2118</sup> <i>ワ</i> リーンスポージワラブ <sup>2118</sup> <i>ワ</i> リーンスポージワラブ	procedure TForm1.Button1Cli	ck(Sender: TObject);
	oegin	-* as5-* 111-+
end; end; 6 プラットフォーム別にコンパイル・実行	//設定しているリソースのGetData	aを呼び出す
6 プラットフォーム別にコンパイル・実行 <u> </u>	EMSEiroDACClient1 CotData	
6 プラットフォーム別にコンパイル・実行 ● Form1 - □ × 10 NAME 1221 ココナッツマリンショッブ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1350 ダイブショッブブルーリーフ 1384 MHMダイバーズクラブ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2156 マリンハウスペんざん	EMSFireDACClient1.GetData	;
6 プラットフォーム別にコンパイル・実行	EMSFireDACClient1.GetData end;	;
6 プラットフォーム別にコンパイル・実行 10 NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 風山ダイブセンター 1351 風山ダイブセンター 1360 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1384 MHMダイバーズクラブ 1384 MHMダイバーズクラブ 1384 MHMダイバーズクラブ 1384 MHMダイバーズフラブ 1384 MHMダイバーズフラブ 1384 MHMダイバーズフラブ 1384 MHMダイバーズスラブ 1384 MHMダイバーズスラブ 1384 MHMダイバーズス 2118 グリーンスポーツクラブ	EMSFireDACClient1.GetData end;	;
6 プラットフォーム別にコンパイル・実行 D NAME	EMSFireDACClient1.GetData end;	;
6 プラットフォーム別にコンパイル・実行	EMSFireDACClient1.GetData end;	;
Form1       ー       〇       ×         ID       NAME        ●       NAME         1221       コフナッツマリンショッブ2       ●       ●       NAME         1513       ダイブハウスタートル5       ●       ●       NAME         3444       ダイビングベース新井8       ●       ●       NAME         1231       アクアダイビングセンター       ●       ●       NAME         1351       亀山ダイブセンター       ●       ●       ●       ●         1380       ダイブショッブブルーリーフ       ●       ●       ●       ●       ●         1384       MHMダイバーズクラブ       ●       ●       ●       ●       ●       ●         1984       ADVENTURE UNDERSEA       ●	EMSFireDACClient1.GetData end;	;
ID       NAME         1221       ココナッツマリンショッブ2         1513       ダイブハウスタートル5         3444       ダイビングペース新井8         1221       ココナッツマリンショップ2         1513       ダイブハウスタートル5         3444       ダイビングペース新井8         1231       アクアダイビングセンター         1351       亀山ダイブセンター         1380       ダイブショッブブルーリーフ         1384       MHMダイバーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         2118       グリーンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行	;
ID     NAME       1221     ココナッツマリンショップ2       1513     ダイブハウスタートル5       3444     ダイビングベース新井8       1231     アクアダイビングセンター       1351     亀山ダイブセンター       1380     ダイブショップブルーリーフ       1384     MHMダイバーズクラブ       1984     ADVENTURE UNDERSEA       2118     グリーンスポーツクラブ       2118     グリーンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行	;
1513       ダイブハウスタートル5         3444       ダイビングベース新井8         1231       アクアダイビングセンター         1351       亀山ダイブセンター         1350       ダイブショップブルーリーフ         1384       MHMダイバーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         2118       グリーンスポーツクラブ         2118       グリンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1	
3444       ダイビングベース新井8         1231       アクアダイビングセンター         1351       亀山ダイブセンター         1350       ダイブショップブルーリーフ         1380       ダイブショップブルーリーフ         1384       MHMダイバーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         2118       グリーンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1	; - • ×
1231       アクアダイピングセンター         1351       亀山ダイブセンター         1350       ダイブショッブブルーリーフ         1384       MHMダイパーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         2118       グリーンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5	X Phone X O NAME 1221
1351       亀山ダイブセンター         1351       亀山ダイブセンター         1380       ダイブショップブルーリーフ         1384       MHMダイバーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         1380       ダリーンスポーツクラブ	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 P Form1 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 44 ダイビングベース新井8	- □ ×
1380       ダイブショップブルーリーフ         1380       ダイブショップブルーリーフ         1384       MHMダイバーズクラブ         1384       MHMダイバーズクラブ         1984       ADVENTURE UNDERSEA         2118       グリーンスポーツクラブ         2118       グリーンスポーツクラブ         2156       マリンハウスべんぎん	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 コンチッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター	- ・ × ・ ・ × ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
1384         MHMダイパーズクラブ         1984         ADVENTURE UNDERSEA           1984         ADVENTURE UNDERSEA         2118         グリーンスポーツクラブ           2118         グリーンスポーツクラブ         2135         パイナップルダイバーズ           2156         マリンハウスべんぎん	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 D NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター	×
1984         ADVENTURE UNDERSEA         2118         グリーンスポーツクラブ           2118         グリーンスポーツクラブ         2135         パイナップルダイバーズ           2156         マリンハウスべんぎん	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 D NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 380 ダイブショップブルーリーフ	- □ × ► Phone × ○ ○ ▷ NAME 1221 □□ナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース航井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ
2118 グリーンスポーツクラブ 2156 マリンハウスべんぎん	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 380 ダイブショップブルーリーフ 384 MHMダイバーズクラブ	- □ ×
a sea and a set of a set of a set	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 D NAME 221 コゴナッツマリンショッブ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 384 MHMダイバーズクラブ 964 ADVENTURE UNDERSEA	- □ ×
2135 パイナップルタイパース 2163 SCUBA HEAVEN	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 Form1 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 513 ダイブハウスタートル5 513 ダイブハウスタートル5 513 ダイブングベース新井8 231 アクアダイピングセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブショップブルーリーフ 584 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ	; <b>X</b> <b>NAME</b> <b>IZ21</b> $\Box$ $J$ $J$ <b>NAME</b> <b>IZ21</b> $\Box$ $J$ $J$ $J$ <b>NAME</b> <b>IZ21</b> $\Box$ $J$
2156 マリンハウスべんぎん 2165 SHANGRI-LA SPORTS CENTER	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 364 MHMダイバーズクラブ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 355 パイナッブルダイバーズ	- □ X
2163 SCUBA HEAVEN 2353 マリンセンター巣柿浦	EMSFireDACClient1.GetData end; プラットフォーム別にコンパイル・実行 Form1 Form1 221 ココナッツマリンショッブ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 364 MHMダイバーズクラブ 364 MHMダイバーズクラブ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 355 バイナッブルダイバーズ 356 マリンハウスべんぎん	- □ ×
	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 380 ダイブショップブルーリーフ 384 MHMダイバーズクラブ 384 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 135 バイナッブルダイバーズ 156 マリンハウスべんぎん 163 SCUBA HEAVEN	- □ ×
2165 SHANGRI-LA SPORTS CENTER 2975 ダイブリゾートパパイヤ	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 364 MHMダイバーズのラブ 364 MHMダイバーズクラブ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 365 パイナップルダイバーズ 366 マリンハウスべんぎん 363 SCUBA HEAVEN 365 SHANGRI-LA SPORTS CENTER	- □ ×
2165         SHANGRI-LA SPORTS CENTER         2975         ダイブリゾートパパイヤ           2315         ダイビングショップマーメイド         2984         サンセットダイビングサービス           3022         ブリントパパイマ         3041         マリンショップアクア	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 D NAME 221 コゴナッツマリンショッブ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 360 ダイブショップブルーリーフ 384 MHMダイバーズクラブ 384 MHMダイバーズクラブ 384 MHMダイバーズクラブ 384 MHMダイバーズクラブ 385 パイナッブルダイバーズ 356 マリンハウスペんぎん 353 SCUBA HEAVEN 355 SHANGRI-LA SPORTS CENTER 315 ダイビングショップマーメイド	- □ ×
2165     SHANGRI-LA SPORTS CENTER     2975     ダイブリゾートパパイヤ       2315     ダイビングショップマーメイド     2984     サンセットダイビングサービス       2353     マリンセンター単物浦     3041     マリンショップアクア       2075     ダイブリゾートパパイヤ     3042     HAJIMAダイビングスクール	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 360 ダイブショップブルーリーフ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 365 SHANGRI-LA SPORTS CENTER 365 SHANGRI-LA SPORTS CENTER 375 ダイビングショップマーメイド 353 マリンセンター戦物満	- □ ×
2165     SHANGRI-LA SPORTS CENTER     2975     ダイブリゾートパパイヤ       2315     ダイビングショップマーメイド     2984     サンセットダイビングサービス       2353     マリンセンター単柿浦     3041     マリンショップアクア       2975     ダイブリゾートパパイヤ     3042     HAJIMAダイビングスクール       2984     サンセットダイビングスクール     3051     ペンションあしか	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 P Form1 P For	- □ X
2165     SHANGRI-LA SPORTS CENTER       2315     ダイビングショップマーメイド       2353     マリンセンター巣柿浦       2975     ダイブリゾートババイヤ       2984     サンセットダイビングサービス       3041     マリンショップアクア       3041     マリンショップアクア	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 コンナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 580 ダイブショップブルーリーフ 584 MHMダイバーズクラブ 584 MHMダイバーズクラブ 584 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 135 パイナッブルダイバーズ 156 マリンハウスペんぎん 163 SCUBA HEAVEN 165 SHANGRI-LA SPORTS CENTER 315 ダイブリンター巣柿満 575 ダイブリゾートバU(イヤ 584 サンセットダイビングサービス 541 マリンショップアクア	- □ X             ×         ×           ×         ×             ×         ×           ×         ×             ×         ×           ×         ×             ×         ×           ×         ×             ×         ×           ×         ×                 ×         ×           ×         ×                 ×         ×           ×         ×         ×
2103 SCUDA HEAVEN 2353 マリンセンター巣枋浦	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 513 ダイブハウスタートル5 514 ダイビングベース新井8 231 アクアダイビングセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブセンター 551 亀山ダイブショップブルーリーフ 584 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 135 パイナッブルダイバーズ 156 マリンハウスべんぎん	- □ ×
2165 SHANGRI-LA SPORTS CENTER	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 380 ダイブショップブルーリーフ 384 MHMダイバーズクラブ 384 MHMダイバーズクラブ 384 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 135 パイナップルダイバーズ 156 マリンハウスべんぎん 163 SCUBA HEAVEN 165 SHANGRUL & SPORTS CENITER	- □ X
2165 SHANGRI-LA SPORTS CENTER 2975 ダイブリゾートパパイヤ	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 7 7 7 7 7 7 7 7 7 7 7 7 7	- □ X
2165         SHANGRI-LA SPORTS CENTER         2975         ダイブリゾートパパイヤ           2315         ダイビングショップマーメイド         2984         サンセットダイビングサービス           2352         ブリンクショップアクア         3041         マリンショップアクア	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 P Form1 P Form2 P Form1 P For	- □ ×
2165     SHANGRI-LA SPORTS CENTER     2975     ダイブリゾートパパイヤ       2315     ダイビングショップマーメイド     2984     サンセットダイビングサービス       2353     マリンセンター単物満     3041     マリンショップアクア       2075     ダイブリゾートパパイヤ     3042     HAJIMAダイビングスクール	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 P Form1 NAME 221 コンナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 亀山ダイブセンター 351 亀山ダイブセンター 360 ダイブショップブルーリーフ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 364 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 365 SUBA HEAVEN 165 SHANGRI-LA SPORTS CENTER 375 ダイビングショップマーメイド 353 マリンセンター製物満	- □ X
2165     SHANGRI-LA SPORTS CENTER     2975     ダイブリゾートパパイヤ       2315     ダイビングショップマーメイド     2984     サンセットダイビングサービス       2353     マリンセンター単柿浦     3041     マリンショップアクア       2975     ダイブリゾートパパイヤ     3042     HAJIMAダイビングスクール       2084     サンセットダムビングスクール     3051     ペンションあしか	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 P Form1 P Form1	- □ ×
2165     SHANGRI-LA SPORTS CENTER       2315     ダイビングショップマーメイド       2353     マリンセンター巣柿浦       2975     ダイブリゾートババイヤ       2975     ダイブリゾートババイヤ       2984     サンセットダイビングカービス       2984     サンセットダイビングスクール       2015     ブリシセットダイビングサービス       2984     サンセットダイビングスクール       2015     ブリシャトダイビングサービス       2984     サンセットダイビングスクール       3051     ペンションあしか       2984     サンセットダイビングフタービス	EMSFireDACClient1.GetData and; プラットフォーム別にコンパイル・実行 Form1 Form1 NAME 221 ココナッツマリンショップ2 513 ダイブハウスタートル5 444 ダイビングベース新井8 231 アクアダイビングセンター 351 竜山ダイブセンター 351 竜山ダイブセンター 360 ダイブショップブルーリーフ 384 MHMダイバーズのラブ 384 MHMダイバーズクラブ 384 MHMダイバーズのラブ 384 ADVENTURE UNDERSEA 118 グリーンスポーツクラブ 385 パイナッブルダイバーズ 156 マリンハウスべんぎん 363 SCUBA HEAVEN 365 SHANGRI-LA SPORTS CENTER 375 ダイブリゾートババイヤ 384 サンセットダイビングサービス	- □ ×

	圖 RAD Server (EMS) 開発サ−パー
	起動 停止 ブラウザを購く コンソールを開く
	ບຸລິ:
	{"ConfigLoaded":{"Filename":"C:¥Users¥Public¥Documents¥Embarcadero¥EMS¥emsserver.ini","Thread":10856}} {"DBConnection":{"InstanceName":"","Filename":"C:¥Users¥Public¥Documents¥Embarcadero¥EMS¥emsserver.ib","Thread":10856}}
	{"Licensing": {"Licensed": false, "DefaultMaxUsers": 5, "Thread": 10856}} {"Loading": {"Filename": "C: ¥Users¥Public¥Documents¥Embarcadero¥Studio¥19.0¥Bpl¥Project1.bpl", "Thread": 10856}}
	{ RegResource : { Resource : CUS1 , Endpoints : [ Get , GetItem ], Thread : 10856}}
図18	コンソールのログイン
	Embarcadero MEAP Sol X + V - O X
- 1	← → ♡ ŵ ○ localhost8081/       □ ☆ ☆ &      └ ☆ …       □// · ン ▶
	embar <sup>#1&gt;1&gt;</sup>
_	
	Enter tenant
	© 2016 Embarcadero Technologies, Inc.
_	デフォルトは
- 1	consoleuser/consolepass
⊠19	
	← → ○ @ localhost0031/ht/totalAPICallsEndPoint □ ★ ★ 2 @ ···· ← → ○ @ 0 loadecate(ht/totalAPICall) ▲ ユーザー ▲ ユーザー ● 07 001117 € 1   1 + 1 + 2 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
	C クループ     API 呼び出しエンドホインド アナリテイク     F API のは     C クループ     API の     C クループ     C ク
1	
	> PR-12P         III I I I I I I I I I I I I I I I I I
	サコーザーの 合計呼び出し版     3
	2 0007-04 0 1/2-F8821-45-58831





