

PSQL Connector for iOS

version 1.0



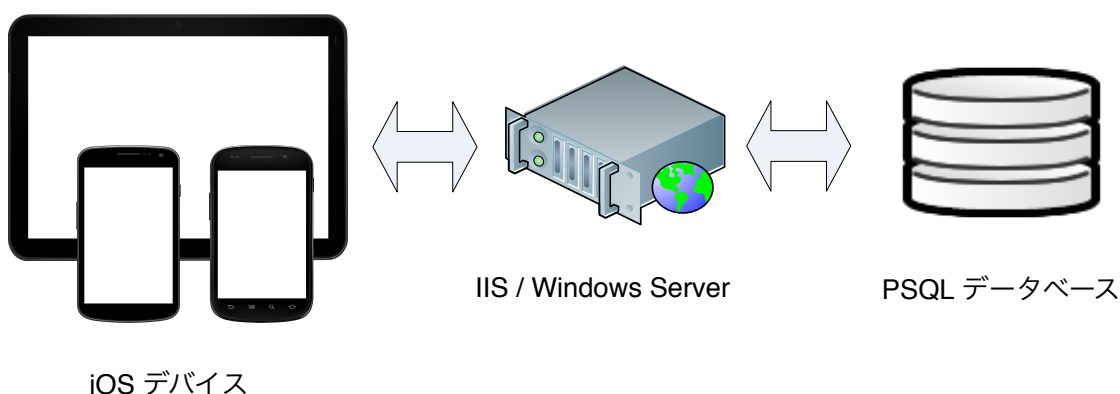
はじめに	4
PSQL CONNECTOR FOR IOSについて	4
ご利用上の注意事項	4
本製品の使用許諾について	5
禁止事項	5
保証規定	5
製品サポート	5
販売元	6
開発元・サポート	6
商標登録	7
インストール	8
サーバーシステム要件	8
モバイルデバイス要件	8
モバイル開発環境要件	8
セットアッププログラムの実行	8
IISサーバー設定について	9
データベース接続について	9
モバイルデバイスプログラミング	10
サンプルコード	10
XCODEプロジェクトでの設定について	10
メソッドリファレンス	11
beginTrans	11
commitTrans	11
connect	11
disconnect	12
execute	12
query	12
queryRows	13
rollbackTrans	13
プロパティリファレンス	14
lastErrorString	14
pwd	14
verbose	14
uid	14
url	14
デリゲートリファレンス	15

queryCompleted.....	15
queryRowsCompleted.....	15
requestCompleted	15
requestFailed	16
rowFetched.....	16
APPENDIX-A エラーコード	18
APPENDIX-B SWIFT サンプルコード	19
APPENDIX-C OBJECTIVE-C サンプルコード	21
APPENDIX-D サンプル WEB.CONFIG.....	22

はじめに

PSQL connector for iOSについて

本製品はiPadやiPhoneにActian PSQLデータを開示してSwift/Objective-Cアプリケーションを作成するお客様向けのモバイルミドルウェアです。Apple社のiOSモバイルデバイスでActian PSQLデータを利用するアプリケーションを容易に開発可能とします。



以下は製品の概要です。

1. PSQLサーバーとiOSデバイス間通信はIIS Web Service経由となります。
2. クライアントライブラリはXcode/Swift/Objective-Cに対応するスタティックライブラリとして提供されます。
3. クライアントライブラリは非同期通信でdelegateに処理完了などが通知されます。
4. トランザクションサポートしてますので更新系業務も対応可能です。
5. ネットワーク構成によりLTE(4G),Wi-Fiで利用可能です。
6. SSL通信に対応しています。
7. 各種 Web Server認証に対応しています。

ご利用上の注意事項

本製品では以下のような注意事項・ご利用制限がございます。

1. BLOB/CLOBなどラージオブジェクトには対応していません。
本製品についてはWebServiceを利用してPSQLと通信しているので現在のところラージオブジェクト（画像など）はFoundation frameworkのNSURLSessionなどを使ってデータを転送していただくのが良い方法だと思います。

本製品の使用許諾について

本製品のご利用ライセンスはサーバー毎となります。1台のサーバーに複数のモバイルデバイスから接続数は無制限でご利用いただけます。開発ライブラリは複数の開発環境でご利用いただけます。複数サーバーでの運用、異なるアプリケーションを別サーバーで運用する場合などはご利用サーバー数と同じ数のライセンスをAGTECHからご購入ください。

禁止事項

1. 本製品の不正複製を禁止します。
2. 本製品のリバースエンジニアリングを禁止します。
3. 本製品をラップし同種の製品を作成し配布・販売することを禁止します。

保証規定

本製品、および付随する著作物に対して商品性及び特定の目的への適合性などについての保証を含むいかなる保証もそれを明記するしないに関わらず提供されることはありません。

本製品の著作者及び、製造、配布に関わるいかなる者も、当ソフトウェアの不具合によって発生する損害に対する責任は、それが直接的であるか間接的であるか、必然的であるか偶発的であるかに関わらず、負わないものとします。それは、その損害の可能性について、開発会社に事前に知らされていた場合でも同様です。

製品サポート

ご購入いただいた初年度はサポートとメンテナンスが含まれています。次年度以降のサポートもしくはバージョンアップを含むメンテナンスは、別途メンテナンス契約が必要です。

販売元



株式会社エージーテック
〒101-0054
東京都千代田区神田錦町1-21-1
ヒューリック神田橋ビル3F

電話: 03-3293-5300
FAX: 03-3293-5270
E-Mail: info@agtech.co.jp
URL: <http://www.agtech.co.jp>

開発元・サポート



(株) テクナレッジ

東京都世田谷区駒沢2丁目16番1号 サンドービル9F

電話 03-3421-7621
FAX 03-3421-6691
E-Mail info@techknowledge.co.jp
URL <http://www.techknowledge.co.jp>

商標登録

本マニュアルに記載される商標、登録商標は該会社の商標または登録商標です。

インストール

PSQL connector iOS のインストールについてご説明します。

サーバーシステム要件

以下のソフトウェアがサーバーシステム構成の要件となります。

1. Windows Server 2012 (R2) または Windows Server 2008 (R2)
2. 上記のサーバーにて動作するMicrosoft IIS (Internet Information Server)
3. Microsoft .NET framework 4.0 以上
4. PSQL v12,v11

モバイルデバイス要件

Apple社のiOSデバイスとして以下をサーバーに接続可能です。

1. iPhone 各バージョン およびその後継機種
2. iPad 各バージョン
3. iPod Touch 各バージョン

モバイル開発環境要件

モバイルデバイス向けアプリケーション開発環境としては以下が必要になります。AppStoreより無償でご利用いただけます。

Xcode 7

モバイルデバイス実機でのデバッグやアプリケーションの配布には別途Apple Developer Program にご加入いただく必要があります。

セットアッププログラムの実行

製品パッケージのsetup.exeをサーバー上で実行すると以下がインストールされます。

1. サーバー実行環境
2. モバイルデバイス向け開発環境

3. サンプルコード

全てサーバー上に配布されますので開発に必要なヘッダー・ライブラリ類はネットワーク接続などで開発PC/Macに転送してください。

IISサーバー設定について

サーバー動作に必要なファイルはインストールフォルダー下のhtdocsフォルダーに転送されます。このフォルダー自体をIISで公開するか別のフォルダーに内容をコピーしてIISにてウェブサイトとして公開してください。

IIS の設定要件は以下になります。

1. .NET framework 4.0を利用可能にすること。
2. System.Data を利用可能な設定をすること。
3. セッションを有効にすること。
4. PSQLへの接続文字列をWeb.Configに設定すること。

上記2番についてWeb.Configでは以下のような設定となります。

```
<system.web>  
<compilation debug="false" strict="false" explicit="true" targetFramework="4.0" />  
</system.web>
```

上記4番については以下のような設定となります。

```
<connectionStrings>  
<add name="psqlLib" connectionString="Driver={Pervasive ODBC Engine  
Interface};ServerName=localhost;dbq=@DEMODATA;" />  
</connectionStrings>
```

データベース接続について

Web.configファイルに接続文字列セクションに以下のようなPSQLデータベース接続情報を追加してください。

```
<connectionStrings>  
<add name="psqlLib" connectionString="Driver={Pervasive ODBC Engine  
Interface};ServerName=localhost;dbq=@DEMODATA;" />  
</connectionStrings>
```

モバイルデバイスプログラミング

サンプルコード

サンプルコードはインストールディレクトリのsamplesフォルダーに以下のサンプルがございます。ファイル名にSwift 文字列が含まれるものはswift のサンプルコードです。

1. querySample.zip
2. queryRowsSample.zip
3. queryRowsiPadSample.zip
4. querySampleSwift.zip
5. queryRowsSampleSwift.zip

それぞれ展開してXcode 7.x で.xcodeprojファイルを読み込んでください。urlプロパティに設定しているサーバーのアドレス等は変更の必要があります。（プライベートメソッドstartConnect でurlプロパティを設定しています）

PSQLのDEMODATA Person表にアクセスするようにサーバー側の接続文字列を設定していただければすぐに動作するようになります。

Xcodeプロジェクトでの設定について

Xcodeのプロジェクトから本製品でPSQLデータベースアクセス機能を追加するには以下の2ファイルをプロジェクトに追加してください。

1. dblib.h
2. libdbLib.a

これらのファイルはサンプルコードやサーバー側にinclude/lib フォルダーに保存されています。libdbLib.a はiOSデバイスとシミュレータで動作するユニバーサルライブラリです。サポートアーキテクチャは armv7, armv7s, arm64, i386, x86_64 です。

swift プロジェクトで利用する場合にはbridging_headerにdbLib.h を設定してください。設定詳細はサンプルコードにありますので参照してください。

メソッドリファレンス

beginTrans

書式

-(dbLibStatus) beginTrans

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

トランザクションを開始します。サーバーには接続完了状態、トランザクションクローズ状態で呼び出してください。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

commitTrans

書式

-(dbLibStatus) commitTrans

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

トランザクションをコミットします。サーバーには接続完了状態、トランザクション状態で呼び出してください。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

connect

書式

-(dbLibStatus) connect

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

プロパティURLで指定されるサーバーに接続します。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

disconnect

書式

-(dbLibStatus) disconnect

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

サーバー接続を遮断します。サーバー側ではPSQLデータベース接続の遮断、トランザクション中の場合はトランザクションの破棄が実行されます。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

execute

書式

-(dbLibStatus) execute:(NSString*) sql

パラメータ

非クエリ系PSQL SQL

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

非クエリ系PSQL SQLを実行します。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

query

書式

-(dbLibStatus) query:(NSString*) sql

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

SQLで指定されるクエリを発行して1行ずつrowFetchedデリゲートにデータが通知されます。データの読み込みが完了するとqueryCompletedデリゲートに通知されます。

queryRows

書式

-(dbLibStatus) queryRows:(NSString*) sql: (NSInteger) maxRows

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

SQLで指定されるクエリを発行して複数行をqueryRowsCompletedデリゲートに通知します。HTTPプロトコルで転送されるデータ量に上限がありますので一定のデータ量で制限したいときには第2パラメータmaxRowsに取得するデータ行の最大数を設定します。maxRowsにゼロ設定の場合は全データを取得します。

rollbackTrans

書式

-(dbfLibStatus) rollbackTrans

戻り値

Normal時は呼び出し成功。それ以外はAppendix-A参照。

解説

トランザクションをロールバックします。サーバーには接続完了状態、トランザクション状態で呼び出してください。サーバーからの実行結果はrequestCompleted/requestFailedデリゲートに通知されます。

プロパティリファレンス

PSQL Connector for iOSのプロパティについて説明をします。

lastErrorString

本製品のメソッドを呼び出してエラーが発生した場合にその理由が明確な場合はこのプロパティに保持されます。エラー情報は以下となります。

1. ネットワークエラー情報
2. サーバー側エラー情報
3. PSQLまたは.NET frameworkエラーメッセージ

pwd

IISでウェブ認証を使う場合にはこのプロパティにパスワードを設定してください。

verbose

詳細なログを出力します。

uid

IISでウェブ認証を使う場合にはこのプロパティにユーザーIDを設定してください。

url

接続先のサーバURLを指定します。（例：<http://some.server.jp/>） サーバーの構成によりSSL接続やポート番号指定も可能です。

デリゲートリファレンス

queryCompleted

書式

-(void) queryCompleted

戻り値

なし。

解説

queryメソッドが完了したときに呼び出されます。データがそろった状態なのでUITableViewのリロードなどを呼び出します。

queryRowsCompleted

書式

-(void) queryRowsCompleted:(NSMutableArray*) rows

戻り値

なし。

解説

queryRowsメソッドが完了したときに呼び出されます。パラメータにはNSMutableArrayで取得したレコード行が複数返されます。NSMutableArrayが保持するのはNSDictionaryオブジェクトでカラム名をキーとしてデータを取得します。キーのカラム名は通常大文字となります。以下サンプルコードです。

```
NSMutableDictionary* row = [_rows objectAtIndex:indexPath.row];
NSString* firstName = [row objectForKey:@"First_Name"];
NSString* lastName = [row objectForKey:@"Last_Name"];
cell.textLabel.text = [NSString stringWithFormat:@"%@@ %@", firstName, lastName];
```

requestCompleted

書式

-(void) requestCompleted:(NSString*)methodName;

戻り値

なし。

解説

メソッドの正常完了を通知するデリゲートです。クエリ系以外のメソッドでこのデリゲートに通知となります。methodNameには発行したメソッドの名前が通知されます。ただしconnectとdisconnectは通知メソッド名Login/Logoutとなります。

requestFailed

書式

-(void) requestFailed:(NSString*)methodName :(NSError*) err

戻り値

なし。

解説

メソッドの異常終了を通知するデリゲートです。クエリ系以外のメソッドでこのデリゲートに通知となります。methodNameには発行したメソッドの名前が通知されます。ただしconnectとdisconnectは通知メソッド名Login/Logoutとなります。NSErrorについてはネットワーク系エラー以外はnilが指定されます。NSErrorの詳細はAppleのマニュアルをご参照ください。

rowFetched

書式

-(bool) rowFetched:(NSMutableDictionary*) row

戻り値

Trueを返すと次のレコードを取得します。Falseを返すとこのレコードで終了となります。

解説

queryメソッド実行後にレコードを受信するたびにこのデリゲートが呼び出されます。パラメータにはレコードイメージがNSDictionary型で保持されます。キーはカラム名となります。カラム名は通常大文字になります。以下はサンプルコードです。

```
NSString* firstName = [row objectForKey:@"First_Name"];  
NSString* lastName = [row objectForKey:@"Last_Name"];
```

Appendix-A エラーコード

メソッド呼び出し時に返るdbLibStatusです。

Normal	0	正常終了
NetworkFail	1	ネットワーク接続が出来ませんでした。URLプロパティと実際のネットワーク接続状態をご確認ください。
NoURL	2	URLプロパティの指定がありませんでした。
NoUID	3	UIDプロパティの指定がありませんでした。
NoPwd	4	PWDプロパティの指定がありませんでした。
WebServiceFailed	5	Web Serviceがエラーを返しました。
WebServiceException	6	Web Serviceが例外を返しました。lastErrorTextに詳細が保持される場合がありますのでご確認ください。
SqlEmpty	7	ブレーク状態をクリアできません。
StillInRequest	8	他のリクエストが終了していないため、新たなメソッドの呼び出しが出来ません。

Appendix-B Swift サンプルコード

```
private var _count : Int = 0
private var _arr: Array<String> = []
private var _db: dbLib!

private func startConnect() {

    UIApplication.sharedApplication().networkActivityIndicatorVisible = true;

    _db = dbLib()
    _db.url = "http://192.168.0.4/"    // <== change here.
    _db.delegate = self
    _db.verbose = true

    let rc: dbLibStatus = _db.connect()

    if(rc != Normal) {
        print("connect failed \(rc)");
    }
}

private func startQuery() {
    let rc:dbLibStatus = _db.query("select ID,First_Name,Last_Name from Person order by
First_Name")
    if(rc != Normal){
        print("query request failed \(rc)")
    }
}

//
// MARK: dbLibDelegate
//

func requestCompleted(methodName:String) {

    print("request completed %@",methodName)

    if(methodName == "Login"){
        startQuery()
    }
}

func requestFailed(methodName:String, error:NSError) {

    print("request failed \(methodName) \(_db.lastErrorText)")
}
```

```

        UIApplication.sharedApplication().networkActivityIndicatorVisible = false
    }

    func rowFetched(row:NSMutableDictionary) -> Bool {

        let empno:String = row.objectForKey("ID") as! String
        let firstName:String = row.objectForKey("First_Name") as! String
        let lastName = row.objectForKey("Last_Name") as! String

        let str:String = "\(empno) \(firstName) \(lastName)"
        _arr.append(str)
        _count += 1

        return true
    }

    func queryCompleted() {

        dispatch_async(dispatch_get_main_queue()){
            self._db.disconnect()
            self.tableView.reloadData()
            self.tableView.setNeedsDisplay()
            UIApplication.sharedApplication().networkActivityIndicatorVisible = false
        }
    }
}

```

Appendix-C Objective-C サンプルコード

```
-(void) startConnect {

    [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;

    _db = [[dbLib alloc] init];
    _db.url = @"http://some.domain.jp";
    _db.delegate = self;

    dbLibStatus rc = [_db connect];

    if(rc != Normal) {
        NSLog(@"connect failed %d",rc);
        return;
    }
}

-(void) startQueryRows {

    bool rc = [_db queryRows:@"select ID,First_Name,Last_Name from Person order by
ID" :0];

    if(rc){
        NSLog(@"query rows request failed");
    }
}

-(void) queryRowsCompleted:(NSMutableArray *)rows {

    _rows = rows;

    [_table reloadData];
    [_db disconnect];
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
}
```

Appendix-D サンプル Web.config

```
<?xml version="1.0"?>
<configuration>
<appSettings/>
<connectionStrings>
<add name="psqlLib" connectionString="Driver={Pervasive ODBC Engine
Interface};ServerName=localhost;dbq=@DEMODATA;" />
</connectionStrings>
<system.web>
<compilation debug="true" strict="false" explicit="true" targetFramework="4.0" />
<authentication mode="Windows"/>
</system.web>
<system.webServer>
  <directoryBrowse enabled="true"/>
</system.webServer>
</configuration>
```

PSQL connector for iOS

第1版

2016年7月15日

版權・著作 株式会社テクナレッジ
Printed In Japan