

CodeIgniter Con 2011, Tokyo Japan, February 19

CodeIgniterで使う FileMaker Server

新居雅行 nii@msyk.net

 Masayuki Nii  msyknii



Certified System
Administrator



Certified Trainer
Mac OS X Server Essentials 10.6



FileMaker 9
Certified Developer

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
Professional

Who am I ? (Masayuki Nii)

CodeIgniterでの開発実績：4件

- いずれもFileMakerからみ

執筆書籍

- FileMaker Server大全
- リレーションで極めるFileMaker、など

ITコンサルタント

- Apple認定システムアドミニストレータ
- Apple認定トレーナー
- Microsoft認定プロフェッショナル
- Microsoft認定トレーナー

今回のプレゼンは、<http://msyk.net/fmp>



Agenda

FileMakerおよびFileMaker Serverについて

FileMakerデータベースの市場的特徴

FileMakerをWeb公開する手法

FX.phpについて

改造版FX.phpについて

CodeIgniterでFileMaker Serverを使う手法

Principle

FileMaker系の方々

- このセッションの前半はよくご存知のことでしょう
- PHPやCodeIgniterを理解するために、かなりお疲れでしょう
- しばらくは、仮眠あるいは休憩しててください
- 核心に入ったら起こします

FileMaker系でない方々

- FileMakerの良さを知ってください
- Web開発の選択肢の1つにご検討ください
- 後半はFileMakerを詳細に知った上での話になり、いきなり訳が分からないかもしれませんが、そんなもんかと思って聞いてください

What is FileMaker?



FileMakerについて

古くは「カード型データベース」の代表格

- 現在は、複数のテーブルを連携させるリレーショナルデータベースにも対応している
- ダブルクリックして起動する「アプリケーションソフト」

1987年に開発がスタート、現在もリリース継続中

- 早くにクラリスが製品を買収、現在はApple Inc.の子会社であるFileMaker Inc.が開発・販売
- Macオンリーから、WindowsとMacでの利用

有償のパッケージソフトウェア

- 昔は1ライセンス、10万を超えた製品
- 現在は、FileMaker Pro 11が4万前後で販売されている

現在のクライアント向け製品

FileMaker 11シリーズ

デスクトップアプリケーション

- FileMaker Pro
- FileMaker Advanced
- →Proの機能にスクリプトデバッガなどを付加した開発者用
- それぞれ、Mac OS X用、Windows用がある

iOS対応デバイス向け

- FileMaker Go
- 作成したデータベースの利用をiPhoneやiPadでできる

テーブル定義

「資源計画」のデータベースの管理

テーブル フィールド リレーションシップ

テーブル: リソース計画 13 フィールド 表示順: カスタム順位

フィールド名	タイプ	オプション/コメント (クリックして切り替え)
✦ 作成者名	テキスト	作成名
✦ 作成日	日付	作成日, 値変更不可
✦ 修正日	日付	修正日, 値変更不可
✦ グローバルリソース番号	テキスト	グローバル
✦ リソース名ソートマーク表示	計算	非保存, from リソース計画, = Case(ソート条件設定 = "リソース名昇順"; GetRepet...
✦ ソートマーク	オブジェクト [2]	グローバル
✦ ソート条件設定	テキスト	グローバル
✦ タイプソートマーク表示	計算	非保存, from リソース計画, = Case(ソート条件設定 = "タイプ昇順"; GetRepetitio...
✦ リソース名	テキスト	
✦ 備考	テキスト	
✦ ナンバーワン	計算	from リソース計画, = 1
✦ リソース番号	テキスト	索引設定済, 番号自動入力, 値変更不可
✦ タイプ	テキスト	

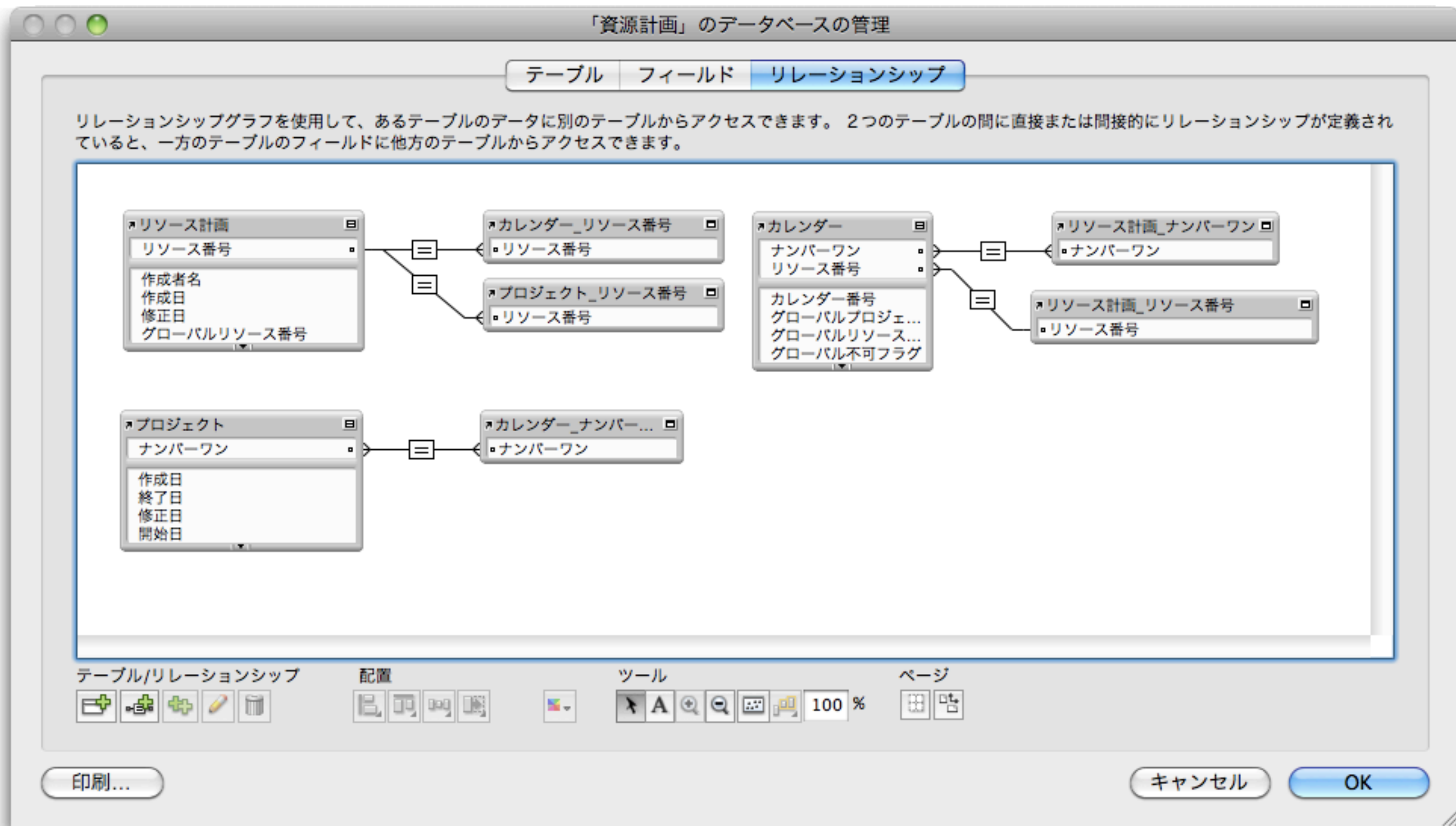
フィールド名: タイプ: テキスト オプション...

コメント:

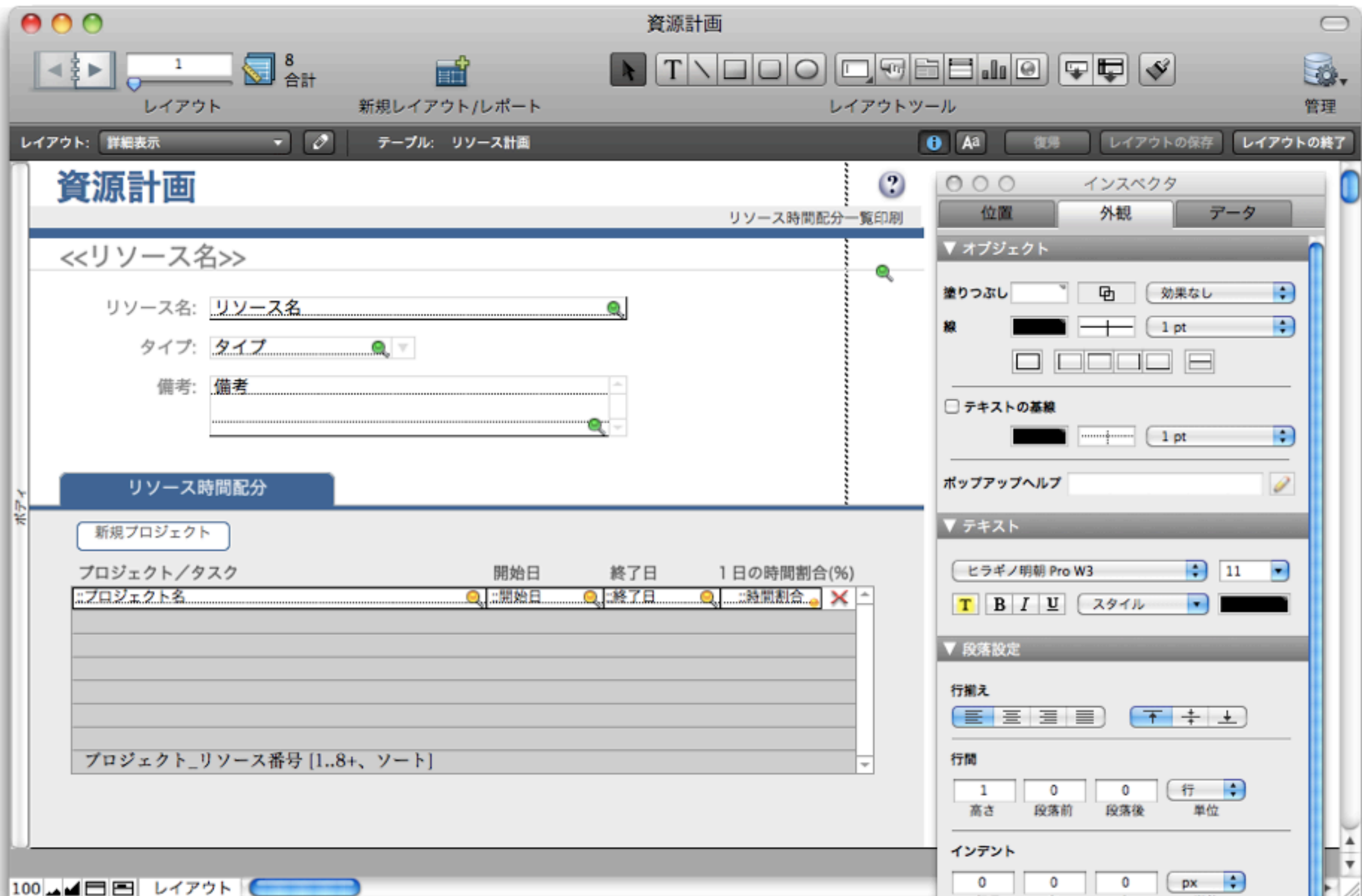
作成 変更 複製 削除

印刷... キャンセル OK

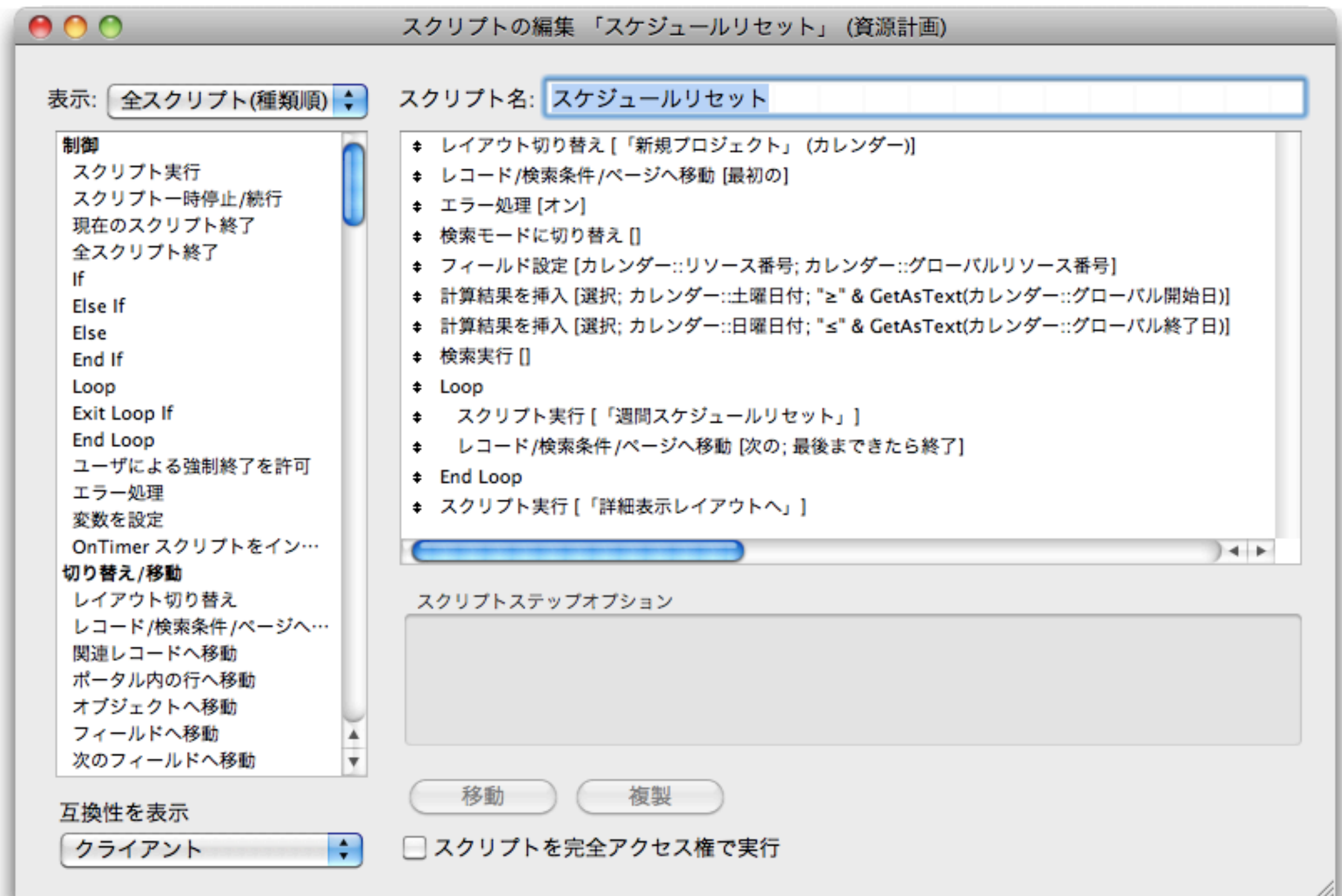
リレーションの設定



レイアウト



スクリプト



FileMakerを使う理由

データベースを利用したソリューション

- 「テーブル」を「レイアウト」に表示するという分かりやすい仕組みでソリューション（業務アプリケーション）構築可能
- 「スクリプト」により複雑な処理の組み込みも可能

簡単かつ修正が容易

- 少ない作業である程度のものが作成できるため、とっつきが良い
- レイアウトという「目に見えるもの」の編集で帳票やユーザインタフェースを作り込むため、見通しが良い
- テーブルに計算フィールドを定義できることや、テーブルの設計変更が後からできるため、きちんと設計しなくても開発に取りかけられる

FileMakerで失敗する例

正しくないスキーマでも何か作れてしまう

- 不適切なテーブル定義
- 無計画なスキーマ拡張／計算フィールドの多用
- テーブル1つで全部！
- 結果的に、改修が大変しづらい、ちゃんと動いていない、直しようがない、何とも言いようがないデータベースが何と業務利用されてしまう

パフォーマンスが悪い

- パフォーマンスを悪く作ることができてしまう
- 「正しくないスキーマ」が結果的にパフォーマンスの低下を導く

What is FileMaker Server?



FileMaker Server

ファイルメーカーのデータベースを共有

- デスクトップアプリの「書類」をサーバで共有
- FileMaker Proで開いて利用できる
- ODBCによりMySQLなどのソースをクライアントに提供

Webアプリケーション

- IWP (Instant Web Publishing) = Advanced版のみ
- PHP Site Assistant
- FileMaker API for PHP / PHPでのサイト構築

ネットワーク共有での必然性

- デスクトップアプリのFileMaker Proでも共有はできるが、パフォーマンスや同時接続数が限られる
- バックアップなどの管理を含めて、サーバを利用するのがベター

FileMakerを使ったWebアプリの利点

なぜ、FileMakerでWebなのか？

Webを利用する利点

- Webの部分と、FileMakerで作る部分を混在できる
- 参照するだけ、あるいは一般多数に公開したい内容をWebで作成
- 印刷が必要なものや管理、編集が必要なデータをFileMakerで扱う
- こうした「ハイブリッド」ができる点がメリット

FileMakerの開発は早く簡単にできる

複雑なユーザインタフェースはWebの方が作りやすい

とは言え…

FileMaker ServerはSQLサーバではない

- ORマッピング系のフレームワークには相手にしてもらえなかった
- さまざまなWebアプリケーションフレームワークでは対応してもらえなかった
- ODBC対応しているものの、ドライバによるSQLのエミュレーションと言えるものなので、あまり使われなかった

過去は、独自言語のCDML、XSLによる公開機能の時代

- 独自性が強すぎ…

FileMaker ServerのPHP対応やMVCフレームワークの登場により、「接点」ができてきた

FileMaker業界と Webアプリケーション開発

FileMakerの開発者

FileMaker Business Alliance

- FileMaker開発者を組織化
- FileMaker社のサイトにディレクトリ
- FileMakerをビジネスにするには加入は必須

FileMaker TechNet

- 技術者向けの情報提供
- FileMaker Serverの試用ができる

インハウスデベロッパー

その他のデベロッパー

- 個人事業者

FileMaker業界からみた客層

もちろん、「普通の仕事」「まともな仕事」もあるが…

偏った(笑)客層が目立つ

- ものすごいFileMakerファン（ついでにMacファン…）
- 管理者や開発者がいない組織なのになぜかソリューションが作られている

かくして、こんな仕事が発生する(一例)

- データベースを作ったやつが、何年か前に退社してしまい、管理パスワードすら分らない。だましだまし使ってきたが、いよいよ改修しないと業務が進まない
- →意味の分らないスキーマ
- →何のためにあるのか分らないスクリプト
- →顧客は「中身は何も分らない」の一点張り

こんな仕事が、またまた発生する

そうした中でたまにWeb開発の要望が出る

古い時代に作ったWebサイト

- 古い時代（CDMLというマークアップ言語があった）に作ったものが存在している
- CDMLだと最近のFileMaker Serverで動かない
- 元があっても明らかにスクラッチから作った方が効率的なのに…

管理者不在のソリューション

- とにかく作って使っている（動いてはいる）
- ライセンスを追加できないので、Webで参照だけでもしたい
- スキーマを見て唖然とするが、力技でWeb構築した
- パフォーマンスが出ない…

FileMaker Web化のジレンマ

Webに活路を見出すものの…

IWPだとFileMakerそのままWeb化できる

- デスクトップアプリケーションで構築したテーブル、レイアウト、スクリプトが、ほぼすべてそのままWebブラウザ上に展開される
- 一見すると、そのままWebに行けるかに見える

しかし、すべてが同じわけではない

- フィールドデータの確定、レコードのロックなど、違いは大きい

違いを知った上で作り込めば実用的に使える

- おっと、最初の前提が違うぞ

…無限ループ…

PHPで作れるらしい

PHPまで勉強するのは大変、でも…

- PHP Site Assistantってのがあって、ソースを自動生成してくれるらしい
- それなりに便利で、実用性も高いのだが、パターンからはずれた仕組みを作るのはかなり面倒（というか、かえって大変）

スクラッチでいくぜー

- FileMaker API for PHP、なんか遅い
- あまりにも基本的杉（要はクエリーや更新処理だけで、アプリケーション作成支援にはほど遠い）

どう解決するか？

現状で、FileMakerベースのWeb開発が発生！

FileMaker標準のライブラリでは遅い

- FX.phpというライブラリを使う（すぐ後で紹介）
- しかし、それでも基本的なクエリーと更新のみ

アプリケーション化にはフレームワークが必要

- モデル化については考慮しないでもいい（後述）
- となると、VCの部分が分かりやすいのがいいよね
- PHPで作るのが無難か？（Javaとかはつなぎの部分で苦勞しそう）
- 階層化により無茶な作りをラッピングできる！？…カナ

CodeIgniterを使おう！！

- CakePHP（w/FMCakeMix）、
Ruby on Rails（w/Rfm）もメジャーな選択肢

FX.phpについて



<http://www.iviking.org/FX.php/>

FX.phpについて

FileMaker ServerのXML出力を利用したライブラリ

- 2002年ごろ、FileMaker自身がPHPに対応していない時代に、Chris Hansenがリリース、オープンソース
- 2004年、新居が日本語の入出力化に成功
- 2007年、FileMakerがPHPのライブラリを出した後でもパフォーマンスが高いために使い続けられている

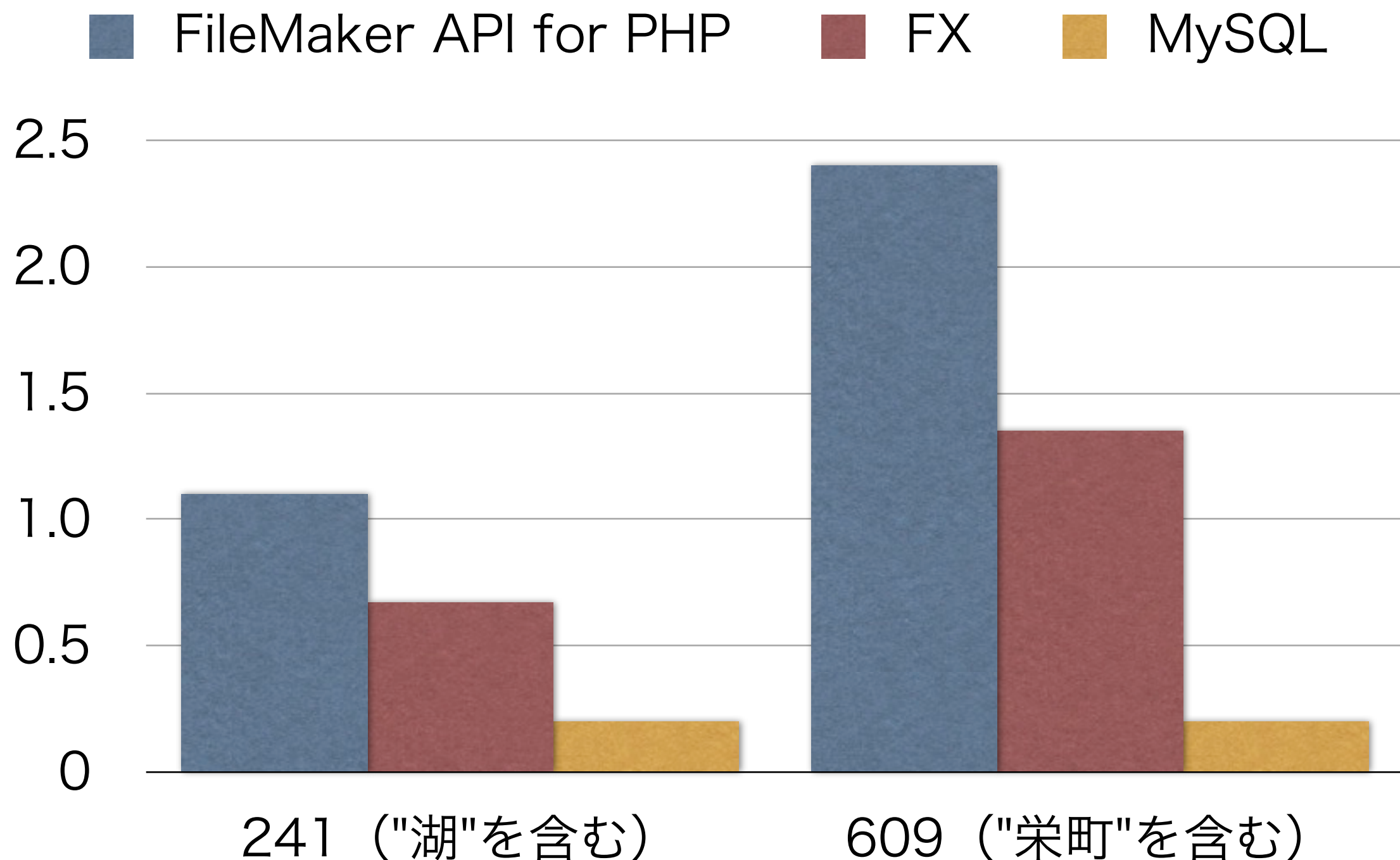
Pros/Cons

- 早い。FileMaker APIと大差ない
- 機能は劣る面があるが、複雑な検索ができないなどの非常に小さなこと

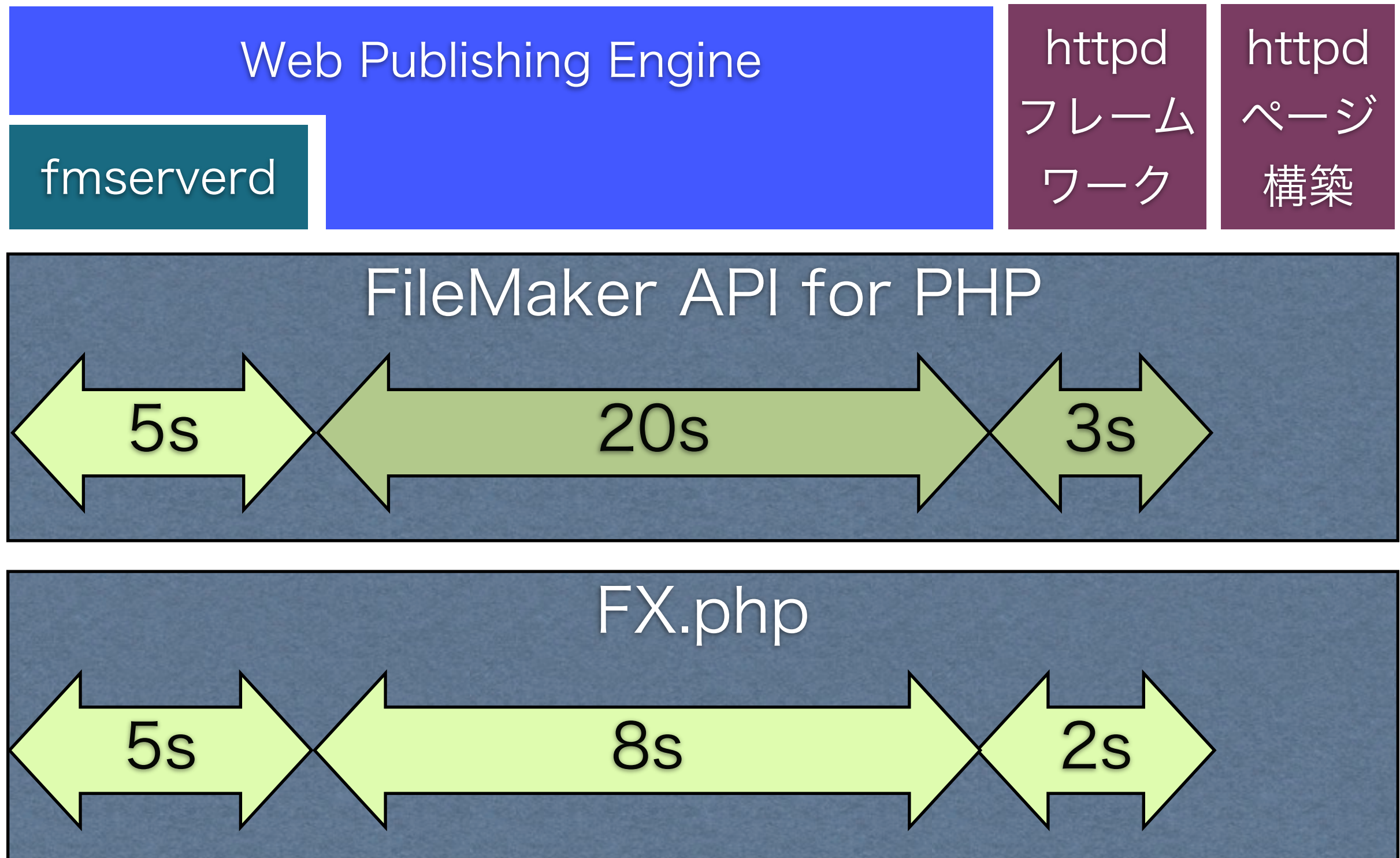


FX.phpのパフォーマンス

10万件の郵便番号データベースからの検索時間



FileMaker Server内部での動作



公開するデータベース

XML共有をアクティブにしたユーザを用意
FileMaker Server上にデータベースを配置



CodeIgniterでFileMaker Serverを利用する



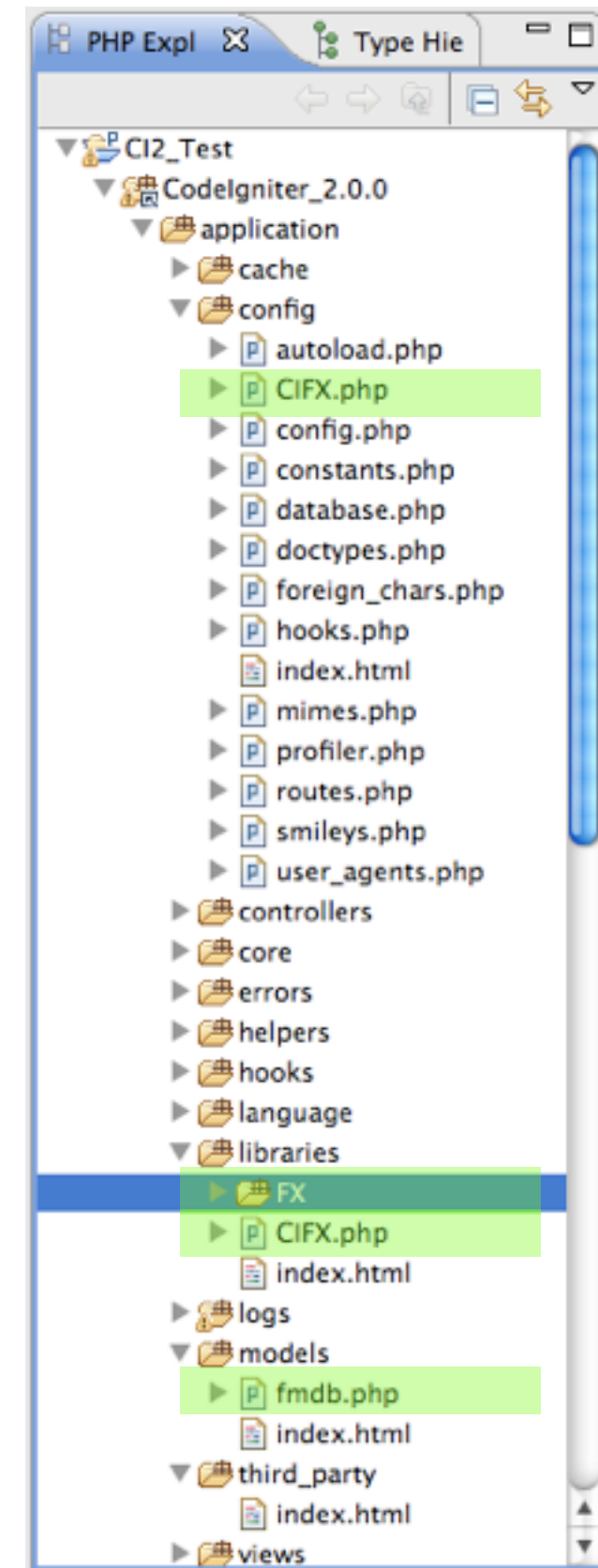
CodeIgniterにFX.phpを組み込む

基本的な手法は本家のWikiに掲載

- http://codeigniter.com/wiki/FilemakerPro_Database_Integration

インストール

- 通常とおりCodeIgniterをインストール
- application/librariesに、FXフォルダを配置 (FX/FX.phpとなるように配置)
- application/librariesに、CIFX.phpを作成
- application/configに、CIFX.phpを作成
- CIFXを使うようにモデルを記述する
- 後はほかと同じ



application/config/ClFX.php

<?php

```
$config[ 'dataServer' ] = '127.0.0.1';
```

```
$config[ 'dataPort' ] = '80';
```

```
$config[ 'dataType' ] = 'FMPro7';
```

```
$config[ 'dataURLType' ] = 'http';
```

```
$config[ 'database' ] = 'TestDB';
```

```
$config[ 'dbuser' ] = 'web';
```

```
$config[ 'dbpassword' ] = 'password';
```

?>

application/libraries/CIFX.php

```
<?php
if( ! defined( 'BASEPATH' )) exit( 'No direct script access allowed' );

require_once( 'FX/FX.php' );

class CIFX extends FX {
    function CIFX ( $params ) {
        parent::FX( $params[ 'dataServer' ], $params[ 'dataPort' ],
                    $params[ 'dataType' ], $params[ 'dataURLType' ] );

        $this->setDBUserPass( $params[ 'dbuser' ], $params[ 'dbpassword' ] );
        $this->initializeFX();
        $this->database = $params[ 'database' ];
    }

    var $database;
    var $fxresult;

    function initializeFX() {
        $this->setCharacterEncoding( 'UTF-8' );
    }
    function initializeFXWithShiftJIS() {
        $this->setCharacterEncoding( 'sjis' );
    }
}
```

CIFXクラスでのアクセス関数例

```
class CIFX extends FX {
    :
    function queryFromDB( $layout, $criteria=array(), $sort=array(), $skip=0, $limit=10000000000, $grouping=null) {
        $this->SetDBData( $this->database, $layout, $limit );
        $this->FMSkipRecords( $skip );
        foreach( $criteria as $field=>$value ) {
            if ( is_array( $value ) ) {
                $this->AddDBParam( $field, $value['value'], $value['op'] );
            } else {
                $this->AddDBParam( $field, $value, 'cn' );
            }
        }
        $c = 1;
        foreach( $sort as $field=>$value ) {
            $this->AddSortParam( $field, $value, $c );
            $c++;
        }
        if ( $grouping != null ) {
            switch( count($grouping) ) {
                case 1: $this->RemainAsArray( $grouping[0] ); break;
                case 2: $this->RemainAsArray( $grouping[0], $grouping[1] ); break;
                // and so on.
                default: $this->RemainAsArray( ); break;
            }
            $this->FlattenInnerArray();
            $this->fxresult = $this->DoFxAction( FX_ACTION_FIND, TRUE, false, 'full' );
        } else {
            $this->fxresult = $this->DoFxAction( FX_ACTION_FIND, TRUE, TRUE, 'full' );
        }
        if ( $this->fxresult['errorCode'] != 0 ) {
            return array();
        } else {
            return $this->fxresult['data'];
        }
    }
}
```

application/models/*.php

```
<?php
```

```
class FMDB extends CI_Model {
    function __construct(){
        parent::__construct();
    }
    function getPerson() {
        $this->load->library('CIFX');
        $result = $this->cifx->queryFromDB( 'person_layout',
            array(), array(), 0, 100,
            array( array('contact_to', 'contact_way_person',
                'contact_kind_person'), 'history_to') );
        if ( $this->cifx->errorCode() != 0 ) {
            return "Error:";
        } else {
            return $result;
        }
    }
}
```

通常のFX.phpの出力

```
$result["data"]["13.4"]["field_name"][0]
```

- 結果は配列に残される
- 1つ目の添字から取得データやエラーコードなどを取り出す
- 2つ目でレコードを指定するが、レコードIDと修正数を組にした文字列が返される
- 3つ目の添字がフィールド名
- 4つ目の添字はフィールドのデータが1つでも必要
- ポータルや繰り返しフィールドの場合はデータの行数分の要素があるが、単一のフィールドもこの形式

ほかの形式も可能

- 最後の添字のない出力もあるが、繰り返しフィールドやポータルの結果は最初の項目だけになる

FX.phpの出力結果

```
array (
  '1.51' =>
  array (
    '-recid' => 1,
    '-modid' => '51',
    :
    'name' => array ( 0 => 'Masayuki Nii',),
    'address' => array ( 0 => 'Saitama2',),
    'mail' => array ( 0 => 'msyk@msyk.netss',),
    'contact_to::id' => array ( 0 => '43', 1 => '126', 2 => '158',),
    'contact_to::person_id' => array ( 0 => '1', 1 => '1', 2 => '1',),
    :
  )
)
```

TestDB (msyk-macbookair)

レコード: 1, 3 合計 (未ソート)

すべてを表示 新規レコード レコード削除 検索 ソート

レイアウト: person_layout 表示方法の切り替え: プレビュー

id	person_id	summary	important	way
43	1	Send Mail222233	2009/01/01 3:03:04	Send Mail6 and 1 Indirect
126	1	Meet at town go to	2000/01/01 0:00	qsss 1 Direct
158	1	Cocoa勉強会	2010/02/11 13:00	Direct
3	1	2002/01/02	2003/03/31	company3
4	1	2003/04/02	2006/03/01	School345
31	1			追加しました

person_id: 1

name: Masayuki Nii

address: Saitama2

mail: msyk@msyk.netss

category: 103

check: 1

location: 202

memo: He is Japanese. He is Japanese. He is

問題点と解決方法

さまざまな問題点

サロゲートペアのUnicode文字の問題

- 読み書きいずれも問題が発生
- 出力時は2バイトずつ別々にUTF-8にしている
- 入力時はサロゲートペアの数だけ文字が後ろから消える
- FX.phpについてはパッチ済み

単一のフィールドでも[0]が必要な問題

- 改造済み／この後説明

日本語のフィールドの問題

- フィールド名、データベース名、テーブル名(TO名)、レイアウト名、グローバルフィールド名は問題ない
- スクリプト名に日本語が使えない
- FileMaker Server側の問題でタッチできない

改造版のFX.php

FX.phpのレポジトリに投稿

- <https://sourceforge.net/projects/fxphp/>

変更点

- サロゲートペアの問題に対応
- [0]を含まない出力結果を得る仕組みを提供
- フィールド名をオブジェクトのプロパティのように扱える出力結果を得る仕組みを提供

サロゲートペアの修正

- もし、FileMaker Serverでなおったら？
- クエリー結果は問題なく処理できるはず
- 更新等の入力では、以下の記述でパッチを無効にするとOKなはず
`define( 'SURROGATE_INPUT_PATCH_DISABLED', true);`

FX.phpの出力形式を変更する

FX.phpの標準形式

```
foreach ( $result as $oneRecord ) {  
    echo "<li>{$oneRecord['f3']}: "  
        . "{$oneRecord['f8']}{$oneRecord['f9']}</li>";  
}
```

[0]を出力しない形式

```
foreach ( $result as $oneRecord ) {  
    echo "<li>{$oneRecord['f3']}: "  
        . "{$oneRecord['f8']}{$oneRecord['f9']}</li>";  
}
```

フィールド名をプロパティとして使う形式

```
foreach ( $result as $oneRecord ) {  
    echo "<li>{$oneRecord->f3}: "  
        . "{$oneRecord->f8}{$oneRecord->f9}</li>";  
}
```

FX.phpの出力形態を変えた

TestDB (msyk-macbookair)

レコード: 1, 3 合計 (未ソート)

すべてを表示, 新規レコード, レコード削除, 検索, ソート

レイアウト: person_layout, 表示方法の切り替え: [Table View], プレビュー

id	person_id	summary	important	way	kind
43	1	Send Mail222233	2009/01/01 3:03:04	Send Mail6 and	1
126	1	Meet at town go to	2000/01/01 0:00	qsss	1
158	1	Cocoa勉強会	2010/02/11 13:00		

id 1, name Masayuki Nii, address Saitama2, mail msyk@msyk.netss, category 103, check 1, location 202, memo He is Japanese. He is Japanese. He is

```
array (
  '1.51' =>
  array (
    '-recid' => '1',
    '-modid' => '51',
    'id' => '1',
    'name' => 'Masayuki Nii',
    'address' => 'Saitama2',
    'mail' => 'msyk@msyk.netss',
```

```
'contact_to' =>
array (
  0 =>
  array (
    'contact_to::id' => '43',
    'contact_to::person_id' => '1',
    :
    'contact_way_person::name' => 'Indirect',
  ),
  1 =>
  array (
    'contact_to::id' => '126',
    'contact_to::person_id' => '1',
    :
    'contact_way_person::name' => 'Direct',
  ),
```

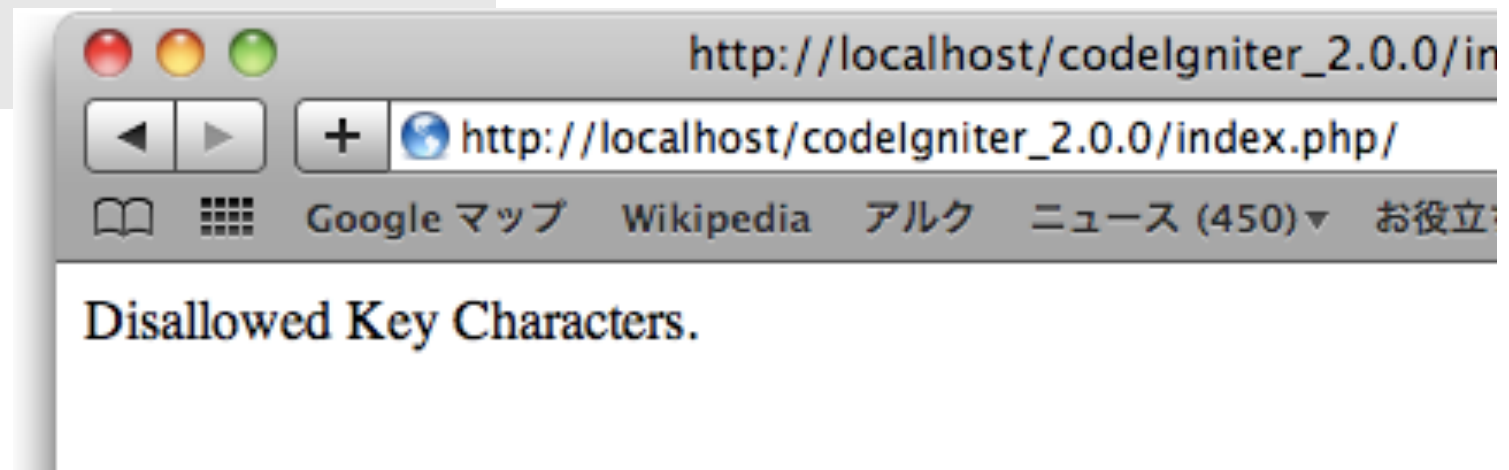
日本語を含むフォームの問題

普通は避けるとしても、それが「前提」の場合もある

- 既存のWebページを更新する場合
- 日本語のフィールド名を使っている場合

そのままだと、Postしたときにエラーになる

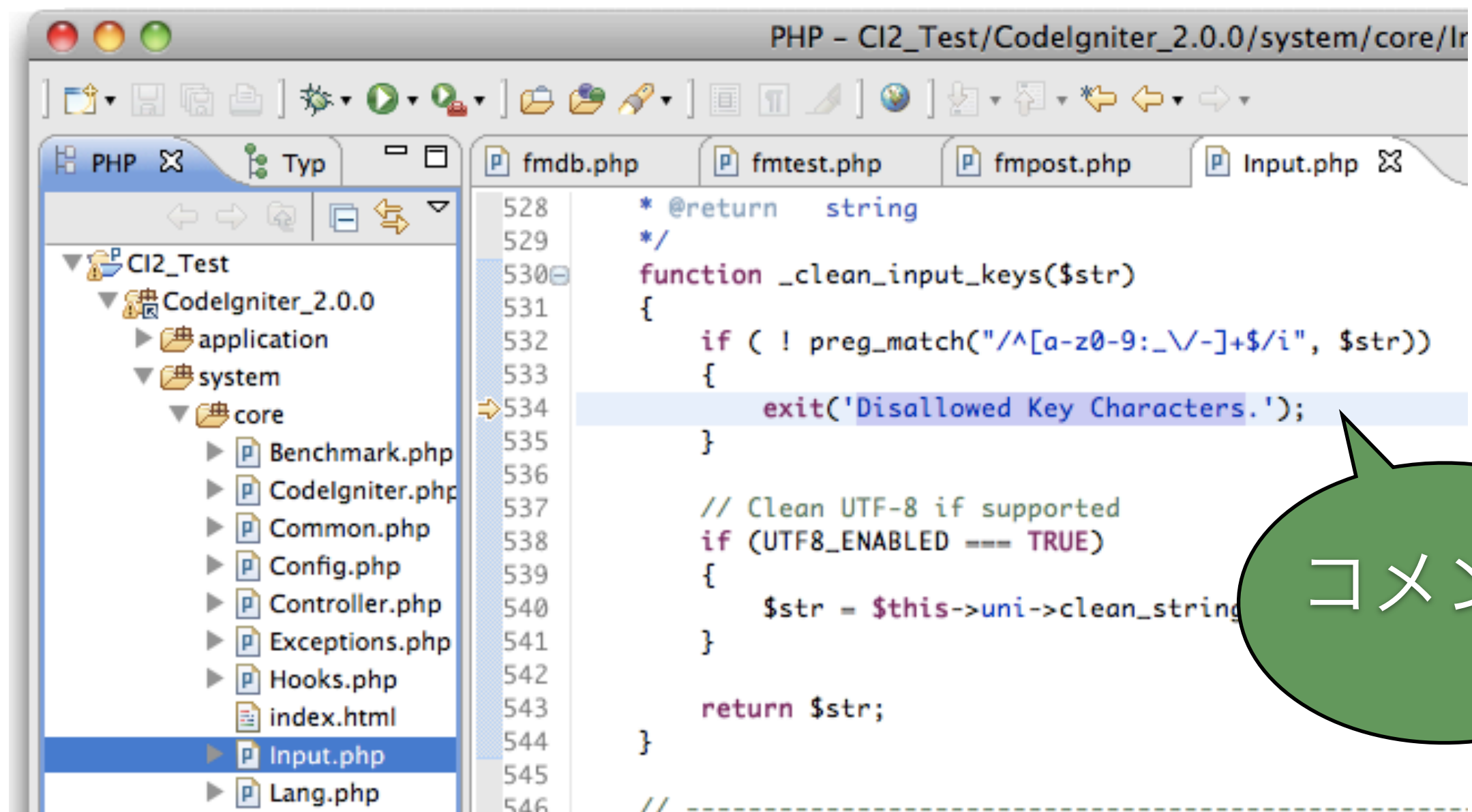
```
<form action="." method="post">  
<input name="町域名" type="text" />  
<button type="submit">検索</button>  
</form>
```



日本語を含むフォームの問題の解決策

CodeIgniterのソースの一部を変えることで対処可能

- ただし、他の場面に影響しないかは確認しよう



FileMakerのWebでスピードを稼ぐ

検索スピードは、数値フィールド > 文字列フィールド

- FileMaker社資料によると、数倍の違いがあるらしい

従って、なるべく、数値フィールドで検索する

- 自動連番でのプライマリキーフィールドの生成
- 数値フィールドでのリレーション
- なるべく、プライマリキー等での検索
- 自由にデータベースを作るのではなく、データベースの設計理論に基づいた設計をすれば不可能ではないはず

検索結果をなるべくコンパクトにする

レイアウトに余分なフィールドを配置しない

- フィールドの数だけXML生成するため、線形的に時間が増える

FileMaker開発者の方々へ

CodeIgniter+FX.phpでWebソリューションを作ろう

- スピードを少しでも損なわないために
- 学習コストの低さはFileMakerを使う理由でもあります

エディタを使って開発するのはやめよう

- 凡ミスの発見に時間やエネルギーを使うのはやめよう
- Macでもeclipseは十分に使えます

ネットワークやWebサービスについて学習しよう

- ApacheもIISも、ある意味、FileMakerほど簡単ではありません
- お客さんに満足してもらえる結果をもたらす知識を付けましょう

まとめ

FileMakerのソリューションをWeb展開する

FX.php+CodeIgniterを使って高速化を計る

FX.phpの改造版ではさまざまな機能を提供

FileMakerのWeb展開でのさまざまなノウハウ

[付録] FX.phpの修正に関する情報

FX.php, Modifying to prevent [0] trailing for single value fields

- <https://msyk.net/users/msyk/weblog/b31fb/>

FX.php modification, multiple-TO in a same portal

- <https://msyk.net/users/msyk/weblog/e9a37/>

FX.php modification, Objective Reference for the result of query

- <https://msyk.net/users/msyk/weblog/cb14c/>