



INTER-Mediatorで “FileMakerのように” Webサイトを作る

新居雅行

Masayuki Nii

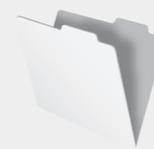
FFN #92 - 2013/3/23



Certified
System Administrator 10.6



Certified Trainer
OS X Support Essentials 10.8



FileMaker[®] 11
CERTIFIED DEVELOPER

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
Professional

Who is the presenter ?

 Masayuki Nii
  msyknii

IT Professional

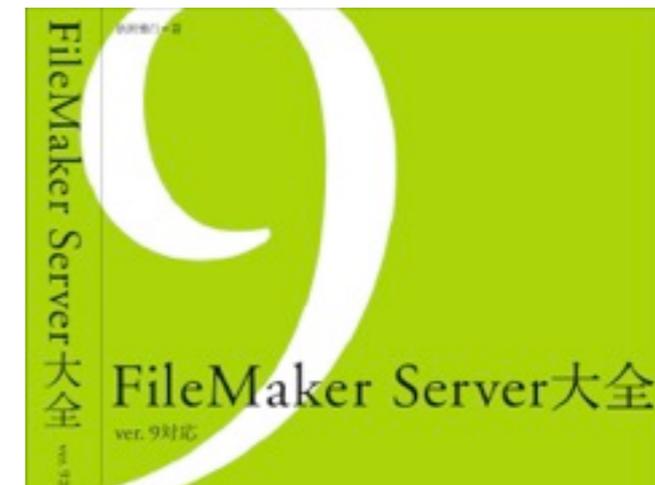
- アップル認定システムアドミニストレータ
- アップル認定トレーナー
- マイクロソフト認定プロフェッショナル
- マイクロソフト認定トレーナー

Developer

- Web Framework “INTER-Mediator”
- iOS Development and Training
- FileMaker 11 Certified Developer

FileMaker community

- A committer of FX.php
- 『リレーションで極めるFileMaker』などの著書



INTER-Mediatorとは

- Webアプリケーションフレームワーク
- FileMaker Server対応、FMソリューションと連携可能
- FileMaker的な簡単さで、カスタムWebサイトの構築が可能

INTER-Mediatorのこの3年間

最新版での開発デモと開発手法

INTER-Mediatorの今後の3年間

Last Three Years

Webアプリケーションのメリットは大きい

- ネイティブもいいが、ネットワーク、シンプル、汎用性、いろいろな意味で、Webアプリは有利
- システム開発のベースにしたい

Webアプリを効率的に作りたい

- プログラミングすることが一般的だが、作るものが変わっても、同じようなプログラムをまいどまいど作っていた
- 単純作業をフレームワークに押し込め、最短距離でWebアプリを作りたい

言語の混在はクールではない

- HTMLと独自のテンプレート言語、PHPの中に書くJavaScriptなど、Webの世界はなかなかワイルド
- だが、その結果、メンテナンス性は低いプログラムを作りがち

データベースとHTML要素をダイレクトに結合

- 何もしなくても、データベースのデータがページ上に展開
- 入力したデータを、何もしなくてもデータベースに書き込み
- シンプルなページはプログラミング不要に

必要な機能を追加する仕組みを提供

- 展開したページ上での処理をJavaScriptで記述
- データベースから取り出した結果を処理してクライアントに送信
- 複雑なデータ処理もできるようにする

言語を混在させない、新たな言語の創造をしない

- HTMLで記述するページは、HTML以外の要素を加えない

2009年末、突然思い付き、正月前後に限定リリース

- FileMaker Serverよりスタートし、MySQLにも対応する

FM-Tokyo : Webアプリケーション開発を根本から変える
INTER-Mediator (2/7)

FFN #59 : カスタムWebための新スタイルのフレームワーク
「INTER-Mediator」 (5/15)

スクラッチから作り直し…

- PDO対応など
- Ver.0.6から0.6.1まで半年かかってしまった

2011年



Ver.0.6として新アーキテクチャのリリースを継続

- マルチデータベース、マルチブラウザ対応

Ver.0.7としてさらに改良

- リレーションシップに関連したページの部分更新のサポート

Ver.1.0をリリース (10/19)

<http://inter-mediator.info> でサイトを開始

関西オープンソース2011に出展 (11/11)

2012年



Ver.1.0.xのリリース

- 認証とアクセス権のサポート

有償のコースウェアを公開

DevCon 2012, Custom Web Users Group

Ver.2.0系列をリリース

- サーバサイド拡張の改良、バリデーションのサポート、画像などのメディア

Ver.3.0をリリース (12/10)

断続的にイベントを開催

- コワーキングスペースを利用したJelly
- 開発状況を報告するMeet-up
- Code Reading

How Does It Work? with Some Demos

基本的な開発手順

データベースを用意する

定義ファイルを作成してサーバに登録

- 接続先のデータベースや、検索条件、ソート条件、テーブル間の連携（外部キーと対応フィールド）などを指定する

もとになるHTMLページを作る

- ヘッダ部のscriptタグで定義ファイルを読み込む
- 要素のclass属性（あるいはtitle属性）に、リンク設定を記述する
- body要素のonload属性に、INTERMediator.construct(true);と記述する

リンクノードの設定

- class="IM[tableName@fieldName@target]
- ノードテキストだけでなく、要素やスタイル、追記も対応
- innerHTMLへの設定も可能

1つのページの基本構成

Page File

```
<input type="text"  
class="IM[table@field]" />
```

HTML



```
<input type="text"  
class="IM[table@field]"  
value="value" />
```

HTML

Modified Page File

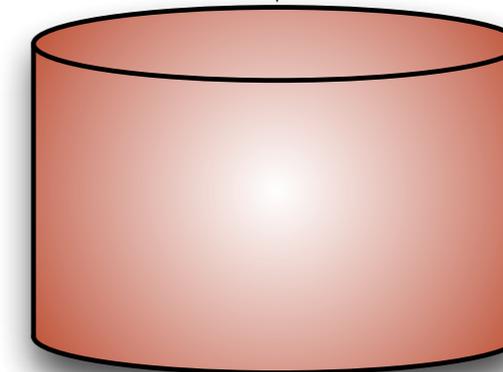
Definition File

```
[ { name=>'table',  
key=>'id', ... },  
{...}, ... ]
```

PHP

Referencing with
"SCRIPT" tag

Call the templating
method



Database

Demo 1

- * データベースにある郵便番号をページに表示する
- * ページング処理ができる
- * 新規レコードや削除
- * テキストフィールドで書き直せば、データベース側も更新される

クライアントサイドでテンプレート処理

- 純粋なHTMLで記述したページを用意
- データベースから取り出したデータがページに埋め込まれる

たとえば、テキストフィールド

- タグ : `<input type="text" class="IM[addressbook@name]" />`
- 上記の記述により、テキストフィールドが画面に出る
- addressbookテーブルのnameフィールドの値が表示される
- テキストフィールドのデータを変更すると、変更結果が元のレコードのnameフィールドに書き戻される

その他の機能

- 属性への設定や、値の書き直しなど
- バリデーション

テンプレート処理の基本



エンクロージャ／リピータ

- リンクノードとその上位ノードをたどり、リピータとエンクロージャを決定する

| | エンクロージャ | リピータ |
|--------|----------------------|---------------------|
| テーブル | tbody | tr |
| 汎用 | div [_im_enclosure] | div [_im_repeater] |
| 汎用 | span [_im_enclosure] | span [_im_repeater] |
| 番号リスト | ol | li |
| 箇条書き | ul | li |
| ポップアップ | select | option |

定義ファイルの構成

コンテキスト

- データベースの利用設定を複数指定できる
- ビューやテーブルだが、検索条件なども加わる
- 一部はJavaScriptでダイナミックに変更可能

オプション設定

- フォーマット処理、認証に関する処理など

データベース設定

- データベースクラスや、接続のための設定
- すべてのコンテキストで同じデータベースを利用する場合に指定

デバッグ指定

```

<?php
/*...*/
require_once ('../INTER-Mediator/INTER-Mediator.php');

IM_Entry(
    array(
        array(
            'records' => 1,
            'paging' => true,
            'name' => 'person_layout',
            'key' => 'id',
            'repeat-control' => 'confirm-delete confirm-insert',
            'query' => array( /* array( 'field'=>'id', 'value'=>'5', 'ope
            'relation' => array(
                array('field' => 'id', 'direction' => 'ascend'
            ),
        ),
        array(
            'name' => 'contact_to',
            'key' => 'id',
            'repeat-control' => 'confirm-delete insert',
            'relation' => array(
                array('foreign-key' => 'person_id', 'join-field' => 'id',
            ),
        ),
        array(
            'name' => 'contact_way',
            'key' => 'id',
        ),
        array(
            'name' => 'cor_way_kind',
            'key' => 'id',
            'relation' => array(
                array('foreign-key' => 'way_id', 'join-field' => 'way',
            ),
        ),
        array(
            'name' => 'history_to',
            'key' => 'id',
        ),
    ),
);

```

Demo 2

- * リレーションシップを設定して、ページの展開時に関連したレコードを別のテーブルから取り出して合成する
- * リピータの中にエンクロージャがあれば、リレーションシップの設定を元に関連レコードに絞り込む

ページ送りナビゲーションの自動生成

1対多の展開、条件付き展開なども可能

- 伝票形式のページ作成が可能
- あるポップアップの値に応じて別のポップアップの選択肢を変える
- 独立した複数の展開も可能

レコードの追加や削除のボタンを自動生成

- ページ送りナビゲーション上に表示
- 繰り返したレコードのそれぞれに「削除」ボタン

認証、アクセス許可

Demo 3

- * 入力専用フォームの作成
- * JavaScriptコンポーネント

さまざまな機能

新規レコード作成専用ページ

- エンクロージャーにいくつかの決められたコンポーネントを用意
- オプションボタンセットに対応する機能をサポート
- ボタンをクリックしてからの処理を指定可能

JavaScriptコンポーネントの利用

- tinyMCE (HTMLエディタ) に対応
- ファイルアップロードをオリジナルで作成
- 任意のコンポーネントに対応可能な仕組みを持つ

Demo 4

- * IMCake
- * 松尾篤さん（株式会社エミック）

System Integration with INTER-Mediator

対応データベース

- FileMaker Server、
- PDO対応データベース (MySQL、PostgreSQL、SQLite)

対応サーバ

- PHP5.2が動くサーバ (Apache/IIS)

対応ブラウザ

- HTML5対応 (もう少し前のものも動くはず)
- Firefox、Chrome、Safari、Opera
- IE8、IE9 (IE7は一部制限あり、IE6は無視！)
- スマートフォンOK、フィーチャーフォン対応予定全くなし

MVCフレームワークが前提

- プログラムで自由に開発ができるが、専門家以外は手を出せない
- 予算が潤沢にあるような基幹的システム開発か、なにもしないかの変な二極化の傾向が出てきた

スマホ&HTML5

- JavaScriptでかなり大掛かりに作成でき、十分なパフォーマンス
- 携帯デバイスは非力というのは昔話

サービスやインフラの充実

- 回線、サーバ、サービスが安価に利用できる
- いろいろな制約が絡んでくる

INTER-Mediatorを使えば



FileMakerのソリューションと組み合わせる

- 社内の処理はFileMaker、外部公開はWeb

エンドユーザーによるメンテナンスも視野に！

- プログラムを絡ませずに修正できる個所が広い

制約は足かせではなく、正しい方向へ向かうためのもの

- きちんとデータベースを設計するの必須、プログラムでごまかしはできない
- 定義とロジックの分離を考える
- プログラムは単一機能のモジュールで与える
- 作り込みより「やり直し」のできるシンプルさ

言語より設計

- 設計に時間をかけるべき

クラウドによりSIが成り立たなくなるか？

- 従来型のSIはかなりつらいと思われる
- 世間の状況に合わせた柔軟なSIが求められる

オンラインサービスの発展でシステム開発いらなくなる？

- オンラインで表計算を使っても、本質は変わらない
- 標準的なサービスを使うだけでは、企業の競争力はつかない
- 情報投資による競争力強化はいつの時代も武器になる

情報化されていないエリアがあるとするれば

- 大企業は法律で縛られていて業務はある意味均一
- 一方で中小企業ほど業務の内容のバリエーションは高い
- 中小企業は言い換えれば「大企業の1つの部門」とも言える
- ただし、顧客の予算は低く、従来型の開発ではコストが合わない

Next Three Years

キーバリューストア対応

- クラウドのサービスを使ってシステム構築
- サーバ管理を不要にする仕組みを目指す
- クラウドの特性を生かした解析処理との統合

開発サポートの強化

- クライアントサイド計算フィールドのサポート
- マスター詳細ページのサポート
- Webページのパターンに沿った開発ガイド

サイト構築中

年賀状印刷 - 切・貼なし全画 ×

www.aisatu.jp

年賀状印刷・暑中見舞い（かもめ〜る）は「ふち無しはがき印刷本舗」。郵便局の官製はがきにふち無しを可能にしました。データ入稿・印刷代行でお客サポート。

官製ハガキのフチ無し印刷専門サービス
ふち無しはがき印刷本舗

お問い合わせは今すぐこちらまで
048-813-3003
月～金曜日 AM 9:00～PM 6:00

データ入稿はコチラから

ふち無し年賀状無料サンプル申し込みは [いいね!](#) をポチして、[コチラ](#)から

ふち無し年賀状印刷 データ入稿 印刷代行 専門サイトです。
喪中はがき・暑中見舞いなど挨拶状のふち無しはがき印刷 データ入稿・印刷代行

**2013年賀状 多数ご注文いただきありがとうございました。
年賀状は終了いたしました。暑中見舞いを受付中です!!**

ふち無しはがき印刷本舗の
暑中見舞い
年賀状の返信・喪中の方への挨拶状 [詳しくはこちら](#)

年賀状の返信に、喪中の方へのご挨拶に、寒中見舞いもふち無しはがき印刷を承ります。
松が明けてからの年賀状の返信や、喪中の方へのご挨拶には暑中見舞いを出します。
本来は寒のうちに出せば良いのですが、新しい年のご挨拶ですので1月中に出すのが望ましいでしょう。

「暑中お見舞い」も白フチの出ないふち無しはがき印刷で!

- ホーム
- ふち無しはがき印刷の特徴
- はがき印刷 注文の流れ
- 年賀状 データ入稿 制作方法
- ふち無し年賀状テンプレート
- オリジナルデザイン
- 宛名印刷・投函代行サービス
- 色校正・出力見本サービス
- ビジネス年賀状
- 喪中はがき
- 暑中見舞い
- 年賀状印刷と書き方辞典

まとめ

* INTER-Mediatorの特徴

- データベースと要素がバインドし、編集結果の書き戻しもサポート
- JavaScriptとPHPによるプログラミングで、ユーザインタフェースや複雑なデータ処理に対応
- オープンソース MIT License
- スマートフォン対応、IE6非対応！
- 認証、ページング、バリデーション、ブラウザ判別に対応
- クラウド対応に向けて改良中

* サイト&コミュニティ

- <http://inter-mediator.info/>
- Facebook、Google Groupsにコミュニティ