INTER-Mediatorで作るWebサイト

新居雅行
Masayuki Nii

OSC2012.DB 2012/7/25

Certified
System Administrator 10.6

Certified Trainer
Mac OS X Server Essentials 10.7

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

大学での非常勤講師

- 慶応義塾大学、専修大学

Agenda

INTER-Mediatorへ至る道

INTER-Mediatorで作るデータベース連動Webページ

INTER-Mediatorの仕組みと動作

複雑なアプリケーションを作るための手法

システム開発における利用

Motivation

HTMLにフィールドのデータを表示

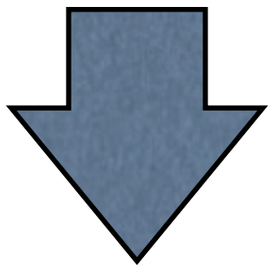
一般的なフレームワークでは…

```
echo “<td><span class=\”x\”>{$row[‘field’]}</span></td>”;
```

or

```
<td><span class=”x”><?php echo $row[‘field’]; ?></span></td>
```

言語の混在



INTER-Mediatorでは…

```
<td><span class=”x IM[table@field]”></span><td>
```

HTMLでのみ記述

繰り返し

一般的なフレームワークでは…

```
<?php foreach ( $result as $row )  { ?>
<tr><td><?php echo $row['field']; ?></td></tr>
:
<?php } ?>
```

言語の混在

ロジックとプレゼンテーションの混在

INTER-Mediatorでは…

```
<table>
<tbody>
<tr><td class="IM[table@field]"></td></tr>
:
</tbody>
</table>
```

繰り返す

ノード複製による繰り返し

HTMLのみで記述

INTER-Mediatorのゴール

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ニグ不要に

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

How Does It Work?
with a Demo

基本的な開発手順

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

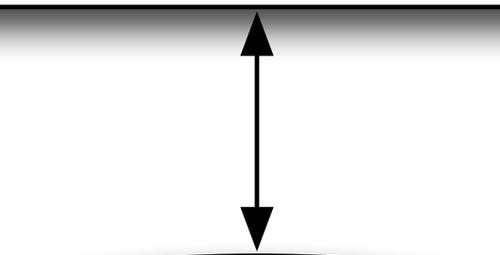
Referencing with
"SCRIPT" tag

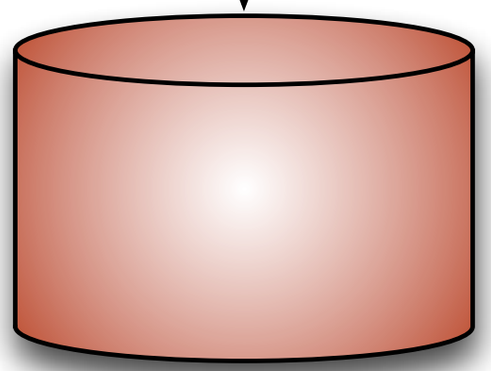
Call the templating
method

Definition File

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

PHP





Database

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

HTML

Modified Page File

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
- 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


管理画面
Miami-D
miami -
Facebook
12 The H
content.h
localhost/im/Demo-Script/demo2.html
アップル
Remember The Milk
Evernote Web
INTER-Mediator - Sa
Facebook
Google+
更新
レコード番号1-10 / 3654
<<
<
>
>>
Postal Code
Prefecture
Address
1000000
東京都
千代田区 以下に掲載がない場合
today is shine.
1020072
東京都
千代田区 飯田橋
next
1020082
東京都
千代田区 一番町
next 2
1010032
東京都
千代田区 岩本町
1010047
東京都
千代田区 内神田
1000011
東京都
千代田区 内幸町
1000004
東京都
千代田区 大手町（次のビルを除く）
1006890
東京都
千代田区 大手町ＪＡビル（地階・階層不明）
1006801
東京都
千代田区 大手町ＪＡビル（１階）
1006802
東京都
千代田区 大手町ＪＡビル（２階）

INTER-Mediatorの動作のポイント

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

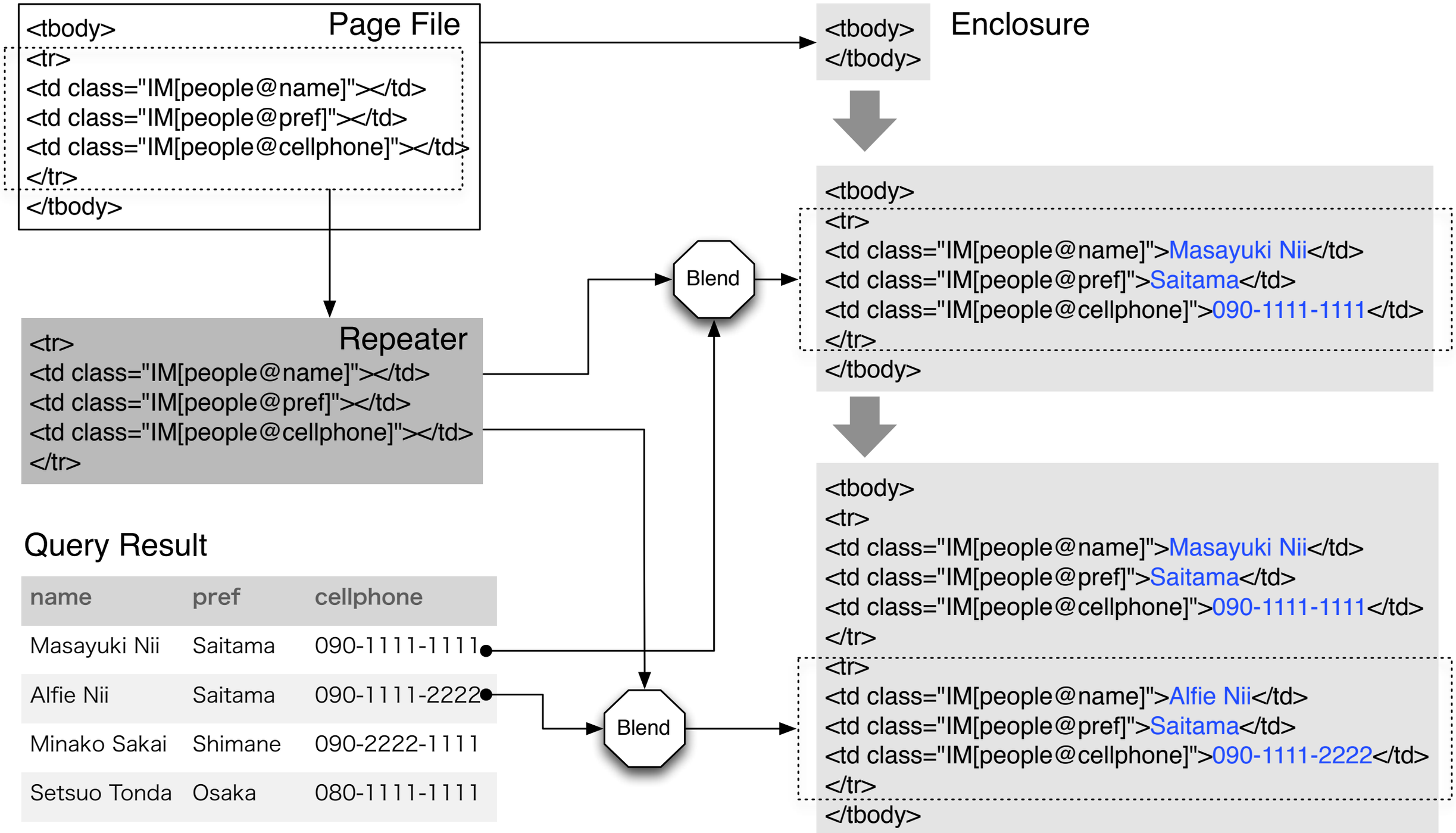
テンプレート処理の基本

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

ページ展開の流れ



Query Result

name	pref	cellphone
Masayuki Nii	Saitama	090-1111-1111
Alfie Nii	Saitama	090-1111-2222
Minako Sakai	Shimane	090-2222-1111
Setsuo Tonda	Osaka	080-1111-1111
:	:	:

リンクノードへの適用

適用可能なもの

- テキストノードがデフォルト、innerHTMLも可能
- 属性やスタイルへの適用が可能
- 追記や置き換えも可能
- 1つのノードで複数の適用も記述可能（｜で区切る）

チェックボックスやポップアップにも対応

- データベースからの選択肢も可能

テキストフィールドの編集処理

- レコードの主キーの指定が必要
- テキストフィールドからフォーカスアウトすると自動的に書き戻し
- 「まとめて書き戻し」もサポート

バリデーションのサポート

- 定義ファイルにJavaScriptのプログラム片を記載

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

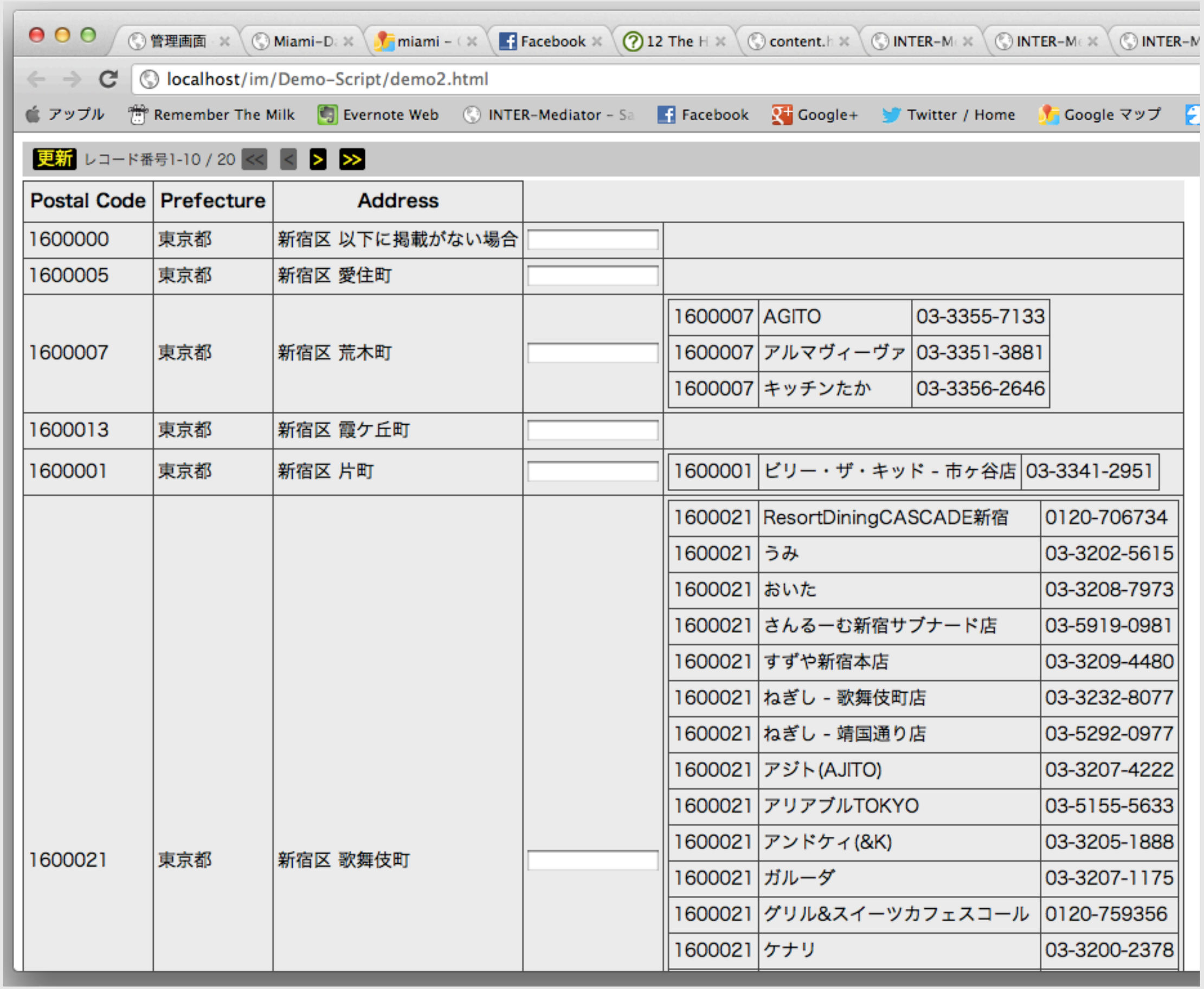
コンテキストに記述可能な内容

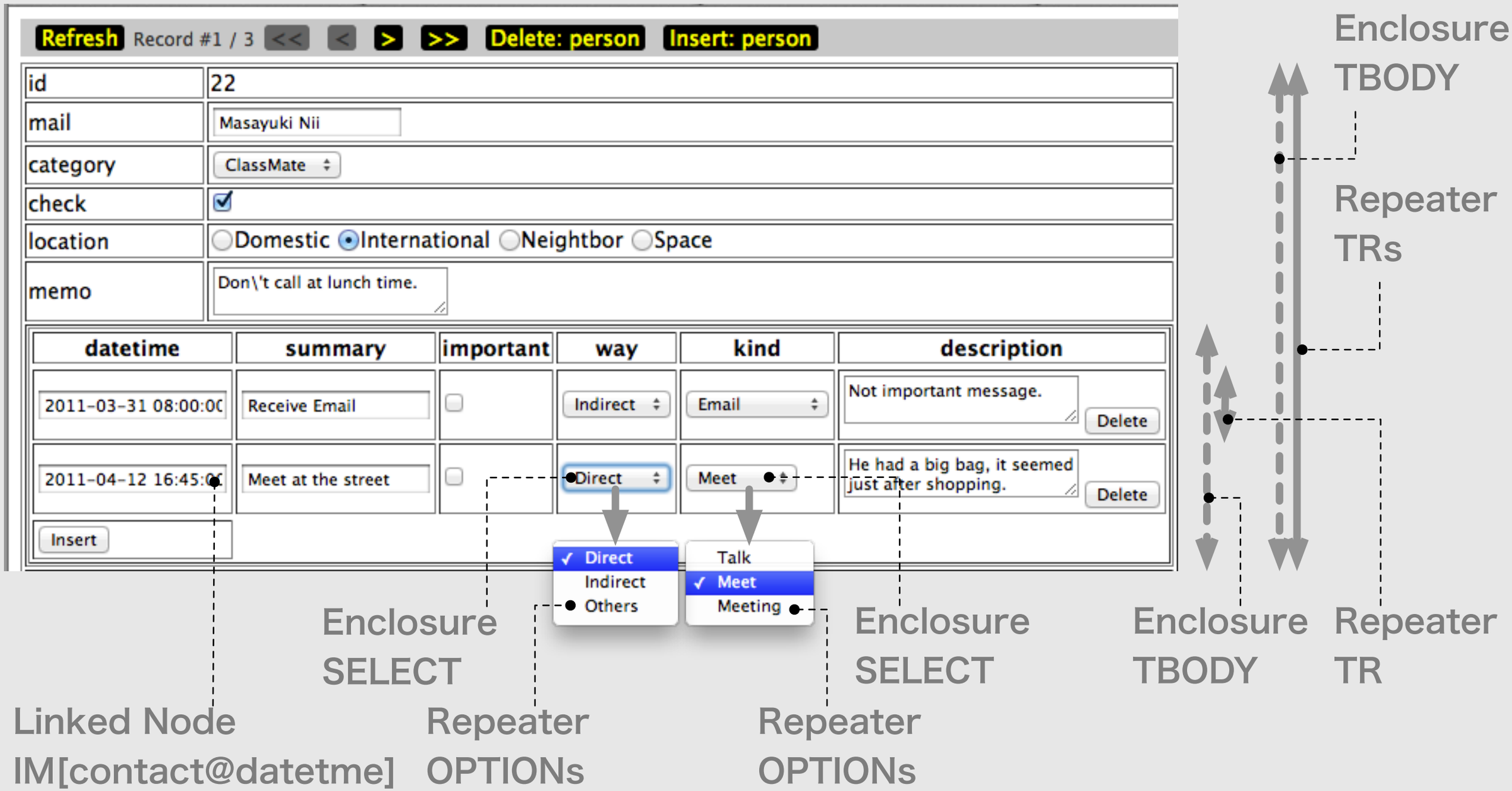
キー名	設定する内容
name	コンテキスト名
view	参照時のデータベースエンティティ
table	更新時のデータベースエンティティ
key	キーフィールド名
paging	ページング処理
query	検索条件
sort	ソート条件
default-values	新規レコード作成時のフィールド値
relation	他のコンテキストとの関連（外部キーなど）
records	先頭から何レコード分を利用するか
repeat-control	レコード追加や削除のユーザインタフェース
authentication	認証とアクセス権に関する設定
extending-class	データベース処理の拡張クラス名
post-repeater, post-enclosure	展開処理で呼び出されるJavaScriptのメソッド
validation	バリデーション
db-class etc.	データベースに関するパラメータ（通常は共通設定）

グレイのボックスは、動的に指定が可能

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


管理画面
Miami-D
miami -
Facebook
12 The H
content.h
INTER-M
INTER-M
INTER-M
localhost/im/Demo-Script/demo2.html
アップル
Remember The Milk
Evernote Web
INTER-Mediator - Sa
Facebook
Google+
Twitter / Home
Google マップ
更新 レコード番号1-10 / 20 << < > >>
Postal Code
Prefecture
Address
1600000 東京都 新宿区 以下に掲載がない場合
1600005 東京都 新宿区 愛住町
1600007 東京都 新宿区 荒木町
1600007 AGITO 03-3355-7133
1600007 アルマヴィーヴァ 03-3351-3881
1600007 キッチンたか 03-3356-2646
1600013 東京都 新宿区 霞ケ丘町
1600001 東京都 新宿区 片町
1600001 ビリー・ザ・キッド - 市ヶ谷店 03-3341-2951
1600021 東京都 新宿区 歌舞伎町
1600021 ResortDiningCASCADE新宿 0120-706734
1600021 うみ 03-3202-5615
1600021 おいた 03-3208-7973
1600021 さんるーむ新宿サブナード店 03-5919-0981
1600021 すずや新宿本店 03-3209-4480
1600021 ねぎし - 歌舞伎町店 03-3232-8077
1600021 ねぎし - 靖国通り店 03-5292-0977
1600021 アジト(AJITO) 03-3207-4222
1600021 アリアブルTOKYO 03-5155-5633
1600021 アンドケィ(&K) 03-3205-1888
1600021 ガルーダ 03-3207-1175
1600021 グリル&スイーツカフェスコール 0120-759356
1600021 ケナリ 03-3200-2378


Refresh
Record #1 / 3
<<
<
>
>>
Delete: person
Insert: person
id
22
mail
Masayuki Nii
category
ClassMate
check
location
Domestic
International
Neightbor
Space
memo
Don\'t call at lunch time.
datetime
summary
important
way
kind
description
2011-03-31 08:00:0C
Receive Email
Indirect
Email
Not important message.
Delete
2011-04-12 16:45:0C
Meet at the street
Direct
Meet
He had a big bag, it seemed just after shopping.
Delete
Insert
Direct
Indirect
Others
Talk
Meet
Meeting
Enclosure
TBODY
Repeater
TRs
Enclosure
SELECT
Enclosure
SELECT
Enclosure
TBODY
Repeater
TR
Linked Node
IM[contact@datetme]
Repeater
OPTIONs
Repeater
OPTIONs

さまざまな機能

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

ブラウザの判定処理

認証とアクセス権

- 独自テーブルおよびデータベースエンジンのユーザでの認証
- メディアに対する認証プロキシもサポート

Demo 3

- 他のフィールドに依存したポップアップメニューの選択肢を構築する
- ページに認証をかける

(1) INTER-Mediator(Ja | INTER-Mediator | (1) Facebook | オープンソースカンフ | Evernote Web | IN

localhost/im/Sample_form/form_FMS.html

アップル | Remember The Milk | Evernote Web | INTER-Mediator - Sa | Facebook | Google+ | Twitter / Home | Google マップ

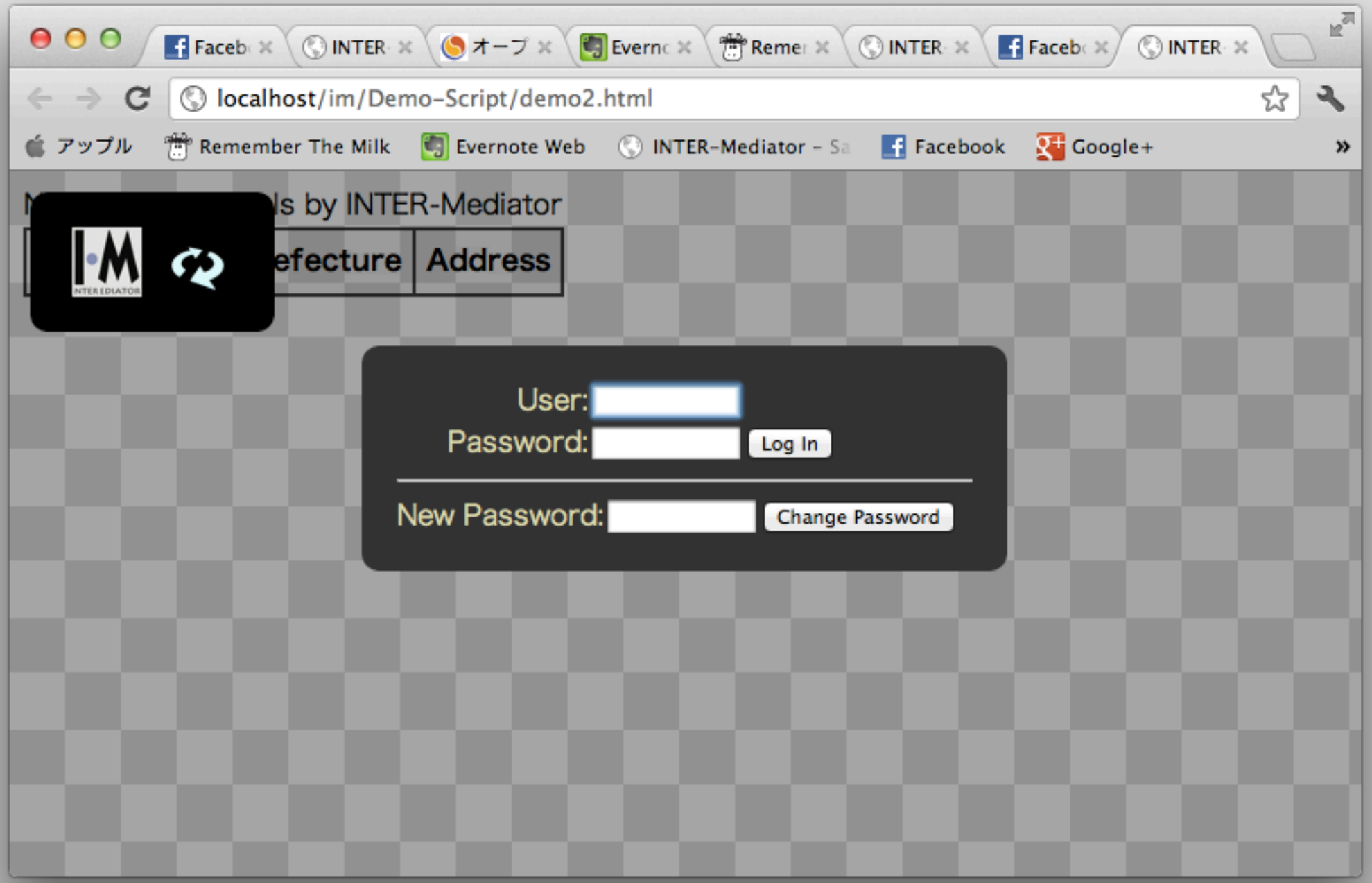
Refresh Record #1 / 3 << < > >> Delete Record: person_layout Add Record: person_layout

id	1
address	Saitama
mail	msyk@msyk.net
category	Collegue
check	☑
location	○Domestic ◉International ○Neightbor ○Space
memo	He is Japanese. He is Japanese. He is Japanese. He

person_id	datetime	summary	important	way	kind	description	
1	2009/01/01 03:03:04	Send Mail	☐	Direct	Talk	Send Mail6	Delete
1	2000/01/01 00:00:00	Meet at town	☑	Indirect	Mail	テスト テスト	Delete
1	2010/02/11 13:00:00	Cocoa勉強会	☐	Others	Twitter		Delete

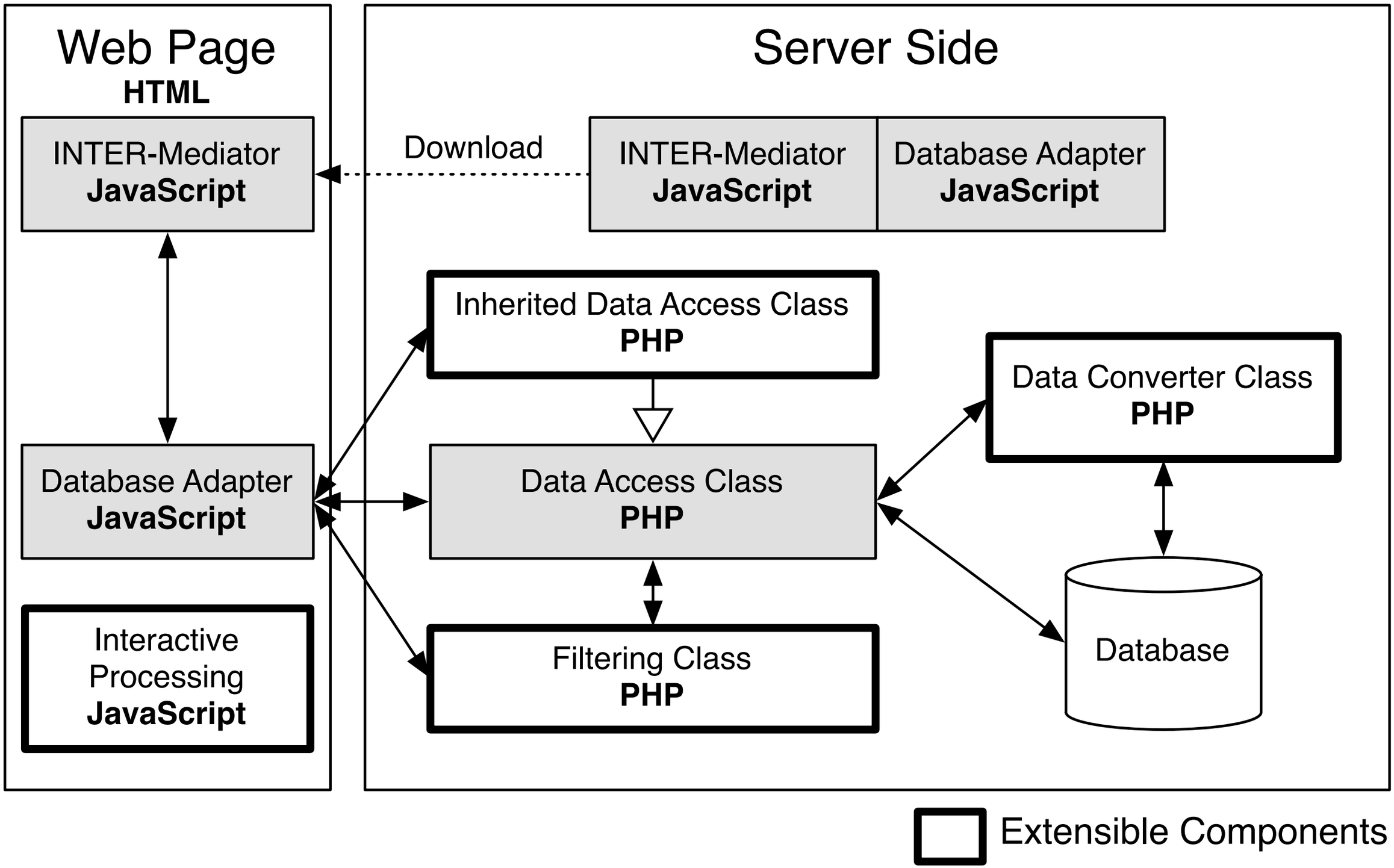
Talk / Meet / Meeting

Calling / ✓ Mail / Email / See on Chat / Twitter


localhost/im/Demo-Script/demo2.html
アップル
Remember The Milk
Evernote Web
INTER-Mediator - Sa
Facebook
Google+
ls by INTER-Mediator
efecture
Address
User:
Password:
Log In
New Password:
Change Password

Extensible

複雑な処理への対応


Web Page
HTML
INTER-Mediator
JavaScript
Database Adapter
JavaScript
Interactive
Processing
JavaScript
Server Side
Download
INTER-Mediator
JavaScript
Database Adapter
JavaScript
Inherited Data Access Class
PHP
Data Access Class
PHP
Filtering Class
PHP
Data Converter Class
PHP
Database
Extensible Components

データコンバータクラスの例

定義ファイル

```
array(
    'formatter' => array(
        array(
            'field' => 'PageInfo@Modified',
            'converter-class' => 'MySQLDateTime',
            'parameter' => 'Y/m/d H:i'),
        array(
            'field' => 'NewsPage@Modified',
            'converter-class' => 'MySQLDateTime',
            'parameter' => 'Y/m/d H:i'),
    )
),
```

```
class DataConverter_HTMLString
{

    function converterFromUserToDB($str)    {
        return $str;
    }

    function converterFromDBtoUser($str)    {
        return str_replace("\n", "<br/>",
            str_replace("\r", "<br/>",
                str_replace("\r\n", "<br/>",
                    str_replace(">", "&gt;",
                        str_replace("<", "&lt;",
                            str_replace("&", "&amp;", $s
    }
}
```

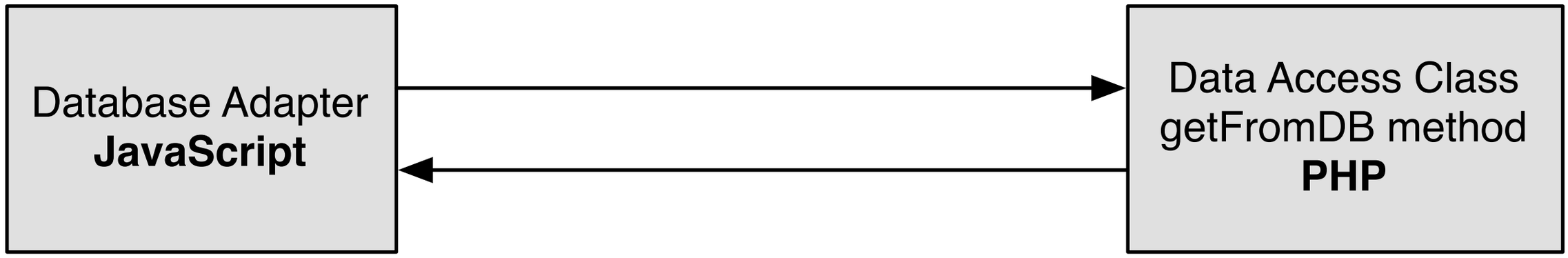
データアクセスクラスの例

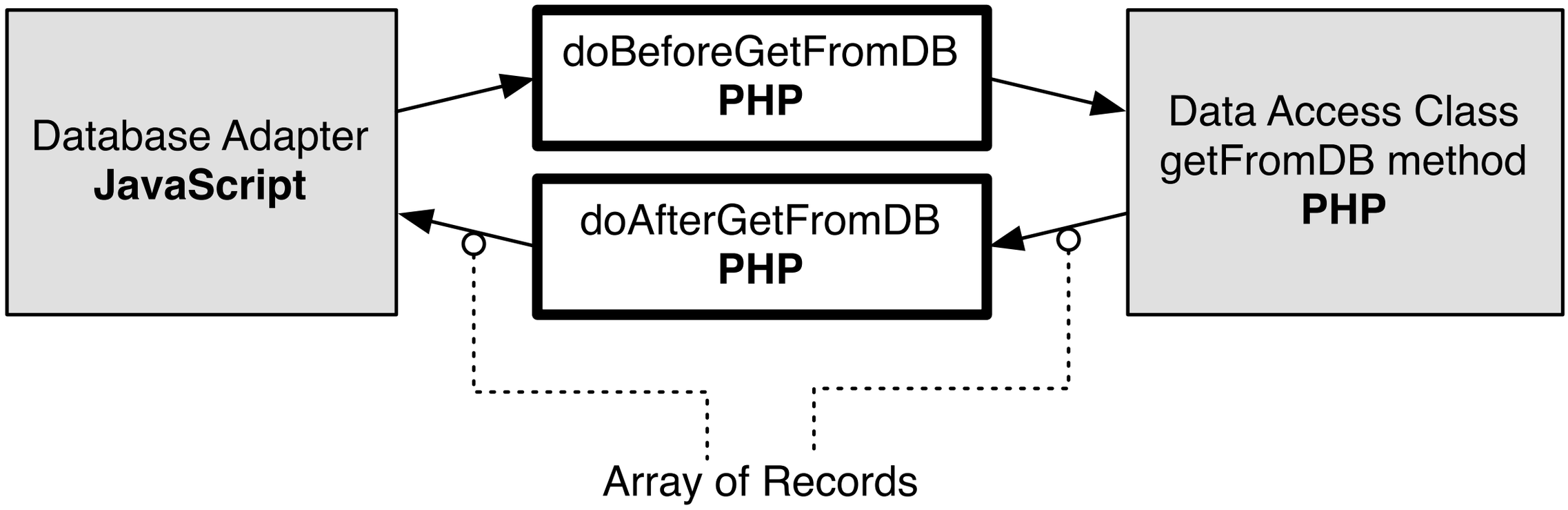
```
include_once('../INTER-Mediator/DB_FileMaker_FX.php');

class DB_WebSite_FMSFX extends DB_FileMaker_FX   {

    function getFromDB( $sourceName )     {
        $returnValue = parent::getFromDB( $sourceName );
        if ( count( $returnValue ) > 1 )    {
            // Check for the language of documents
            $lang = array();
            foreach( $returnValue as $record )  {
                $lang[$record['Language']] += 1;
            }
                    :
        return $returnValue;
    }
    function setToDB($dataSourceName)  { … }
    function newToDB($dataSourceName) { … }
    function deleteFromDB($dataSourceName) { … }
}
```

フィルタリングによる拡張





フィルタリングクラスの例

```
class SumForCustomers implements Extending_Interface_AfterGet {

    function doAfterGetFromDB($dataSourceName, $result) {
        $sum = array();
        foreach ($result as $record) {
            $sum[$record["customer"]] += $record["total"];
        }
        arsort($sum);
        $result = array();
        $counter = 10;
        foreach ( $sum as $customer => $totalprice )  {
            $result[] = array(
                "customername"=>$customer,
                "totalprice"=>number_format($totalprice)
            );
            $counter--;
            if ( $counter <= 0 )    {
                break;
            }
        }
        return $result;
    }
```

定義ファイル

```
array(
    'name' => 'summary2',
    'view' => 'saleslog',
    'extending-class' => "SumForCustomers",
),
```

クライアント側のプログラム例

定義ファイル

```
array(
    'name' => 'invoice',
        :
    'post-enclosure' => 'invoiceExpanded',
),
array(
    'name' => 'item',
        :
    'post-repeater' => 'itemsExpanded',
),
```

```
INTERMediatorOnPage.invoiceExpanded
    = function (target) { calcTotal(target); };

INTERMediatorOnPage.itemsExpanded
    = function (repeaters) { modLine(repeate

function modLine(target) {
        :
}

function calcTotal(target) {
        :
}
```

https://github.com/voodootikigod/logo.js

クライアント側のプログラム例

```
function modLine(target)  {
    var qtyId = INTERMediatorOnPage.getNodeIdFromIMDefinition("items@qty",target);
    var unitPriceId
        = INTERMediatorOnPage.getNodeIdFromIMDefinition("items@unitprice",target);
    var productPriceId
        = INTERMediatorOnPage.getNodeIdFromIMDefinition("product@unitprice",target);
    var amountId
        = INTERMediatorOnPage.getNodeIdFromIMDefinition("items@amount",target);
    if ( qtyId != null && unitPriceId != null && productPriceId != null && amountId != null) {
        var unitPrice = new Number( document.getElementById(unitPriceId).value > 0 ?
                                document.getElementById(unitPriceId).value :
                                document.getElementById(productPriceId).innerHTML );
        var amountPrice = unitPrice * document.getElementById(qtyId).value;
        document.getElementById(amountId).innerHTML
                    = INTERMediatorLib.numberFormat(amountPrice);
    }
}
```

https://github.com/voodootikigod/logo.js

INTER-Mediatorで
システム開発

動作環境

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

パターンなWebページ構成

入力フォーム系

- フォームを適当に作る
- JavaScriptのAPIを呼び出して新規レコード作成

検索ページ

- フォームとボタンを用意し、ボタンを押したらページ生成する

一覧と詳細のページを行き来する

- URLのパラメータにキーフィールド値を乗せて詳細ページへ
- 1つのページに両方のページを作成し、一方のCSSのdisplay属性をnoneに

計算結果を表示する

- 前出のJavaScriptのAPIを利用

メールを送付する

- 前出のフィルタクラスで記述可能

INTER-MediatorとWeb開発

パブリックなページには使いにくい

- JavaScriptによるテンプレート処理であり、検索エンジンが中身を読めない
- 従って、中小企業や部門レベルの業務システムに向く

開発をする上での特徴

- HTMLのモックアップから始めることができる
- データベースのバインドまでは簡単＆プログラミング不要
- 開発コストの低減を目指せるかもしれない

開発の上でのエンジニアリング

- データベースの論理設計は正しくされていないとかなりつらい
- バインド以上の処理を、どこでどのように組むかが悩みどころ
- 仕組みの上ではFileMakerに似ている

フレームワークとしての機能

MVCではない

- MVCを否定するものではありませんが…
- 簡単なページを簡単に作る仕組みを考えた結果

O/Rマッピングではない

- 「組み込まない」のではなく、「組み込んでいない」
- 基本的に「メタデータの流れ」でフレームワークは稼働する

短期間の開発が求められている

- Visual Studio LightSwitch、UnitBase、kintoneなども登場
- 特集3　システムを2週間で作る～日経SYSTEMS 2012/08

低コストの開発が求められている

- いままでのやり方を見直す必要がある

今後の開発プラン

メディア対応

- ファイルのアップ/ダウン、画像、ビデオ等
- 認証トークンが必要なメディアプロキシは実装済み

Local DB/Cloud DB対応

サーバサイドのプログラムの高度化

ドキュメント整備

テストの整備

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
- サイト&コミュニティ
  - http://inter-mediator.info/
  - Facebook、Google Groupsにコミュニティ
  - 2012/7/27(Fri)：INTER-Mediator Jelly #2
  - 2012/7/31(Tue)：INTER-Mediator Meet-up #2