INTER-Mediator Meet-up #3

Masayuki Nii

July 3, 2013

KDDI Web Communications Seminar Room

Agenda

事例報告

開発アップデート

IMCake（松尾さん）

INTER-Mediatorを使った新サービス（鈴木さん）

今後の開発計画とディスカッション

INTER-Mediator

inter-mediator.info

[←CoolNotify]| ←FileMaker情報 | ←Browsing Mac OS X

INTER-Mediator

Updated at 2013-06-16 08:35:20

新居雅行 Masayuki Nii／msyk@msyk.net

INFO 開発状況のアップデート報告などを含む「INTER-Mediator Meet-up #3」を、2013年7月3日(水)に、KDDIウェブコミュニケーション様のセミナールームを借りて行います。参加無料です。

INTER-Mediatorについて、じっくり学習したい方向けに、**e-Learningコース**をご用意しました。
有償（20,000円税込）ですが、基礎から応用までが座学と演習でマスターできます。こちらよりお申し込みください。

“ふち無しはがき印刷本舗”のオーダー受付管理

年賀状や暑中見舞いなどのはがき印刷を請け負う「ふち無しはがき印刷本舗」では、受注および進捗管理のためのサイトをINTER-Mediatorで構築し、2013年の暑中見舞いシーズンより公開ました。ふち無しはがき印刷本舗では独自の技術で官製はがきの周囲をカットせずにふち無し印刷を実現しています。はがきの印刷内容は発注ごとに内容が変わり、その都度デザインを作って対応します。印刷申し込みから入稿、デザイン決定、そして発送までのやりとりを、INTER-Mediatorで作った会員向けのサイトで実現しました。データベースはFileMaker Server 12、サーバ機はMac miniを利用し、内部での業務処理はFileMaker Proから行い、発注者に対するユーザインタフェースをWebアプリケーションで構築しています。なお、サービスを紹介する一般向けサイトはJimdoを利用しています。

ふち無しはがき印刷本舗を展開する西ヶ谷一志代表は、「昨年までのフォームによる申し込みとファイル転送サービスの利用では、入稿ファイルやお客様からのメールの突き合わせに時間が取られてしまっていました。そのために新しいシステムをじっくりと考えて作りました。オンデマンド印刷サービスの会員向けサービスとしては、すべての機能がまとまっているため、使いやすく感じてもらえるはずです」と、新システムによる顧客満足度の向上に手応えを感じているそうです。また、Webインタフェースについては「INTER-Mediatorを使うことで開発時間は短縮できることを聞いていたので、ぎりぎりまで内容の検討に時間をかけることができました。思った以上に短期間でWebアプリケーションが立ち上がり、またその後の修正も容易なので驚いています」と話します。（掲載日：2013-6-16）

CLOUD CORE

VPSとクラウドでオープンソース開発者を支援します

Contents

Community

- Facebook INTER-Mediator グループ日本語|英語 ： Facebookの「グループ」機能を使ったグループです。
- GitHub ：ソースコードのレポジトリです。／変更履歴,将来計画（英語）／開発者によるBlog - 以前のBlog
- ショップ（ロゴグッズ）
- Google Group ：メーリングリストなどがありますが、事実上休止状態です。

News and Updates

- 2013-06-25 Ver.3.5をリリースしました。ユーザ名に対応付けたメールアド

Ver.3.0-3.6

JavaScript Componentのサポート

- HTMLエディタTinyMCEをサポート
- ファイルアップロードコンポーネントを独自作成

入力専用フォーム

APIの追加

- 特に、INTERMediatorOnPage.doAfterConstruct

認証ダイアログボックスのカスタマイズ

たくさんのバグフィックス

JavaScriptコンポーネント

ページファイルの記述

- <td class="IM[testtable@text1] IM_WIDGET[tinymce]"></td>
- <td class="IM[testtable@vc1] IM_WIDGET[im_fileupload]"></td>

定義ファイルの設定

- ファイルのアップロードでの指定
- アップロード結果，コンテキストに新しいレコードを追加
- 'file-upload' => array(
    array('field'=>'vc1', 'context'=>'fileupload',)),

汎用化

- INTER-Mediator-Parts.jsファイルを参照

入力専用フォーム

ページファイルでの記述

- class="_im_post"
- コンテキスト内のコンテキストは、通常通りテンプレート処理を行うので、マスターから値を取り出すこともできる
- ボタンをクリックした後の処理を定義できる

```
<table>
    <tbody class="_im_post">
    <tr>
        <td>
            <input type="text" id="message" class="IM[chat@message]"/>
        </td>
        <td>
            <button class="_im_post">Post</button>
        </td>
    </tr>
    </tbody>
</table>
```

APIの追加と変更

テンプレート処理

- INTERMediatorOnPage.doAfterConstruct = function(){};
- テンプレート処理の終了直後にコール

ノード検索

- INTERMediatorOnPage.getNodeIdsFromIMDefinition
  (imDefinition, fromNode, justFromNode)
- 3つ目がtrue=fromNodeから、false=上位のエンクロージャから

コンソールログ

- INTERMediator.supressDebugMessageOnPage = true;
- INTERMediator.supressErrorMessageOnPage = true;

ファイルアップロードコンポーネント

認証ダイアログボックスのカスタマイズ

定義ファイルで電子メールをアカウント利用

- $options['authentication']['email-as-username'] = true;

INTERMediator.construct();の前に変数設定

```
INTERMediatorOnPage.isShowChangePassword = false;
INTERMediatorOnPage.defaultBackgroundImage = null;
INTERMediatorOnPage.defaultBackgroundColor = "white";
INTERMediatorOnPage.loginPanelHTML = '<table class="loginbox"><tbody>
<tr><th>アカウント</th>

<td><input type="text" id="_im_username" name="username"/></td></tr>
<tr><th>パスワード</th>

<td><input type="password" id="_im_password" name="password"/></td></tr>
<tr><th></th><td><button id="_im_authbutton">ログイン</button></td></tr>

</tbody></table>';
```

INTER-Mediatorの将来

KVS対応

- 次のスライド

パターンに基づくWebサイト構築

- 現在、パターンを抽出中

Key-Value Storeへのサポート

KVSを使った場合のメリット

- スキーマ設計が不要（あるいはSQLを書かなくてもいい）
- つまり、ページファイルや定義ファイルの記述がスキーマの一部
- AWSなどのクラウドサービスが利用できる

デメリット、あるいはメリットにならない？

- スキーマの自動生成はSQLでも可能
- ページファイルや定義ファイルでカバーしない範囲のモデルは？
- 自動的に形成されたスキーマが正しいかどうかの判定はどうする？

どのメリットが大きいでしょうか？

何がメリットでしょうか？

ディスカッション

INTER-Mediatorを使う上で困っていることは？