

INTER-Mediator

ハンズオンセッション手順書



INTER-Mediator 《大》勉強会
2016年8月6日@国立情報学研究所

INTER-Mediator Directive Committee
inter-mediator.org

はじめに

この手順書は、勉強会で利用するために作成し、その後に何度か改定したものです。INTER-Mediatorを使ったWeb開発がどのようなものかを知りたい方が個人で進めることもできますが、この手順書を利用して、2～3時間かけたオフラインのハンズオンミーティングを開催することもできます。INTER-Mediatorをまったく触ったことがないが、どんな感じなのかを知りたい方、あるいはINTER-Mediator普及のために、活用していただけます。

この文書は、印刷して利用することを前提にしていますが、ほぼ同一の文書が以下のURLよりオンラインでも参照できます。オンライン版では、URLをリンクとして用意していますし、プログラムのコピー&ペーストも可能です。状況に応じて使い分けてください。

<http://inter-mediator.info/ja/for-novices/handson.html>

使用までの準備

以下のURLにある『INTER-Mediatorがインストールされた動くサーバのVirtual Machine』を参照して、INTER-Mediatorが稼働するVMを用意してください。ダウンロードしてすぐにINTER-Mediatorを稼働することができます。

<http://inter-mediator.info/ja/for-novices/vm.html>

MySQLをご利用される方は、上記のVMで準備完了です。FileMaker Serverを利用される方は、ホストOS側に、FileMaker Serverをインストールしてください。加えて、以下のリンク先のデータベースファイルTestDBをダウンロードして、FileMaker Serverで公開をしておいてください。

<https://github.com/INTER-Mediator/INTER-Mediator/blob/master/dist-docs/TestDB.fmp12?raw=true>

なお、以下の手順では、VirtualBoxのホストオンリーアダプターが、既定の設定（ホストのIPアドレスが192.168.56.1、ゲストが192.168.56.101）になっていることを前提として記述しています。

VMの準備が整ったら、VMを起動します。そして、ホストOS側では、ブラウザを起動して、<http://192.168.56.101/> に接続します。

<http://192.168.56.101/> で表示されるページ = 「VMのホーム」と呼びます。

セッション1：入力専用ページ

1-1：質問の入力ページを作る

定義ファイルの作成（ファイル名：def01.php）

1. VMのホームで、「def01.phpを編集」をクリックします。
2. Queryの下にある行の右側の「削除」をクリックして、Queryの下に行がないようにします。
3. Sortの下にある行の右側の「削除」をクリックして、Sortの下に行がないようにします。
4. nameに「survey」、keyに「id」と入力します。Contextsセクションには、この2つのテキストフィールドだけがあるようにします。
5. Optionsセクションには何も入力されていない状態のままでかまいません。
6. Database Settingsセクションには以下のように入力します。
 - a. [FileMaker Server]
Database Settingsセクションのdb-classには「FileMaker_FX」、databaseには「TestDB」、userには「web」、passwordには「password」、serverには「192.168.56.1」、portには「80」、protocolには「http」、datatypeには「FMPro12」と入力します。
 - b. [MySQL]
Database Settingsセクションのdb-classには「PDO」、dsnに「mysql:host=localhost;dbname=test_db;charset=utf8mb4」userには「web」、passwordには「password」と入力します。
7. Debugのところは「false」と入力します。

ページファイルの作成（ファイル名：page01.html）

1. VMのホームで、「page01.htmlを編集」をクリックします。
2. ページファイルエディタが起動します。以下のソースコードを入力します。

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html;charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>INTER-Mediator Demo</title>
  <script type="text/javascript" src="def01.php"></script>
</head>
<body>
<table>
  <tbody data-im-control="post">
    <tr>
      <th>名前</th>
      <td><input type="text" data-im="survey@Q5"/></td>
    </tr>
    <tr>
      <th>質問1</th>
```

```

        <td><input type="text" data-im="survey@Q1"/></td>
    </tr>
    <tr>
        <th></th>
        <td><button data-im-control="post">Answer</button></td>
    </tr>
</tbody>
</table>
</body>
</html>

```

ページを表示して、適当に内容を答える

1. VMのホームで、「page01.htmlを表示する」をクリックします。
2. フォームが表示されます。適当に入力して、Answerボタンをクリックします。
3. [FileMaker Server]
TestDBのsurveyレイアウトを確認して、データが入力されていることを確認します。
[MySQL]
test_dbのsurveyテーブルを確認して、データが入力されていることを確認します。以下、いちばん手軽な手順を示します。
 1. VirtualBoxのVMのウインドウ（真っ黒のウインドウ）をクリックして、ユーザ名「developer」、パスワード「im4135dev」でログインをします。
 2. プロンプトで、「mysql -u web -p test_db -e 'select * from survey;」と入力します。
 3. パスワードをたずねられるので「password」と入力します。
4. 適当にいくつかフォーム入力して、レコードが増えることを確認します。

1-2：入力フォームらしい感じにする

1. VMのホームで、「def01.phpを編集」をクリックするか、すでに開いている定義ファイルエディタのページを参照します。
2. 右上のShow Allボタンをクリックします。
3. post-reconstructに「true」と入力します。
4. post-dismiss-messageに「送信しました」と入力します。
5. post-move-urに「http://inter-mediator.org/」と入力します。URLは別のものでもかまいません。その後、入力を確定するために、Tabキーを押しておきます。
6. VMのホームで、「page01.htmlを表示する」をクリックするか、すでに開いているpage01.htmlを更新します。
7. フォームが表示されます。適当に入力して、Answerボタンをクリックします。今度は、Answerボタンが消え、数秒後に別のページにジャンプしました。
8. [FileMaker Server]
TestDBのsurveyレイアウトを確認して、データが入力されていることを確認します。
[MySQL]
test_dbのsurveyテーブルを確認して、データが入力されていることを確認します。（手順はすでに説明しました。）

1-3：テキストフィールド以外のフォーム要素

チェックボックス

page01.htmlの中のTABLEタグの内容を以下の様に変更して、フォームを表示すると、チェックボックスが追加されます。フォームに入力して、Answerボタンを押してみて、データベースにどのように入力されるのかを確認しましょう。

```
<table>
  <tbody data-im-control="post">
    <tr>
      <th>名前</th>
      <td><input type="text" data-im="survey@Q5"/></td>
    </tr>
    <tr>
      <th>質問1</th>
      <td><input type="text" data-im="survey@Q1"/></td>
    </tr>
    <tr>
      <th>質問2</th>
      <td><input type="checkbox" value="1" data-im="survey@Q2"/>はい</td>
    </tr>
    <tr>
      <th></th>
      <td><button data-im-control="post">Answer</button></td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

ラジオボタン

page01.htmlの中のTABLEタグの内容を以下の様に変更して、フォームを表示すると、ラジオボタンが追加されます。フォームに入力して、Answerボタンを押してみて、データベースにどのように入力されるのかを確認しましょう。

```
<table>
  <tbody data-im-control="post">
    <tr>
      <th>名前</th>
      <td><input type="text" data-im="survey@Q5"/></td>
    </tr>
    <tr>
      <th>質問1</th>
      <td><input type="text" data-im="survey@Q1"/></td>
    </tr>
    <tr>
      <th>質問2</th>
      <td><input type="checkbox" value="1" data-im="survey@Q2"/>はい</td>
    </tr>
    <tr>
      <th>質問3</th>
      <td>
        <div>
          <input type="radio" value="11" name="q3" data-im="survey@Q3"/>東
        </div>
      </td>
    </tr>
  </tbody>
</table>
```

```

        <input type="radio" value="12" name="q3" data-im="survey@Q3"/>西
        <input type="radio" value="13" name="q3" data-im="survey@Q3"/>南
        <input type="radio" value="14" name="q3" data-im="survey@Q3"/>北
    </div>
</td>
</tr>
<tr>
    <th></th>
    <td><button data-im-control="post">Answer</button></td>
</tr>
</tbody>
</table>
</body>
</html>

```

チェックボックスセット

page01.htmlの中のTABLEタグの内容を以下の様に変更して、フォームを表示すると、チェックボックスセットが追加されます。フォームに入力して、Answerボタンを押してみ、データベースにどのように入力されるのかを確認しましょう。

```

<table>
  <tbody data-im-control="post">
    <tr>
      <th>名前</th>
      <td><input type="text" data-im="survey@Q5"/></td>
    </tr>
    <tr>
      <th>質問1</th>
      <td><input type="text" data-im="survey@Q1"/></td>
    </tr>
    <tr>
      <th>質問2</th>
      <td><input type="checkbox" value="1" data-im="survey@Q2"/>はい</td>
    </tr>
    <tr>
      <th>質問3</th>
      <td>
        <div>
          <input type="radio" value="11" name="q3" data-im="survey@Q3"/>東
          <input type="radio" value="12" name="q3" data-im="survey@Q3"/>西
          <input type="radio" value="13" name="q3" data-im="survey@Q3"/>南
          <input type="radio" value="14" name="q3" data-im="survey@Q3"/>北
        </div>
      </td>
    </tr>
    <tr>
      <th>質問4</th>
      <td>
        <div data-im-group="survey@Q4">
          <input type="checkbox" value="21"/>Docomo
          <input type="checkbox" value="22"/>au
          <input type="checkbox" value="23"/>Softbank
          <input type="checkbox" value="24"/>Willcom
        </div>
      </td>
    </tr>
    <tr>
      <th></th>
      <td><button data-im-control="post">Answer</button></td>
    </tr>
  </tbody>
</table>

```

```
</tr>
</tbody>
</table>
```

1-4：回答を一覧表示する

定義ファイルの作成（ファイル名：def02.php）

1. VMのホームで、「def02.phpを編集」をクリックします。
2. Queryの下にある行の右側の「削除」をクリックして、Queryの下に行がないようにします。
3. Sortの下にある行の右側の「削除」をクリックして、Sortの下に行がないようにします。
4. nameに「survey」、keyに「id」と入力します。recordsには「10」、maxrecordsにも「10」と入力します。
5. Optionsセクションには何も入力されていない状態のままでかまいません。
6. Database Settingsセクションには、次のように入力します。
 - a. [FileMaker Server]
db-classには「FileMaker_FX」、databaseには「TestDB」、userには「web」、passwordには「password」、serverには「192.168.56.1」、portには「80」、protocolには「http」、datatypeには「FMPro12」と入力します。
 - b. [MySQL]
Database Settingsセクションのdb-classには「PDO」、dsnには「mysql:host=localhost;dbname=test_db;charset=utf8mb4」、userには「web」、passwordには「password」と入力します。
7. Debugのところは「false」と入力します。

ページファイルの作成（ファイル名：page02.html）

1. VMのホームで、「page02.htmlを編集」をクリックします。
2. ページファイルエディタが起動します。以下のソースコードを入力します。

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html;charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>INTER-Mediator Demo</title>
  <script type="text/javascript" src="def02.php"></script>
</head>
<body>
<table>
  <thead>
    <tr><th>id</th><th>名前</th><th>質問1</th><th>質問2</th>
    <th>質問3</th><th>質問4</th></tr>
  </thead>
  <tbody>
    <tr>
      <td data-im="survey@id"></td>
      <td data-im="survey@Q5"></td>
      <td data-im="survey@Q1"></td>
      <td data-im="survey@Q2"></td>
      <td data-im="survey@Q3"></td>
```

```

        <td data-im="survey@Q4"></td>
    </tr>
</tbody>
</table>
</body>
</html>

```

3. VMのホームで、「page02.htmlを表示する」をクリックします。回答一覧が表示されました。

ページネーション、追加、削除

1. VMのホームで、「def02.phpを編集」をクリックするか、すでに開いている定義ファイルエディタのページを参照します。
2. pagingに「true」、repeat-controlに「confirm-insert confirm-delete」と入力します。
3. VMのホームで、「page02.htmlを編集」をクリックするか、あるいはすでに開いているページファイルエディタを表示して。以下のソースコードを入力します。「名前」の列はテキストフィールドにします。

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>INTER-Mediator Demo</title>
  <link rel="stylesheet" type="text/css" href="INTER-Mediator/Samples/sample.css">
  <script type="text/javascript" src="def02.php"></script>
</head>
<body>
  <div id="IM_NAVIGATOR"></div>
  <table>
    <thead>
      <tr><td>名前</td><td>質問1</td><td>質問2</td>
      <td>質問3</td><td>質問4</td></tr>
    </thead>
    <tbody>
      <tr>
        <td><input type="text" data-im="survey@Q5"/></td>
        <td data-im="survey@Q1"></td>
        <td data-im="survey@Q2"></td>
        <td data-im="survey@Q3"></td>
        <td data-im="survey@Q4"></td>
        <td></td>
      </tr>
    </tbody>
  </table>
</body>
</html>

```

4. VMのホームで、「page02.htmlを表示する」をクリックするか、すでに表示されているのならそのページを更新します。回答一覧が表示されました。
5. レコードの追加削除ができることを確認します。
6. 名前の文字列を修正して、Tabキーを押し、データベースの内容を確認して、データの修正ができることを確認します。
7. 10レコード以上に増やしてみて、10レコードずつ表示されることを確認します。

セッション2：住所録

定義ファイルの作成（ファイル名：def03.php）

1. VMのホームで、「def03.phpを編集」をクリックします。
2. Queryの下にある行の右側の「削除」をクリックして、Queryの下に行がないようにします。
3. Sortの下にある行の右側の「削除」をクリックして、Sortの下に行がないようにします。
4. [FileMaker Server] nameに「person_list」、tableを「person_layout」、viewに「person_layout」、keyに「-recid」、pagingに「true」、repeat-controlに「insert」、recordsに「10」、maxrecordsに「100」と入力します。他は空白にします。
5. [MySQL] nameに「person_list」、keyに「id」、viewに「person」、pagingに「true」、repeat-controlに「insert」、recordsに「10」、maxrecordsに「100」と入力します。他は空白にします。
6. Optionsセクションには何も入力されていない状態のままでかまいません。
7. Database Settingsの設定は、前のセッションと同一にします。
 - a. [FileMaker Server]
Database Settingsセクションのdb-classには「FileMaker_FX」、databaseには「TestDB」、userには「web」、passwordには「password」、serverには「192.168.56.1」、portには「80」、protocolには「http」、datatypeには「FMPro12」と入力します。
 - b. [MySQL]
Database Settingsセクションのdb-classには「PDO」、dsnには「mysql:host=localhost;dbname=test_db;charset=utf8mb4」、userには「web」、passwordには「password」と入力します。
8. Debugのところは「false」と入力します。

ページファイルの作成（ファイル名：page03.html）

1. VMのホームで、「page03.htmlを編集」をクリックします。
2. ページファイルエディタが起動します。以下のソースコードを入力します。

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html;charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>INTER-Mediator Demo</title>
  <link rel="stylesheet" type="text/css" href="INTER-Mediator/Samples/sample.css">
  <script type="text/javascript" src="def03.php"></script>
</head>
<body>
<div id="IM_NAVIGATOR"></div>
<table>
  <tbody>
    <tr><td></td><td data-im="person_list@name"></td></tr>
  </tbody>
</table>
</body>
```

</html>

3. VMのホームで、「page03.htmlを表示する」をクリックします。適当な一覧が見えています。

コンテキストの追加（ファイル名：def03.php）

1. VMのホームで、「def03.phpを編集」をクリックするか、すでに開いている定義ファイルエディタのページを参照します。
2. Contextsの下の「追加」ボタンをクリックします。nameが「= new context =」の項目が増えました。
3. Database Settingsの設定は、前のセッションと同一にします。
 - a. [FileMaker Server] 増えた項目で、nameを「person_detail」、tableを「person_layout」、viewを「person_layout」、keyを「-recid」、recordsを「1」、maxrecordsを「100」として、あとは空白のままにします。
 - b. [MySQL] 増えた項目で、nameを「person_detail」、viewを「person」、keyを「id」、recordsを「1」、maxrecordsを「100」として、あとは空白のままにします。

ページファイルの作成（ファイル名：page03.html）

1. VMのホームで、「page03.htmlを編集」をクリックするか、すでに開いている場合にはそのページを更新します。
2. ページファイルエディタで、以下のソースコードを変更します。

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>INTER-Mediator Demo</title>
  <link rel="stylesheet" type="text/css" href="INTER-Mediator/Samples/sample.css">
  <script type="text/javascript" src="def03.php"></script>
</head>
<body>
<div id="IM_NAVIGATOR"></div>
  <table>
    <tbody>
      <tr><td data-im="person_list@name"></td></tr>
    </tbody>
  </table>
  <table>
    <tbody>
      <tr>
        <th>名前</th>
        <td><input type="text" data-im="person_detail@name"/></td>
      </tr>
      <tr>
        <th>住所</th>
        <td><input type="text" data-im="person_detail@address"/></td></tr>
      <tr>
        <th>メール</th>
        <td><input type="text" data-im="person_detail@mail"/></td>
      </tr>
      <tr>
        <th>地域</th>
        <td><input type="text" data-im="person_detail@location"/></td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

```

    </tr>
    <tr>
      <th>分類</th>
      <td><input type="text" data-im="person_detail@category"/></td>
    </tr>
    <tr>
      <th>メモ</th>
      <td><textarea data-im="person_detail@memo"></textarea></td>
    </tr>
  </tbody>
</table>
</body>
</html>

```

3. VMのホームで、「page03.htmlを表示する」をクリックするか、すでに表示されているページを更新します。一覧の下に、なんとなく、編集フォームのようなものが登場しました。

iPadのようなナビゲーションを実現する

1. page03.htmlの編集ページを表示します。以下の様に、2つあるTABLEタグにstyle属性を設定します。また、最後のTABLE閉じタグのあとに、BRタグを追加します。

```

<table style="float: left">
  :
</table>
<table style="float: left; margin-left: 12px">
  :
</table>
<br clear="all"/>

```

2. VMのホームで、「page03.htmlを表示する」をクリックするか、すでに表示されているページを更新します。一覧と詳細が左右に分離しました。
3. VMのホームで、「def03.phpを編集」をクリックするか、すでに開いている定義ファイルエディタのページを参照します。
4. nameが「person_list」の方のnavi-controlを「master-hide」にします。
5. nameが「person_detail」の方のnavi-controlを「detail」にします。
6. VMのホームで、「page03.htmlを表示する」をクリックするか、すでに表示されているページを更新します。一覧と詳細を行き来するユーザインタフェースができあがりしました。
7. nameが「person_list」の方のnavi-controlを「master」にします。
8. VMのホームで、「page03.htmlを表示する」をクリックするか、すでに表示されているページを更新します。スプリットビュー的な表示になりました。

セッション3：伝票とリレーションシップ

1. def04.phpに、以下の様に3つのコンテキストの定義を行います。このセッションから、細かな手順の説明は行いません。定義ファイルエディタで開き、Show Allボタンを押して、全項目を表示しながら作業をしましょう。

```
name: invoice
key: id
paging: true
repeat-control: confirm-insert confirm-delete
records: 1
maxrecords: 100
Calculation:
  [field: total_calc, expression: format(sum(item@amount_calc))]
```

```
name: item
key: id
repeat-control: confirm-insert confirm-delete
records: 100
maxrecords: 100
Relationship:
  [foreign-key: invoice_id, join-field: id, operator: =] //←[MySQL]の場合
  [foreign-key: invoice_id, join-field: id, operator: eq] //←[FileMaker Server]の場合
Calculation:
  [field: amount_calc, expression: format(qty * product@unitprice)]
```

```
name: product
key: id
records: 100
maxrecords: 100
Relationship:
  [foreign-key: id, join-field: product_id, operator: =] //←[MySQL]の場合
  [foreign-key: id, join-field: product_id, operator: eq] //←[FileMaker Server]の場合
```

2. Database Settingsの設定は、前のセッションと同一にします。
 - c. [FileMaker Server] Database Settingsセクションのdb-classには「FileMaker_FX」、databaseには「TestDB」、userには「web」、passwordには「password」、serverには「192.168.56.1」、portには「80」、protocolには「http」、datatypeには「FMPro12」と入力します。
 - d. [MySQL] Database Settingsセクションのdb-classには「PDO」、dsnには「mysql:host=localhost;dbname=test_db;charset=utf8mb4」、userには「web」、passwordには「password」と入力します。
3. page04.htmlは以下の様に入力します。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title></title>
  <link rel="stylesheet" type="text/css" href="INTER-Mediator/Samples/sample.css">
  <script type="text/javascript" src="def04.php"></script>
</head>
<body>
<div id="IM_NAVIGATOR"></div>
```

```

<table border="1">
  <tbody>
    <tr>
      <th>id</th><td><input type="text" data-im="invoice@id"></td>
    </tr>
    <tr>
      <th>issued</th><td><input type="text" data-im="invoice@issued" value=""></td>
    </tr>
    <tr>
      <th>title</th><td><input type="text" data-im="invoice@title" value=""></td>
    </tr>
    <tr>
      <td colspan="2">
        <table border="1">
          <thead>
            <tr>
              <th>product</th><th>qty</th><th>unitprice</th><th>amount</th><th></th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>
                <input type="text" data-im="item@product_id" size="2">
                <span class="inline" data-im-control="enclosure">
                  <span class="inline" data-im-control="repeater"
                    data-im="product@name"></span>
                </span>
              </td>
              <td>
                <input class="price" type="text" data-im="item@qty" size="5">
              </td>
              <td>
                <span class="inline" data-im-control="enclosure">
                  <span class="inline" data-im-control="repeater"
                    data-im="product@unitprice"></span>
                </span>
              </td>
              <td align="right">
                <span align="right" data-im="item@amount_calc"></span>
              </td>
            <tr>
          </tbody>
        </table>
      </td>
    </tr>
    <tr>
      <th>Total:</th><td data-im="invoice@total_calc"></td>
    </tr>
  </tbody>
</table>
</body>
</html>

```

4. 実行します。以下の点を確認しましょう。

- ・ 伝票での明細行に相当する仕組みがリレーションシップにより取り出されています。
- ・ 明細の各行では、商品マスターからの商品名と単価の取り出しが行われています。
- ・ productの数字を書き換えると、商品マスターの違うレコードが取り出されます。
- ・ qtyを変更すると、単価x個数が計算され直され、トータルも再計算されています。

セッション4：資産管理

4-1：アプリケーションの動作の確認

このセッションでは、INTER-Mediatorのサンプルにあるアプリケーションの動作を調べて、アプリケーションの改変を行います。

1. VMのホーム (<http://192.168.56.101/> で表示されるページ) 開きます。
2. 「リンク」の中に、「サンプルプログラム」というリンク文字列があります。こちらをクリックします。
3. 「INTER-Mediator Samples」というページが表示されます。この中の、Asset Management Sampleという行を特定します。
4. MySQLあるいはFileMakerの列の「Improved」というリンクをクリックします。

どこかの会社の機材の管理データベースのようです。

まず、このファイルのファイル名をメモしてください。 _____

5. 「詳細」ボタンを押してみます。詳細部分が表示されました。
6. その機材の貸出履歴が出てきました。
7. 「追加」ボタンをクリックすると、現在の日付が「貸出日」に設定されて新たなレコードが追加されます。担当者やメモは適当に設定します。
8. 「本日返却」ボタンをクリックすると、返却日が空欄の貸出履歴に、本日の日付が設定されました。
9. いくつかのレコードをみて、動作を確認します。

4-2：ページファイルの確認

1. このアプリケーションのソースプログラムは、以下のURLで参照できます。

https://github.com/INTER-Mediator/INTER-Mediator/tree/master/Samples/Sample_Asset

2. ページファイルを開いて内容を参照してください。
3. 定義ファイルのファイル名はなんですか？ _____
4. それぞれのTABLEタグによるテーブル内で使用されているコンテキスト名はなんですか？
1つ目のTABLEタグ内 _____
2つ目のTABLEタグ内 _____
上記の内部のTABLEタグ内 _____
上記の内部のSELECTタグ内 _____
5. JavaScriptのプログラムにあるsetBackDate関数は、「本日返却」ボタンをクリックしたときに呼び出されます。以下の情報手掛かりにプログラムを解析しましょう。
 - ・ generateToday関数は現在の日付を文字列で返す関数です。
 - ・ IMLibContextPool.getContextFromName(contextName)は、引数に指定したコンテキスト名のコンテキストオブジェクトを返します。コンテキストオブジェクトは、そのコンテキストに対する取得したデータを保持している「モデル」相当のオブジェクトです。

- ・ コンテキストオブジェクトに対して、setDataAtLastRecord(field, value)を実行すると、コンテキストの最後のレコードのfieldの値をvalueにして、それを参照している箇所を更新するとともに、データベースへの書き込みを行います。

4-3：定義ファイルの確認

1. 以下のページに戻ります。

https://github.com/INTER-Mediator/INTER-Mediator/tree/master/Samples/Sample_Asset

2. 前に調べた名称の定義ファイルを開きます。VMのホームから定義ファイルエディタで開いてもかまいません。
3. navi-controlキーが設定されているコンテキストを確認します。実際の動作と比べて、このキーによる設定がどのように適用されているのかを検討してみましょう。
4. 使用されていないコンテキストがあります。どれでしょうか？ _____
5. 使用されていないコンテキストがあります。どれでしょうか？ _____
6. nameキーがrentのコンテキストの定義を参照してください。ここにあるrelationの設定により上位のコンテキストの値を元に検索を行っています。
7. rentコンテキストのcalculationキーにある計算式を参照してください。計算式の意味を考えてください。そして、そのfieldキーの値を書き出しておきます。
fieldキー： _____ 式の意味： _____
8. assetdetailコンテキストのcalculationキーにある計算式を参照してください。計算式の意味を考えてください。そして、そのfieldキーの値を書き出しておきます。
fieldキー： _____ 式の意味： _____

4-4：アプリケーションの改造

ページファイルエディタで開きます。こちらより開くことができます。

これまでに調べたことを応用して、次の設定を行ってください。

- ・ 貸出履歴のそれぞれの項目に、貸出期間の日数を表示してください。
- ・ それぞれの機材に対する平均貸出日数を表示してください。

セッション5：ブログ

5-1：定義ファイルとページファイルの作成

1. def05.phpに2つのコンテキストの定義を行います。定義ファイルエディタで開き、Show Allボタンを押して、全項目を表示しながら作業をしましょう。

```
name: message
table: chat
view: chat
key: id
query:
  [field: groupname , operator: IS NULL, value:] // ←[MySQL]の場合
  [field: groupname, operator: eq, value: =] // ←[FileMaker Server]の場合
sorting:
  [field: postdt, direction: desc]
post-reconstruct: true
post-dismiss-message: 投稿しました
validation:
  [field: user, rule: value != '', message: 入力してください。]
  [field: message, rule: value != '', message: 入力してください。]
```

```
name:comment
table: chat
view: chat
key: id
relationship:
  [foreign-key: groupname, join-field: id, operator: =] // ←[MySQL]の場合
  [foreign-key: groupname, join-field: id, operator: eq]
  // ←[FileMaker Server]の場合
post-reconstruct: true
post-dismiss-message: 投稿しました
sorting:
  [field: postdt, direction: desc]
validation:
  [field: user, rule: value != '', message: 入力してください。]
  [field: message, rule: value != '', message: 入力してください。]
```

2. 定義ファイルで開いたdef05.phpに対して、以下のようにOptionsの設定を行います。

```
formatters:
  [field: chat@message, converter-class: HTMLString]
```

3. Database Settingsの設定は、前のセッションと同一にします。
 - a. [FileMaker Server]
Database Settingsセクションのdb-classには「FileMaker_FX」、databaseには「TestDB」、userには「web」、passwordには「password」、serverには「192.168.56.1」、portには「80」、protocolには「http」、datatypeには「FMPro12」と入力します。
 - b. [MySQL]
Database Settingsセクションのdb-classには「PDO」、dsnには「mysql:host=localhost;dbname=test_db;charset=utf8mb4」、userには「web」、passwordには「password」と入力します。

4. page05.htmlは以下の様に入力します。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title></title>
  <script type="text/javascript" src="def05.php"></script>
  <script type="text/javascript">
    INTERMediatorOnPage.processingBeforePostOnlyContext = function (node) {
      var dtString, nodeIds, idValue, aNode;
      dtString = INTERMediatorLib.dateTimeStringISO(); // ←[MySQL]の場合
      dtString = INTERMediatorLib.dateTimeStringFileMaker(); // ←[FileMaker Server]の場合
      nodeIds = INTERMediatorOnPage.getNodeIdsHavingTargetFromNode(
        node, "message@postdt");
      if (nodeIds.length > 0) {
        document.getElementById(nodeIds[0]).value = dtString;
      }
      nodeIds = INTERMediatorOnPage.getNodeIdsHavingTargetFromNode(
        node, "comment@message");
      if (nodeIds.length > 0) {
        aNode = node.parentNode.parentNode.parentNode;
        nodeIds = INTERMediatorOnPage.getNodeIdsHavingTargetFromNode(
          aNode, "message@id");
        idValue = document.getElementById(nodeIds[0]).value;
        INTERMediator.additionalFieldValueOnNewRecord = {};
        INTERMediator.additionalFieldValueOnNewRecord['comment'] = [
          {field: "groupname", value: idValue},
          {field: "postdt", value: dtString}
        ];
      }
      return true;
    };
  </script>
  <style>
    TEXTAREA {
      width: 400px;
      height: 60px;
    }
  </style>
</head>
<body>
<table>
  <tbody data-im-control="post">
    <tr>
      <th>From:</th>
      <td>
        <input type="text" data-im="message@user">
        <input type="hidden" data-im="message@postdt">
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <textarea data-im="message@message"></textarea>
        <button data-im-control="post">投稿</button>
      </td>
    </tr>
  </tbody>
</table>
<table>
```

```
|  |  |  |  |
| --- | --- | --- | --- |
| --- | | | |
| From: |  | Date: |  |
|  | | | |
| Comment | | From:                                                                                                   | <input data-im="comment@user" type="text"/> | |---------------------------------------------------------------------------------------------------------|---------------------------------------------| | <input data-im="comment@message" type="text"/> <input data-im-control="post" type="button" value="投稿"/> |                                             |    | From: |  | Date: |  | |-------|--|-------|--| |       |  |       |  | | | |

```

5. 実行します。以下の点を確認しましょう。

- ・ 最初は単にフォームだけが表示されています。「投稿」ボタンをクリックしても、何も入力しないと警告が表示され、メッセージは追加されません。

- ・ Fromに自分の名前、messageに適当なメッセージを入力して「投稿」ボタンをクリックします。入力したメッセージが表示されましたが、同時に投稿ボタンを押した日時も設定されています。
- ・ ページの最初のテキスト領域を利用して、メッセージを3つほど入力します。メッセージは、日付の逆順に表示されています。
- ・ 2つ目のメッセージの直後にも、Formとメッセージの記入部分があります。こちらにも入力をして「投稿」ボタンをクリックします。こちらは、該当メッセージへのコメントになります。いくつかコメントを追加してください。コメントも、日付が自動的に設定され、日付の逆順で表示されます。異なるルートメッセージにコメントを記入し、正しいメッセージの後に付随していることを確認します。
- ・ メッセージはテキストエリアです。改行を含むメッセージを入力してみてください。正しく、改行も表示されています。
- ・ メッセージにJavaScriptのプログラムを、たとえば「`<script>alert(9)</script>`」のように記述して投稿してみます。タグはHTMLのタグとして解釈はされず、記述した通りの文字列で表示されます。
- ・ ページファイルには少し長いプログラムが入力されています。

INTERMediatorOnPage.processingBeforePostOnlyContextは、「投稿」ボタンを押した直後で、データベース処理を行う前に呼び出されます。何をしているのかを以下の手がかりをもとに検討してみましょう。

- ・ INTERMediatorLib.dateTimeStringISO()、
INTERMediatorLib.dateTimeStringFileMaker() : 現在の日時を返す
- ・ INTERMediatorOnPage.getNodeIdsHavingTargetFromNode(node, target) : node以下、指定されたtargetを持つ要素のid属性値を配列で返す
- ・ document.getElementById(idValue) : 指定したid属性値の要素への参照を得る
- ・ node.parentNode : nodeの親ノードへの参照を得る
- ・ INTERMediator.additionalFieldValueOnNewRecord : 新規レコードを作ったときの既定値を定める。最初のプロパティはコンテキスト名で、それに対してフィールド名と値を指定したオブジェクトの配列を与える

最小限の機能ですが、あっという間にBBSが完成しました。HTMLコードはコピペして作っていただきましたが、15分から30分くらいでできたのではないのでしょうか？ INTER-Mediatorを使えば、Webアプリケーションが簡単に構築できることがお分かりいただけたでしょうか？

5-2 : アプリケーションの改造

コメントの並び方を改良

先ほど作ったBBSのページを開きます。こちらより開くことができます。試しにメッセージを二つ以上投稿し、さらに、一番上に表示されているメッセージに二つ以上のコメントを投稿してみてください。

どうでしょうか。投稿されたコメントは読みやすい並びに表示されていますか？

メッセージは新しいものほど上の方に表示されていて話の流れについて行きやすくなっていますが、コメントも同じように上から新しいものが並んでいます。誰かの発言に関連するコメントは、古いものから順番

に読んで行かなければ話の流れをつかむことができません。Facebookなどのタイムラインも、メッセージは新しいものから、コメントは古いものから順に並んでいますよね。

では、これまで学んだことから、どこをどのような修正すればいいか、考えてみてください。

コメントを表示させているコンテキストは何でしょう？ _____

Sortingの方向 (direction) を降順に指定するパラメーターは英語のdescendingを略したdescでした。昇順は英語でascendingと言います。パラメータはどう指定すればいいでしょうか？ _____

実際に改修作業を進めて行きましょう。あるコンテキストの中に表示されるデータベース項目の順序は定義ファイルを設定することでコントロールできます。

1. INTER-Mediatorトライアルのトップページに戻ります。
2. 定義ファイル def05.php を特定し、クリックします。定義ファイルエディターが開きます。
3. コンテキスト「name: comment」を特定し、Sorting項目のdirectionを昇順を意味するascに変更します。タブキーを押して変更したことを定義ファイルエディターにしっかりと伝えてください。
4. では、再びBBSのページを表示させてください。一旦画面をリフレッシュします。どうでしょうか、メッセージの並びは登録日時の降順に、コメントは昇順になったでしょうか？

簡単な作業でデータベースから取得した項目の表示順を切替えることができましたね。他に何か不都合なことはありませんでしょうか？

コメント入力的位置をコメントの後に移動

メッセージにコメントを付け加えるための入力フィールドが、コメントの並びの一番上に位置しています。これはちょっと不便ですね。コメントは通常、たくさん付いたコメントの一番最新のコメントに対してさらに続けるという使い方が一般的です。そのような使い方をするとき、たくさん付いたコメントの一番先頭にコメント入力フィールドがあると、上下にスクロールさせながら入力しなければならなくなります。

これも、少し改造してみましよう。

コメントの入力フィールドの位置を、コメントの並びの最下段に移動させてみましょう。画面の表示場所などの指定はページファイルで行います。

1. INTER-Mediatorトライアルのトップページに戻ります。
2. ページファイル page05.html を特定し、クリックします。ページファイルエディターが開きます。
3. HTMLファイルの中身をざーっと眺めて、どの部分がどこの表示を担当しているのか、確認してください。コメントの詳細を表示している部分がどこなのか、分かりますね？ ヒントはコメントを投稿するためのテーブルタブ内に納められている、というところです。見つけましたか？ ではそのテーブルタグごと、位置を移動させてください。
4. さて、作業完了しましたね？ 再びBBSのページを表示させてください。一旦ブラウザーに見えている画面をリフレッシュします。どうでしょうか、コメントの入力フィールドはコメントの並びの最下段になったでしょうか？

INTER-Mediatorを使ったWebアプリケーション開発はいかがだったでしょうか？ このようにして基本の骨格を作り上げ、要望に応じて徐々に改良していく。とても効率的な開発プラットフォームですね。



おわりに

今日学んだことをベースに、こんなことはできないだろうか、あんな形式に変更はできないだろうか、といういろいろと工夫をしながら開発スキルを深めて行ってください。