実践４『登録画面（紐づけボタン付)』を作成する

事前準備

手順

画面の確認

確認する画面が４つあります。

『**一覧画面**』と『**登録画面**』に『**検索ボタンを押下した先の画面**』と『**紐付けボタンを押下した先の画面**』です。

①［DG_740R　一覧画面］

この一覧画面は実践１とは違い**データが２段に表示**されています。

２段表示ができるように作成します。

②［DG_750U　登録画面］

この画面も実践２とは違い、**入力欄が２段表示**になっています。

そして入力欄にセレクトBOXがあります。セレクトBOXの項目名はDBから持ってくるのではなく、記載が必要になる為、Html/データセットJSONで作成します。

③［ DG_720R_serect　検索ボタン押下先画面］

この画面は他の画面から ID と名称を持ってきて紐付けます。

実践 3 の様に紐付けデータを入れるテーブルはありません。

DG_720R_serect を作成しここから紐付けます。（DG_720R_serect は作成済）

④［DG_751U　紐付けボタン押下先画面］

この画面は実践 3 の紐付け機能を別画面で行うだけで仕組みは同じです。

実践 3 では 1 度にテーブル全てを更新しましたが、ここでは右のテーブルデータが紐つくところだけが更新できるようにプログラムを修正して作成します。

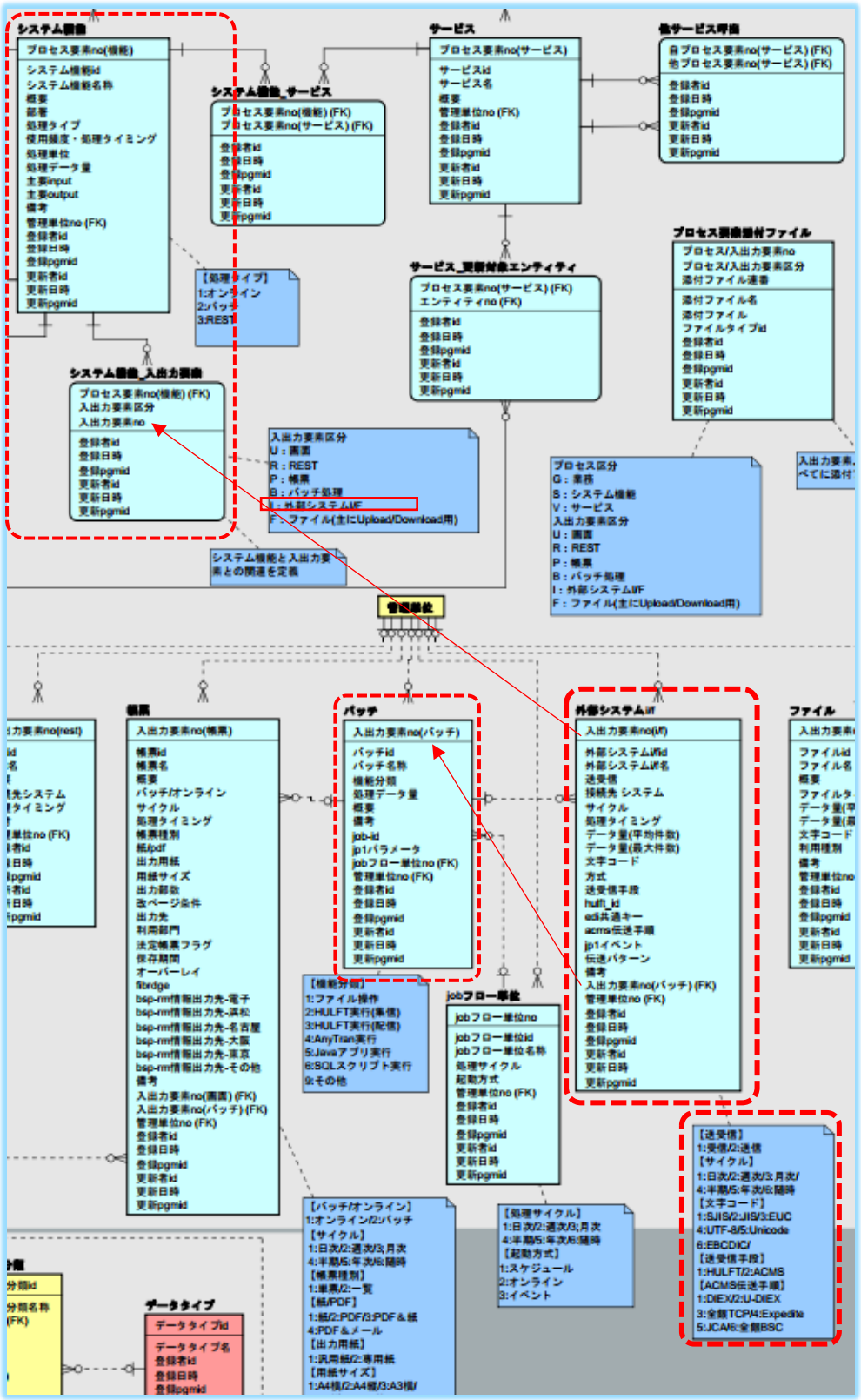
実践 4 では上記 4 つの画面を作成します。

ER 図を確認

この画面は『外部システムｉ/ｆ』です。

矢印の項目がそれぞれのテーブルと紐つきます。この 2 本が 2 つのボタンです。

下の青いテキストがテキストＢＯＸの項目名を指示しています。



プログラムのコピー

①［DG_740R　DG_750U］

実践 3 ではテーブルが 2 つある雛型からプログラムをコピーしました。

実践 1.2 ではテーブルが 1 つの雛型からプログラムをコピーしました。

実践 4 では実践 3 と同じ紐付け機能がありますが、

画面で確認したように、ボタンを作成し、別画面で紐付けを行うため、

実践 1.2 と同じ、テーブルが 1 つの雛型からコピーを実行します。

	雛型		**作成**
『一覧画面』	**DG_140R**	→	**DG_740R**
『登録画面』	**DG_150U**	→	**DG_750U**

②［ DG_720R_serect］

作成済をもらいました。

③［DG_751U］

画面の確認で確認したように、テーブルが 2 つあります。

テーブルが 2 つある実践 3 で作成した『登録画面（紐付け）』をコピーします。

DG_630U　→　DG_751U

クライアント側の作成

Html（一覧画面を２段表示を作成する）

①項目名を〈div〉〈/div〉で挟んで二段表示を作成する。

※使いやすいデザインを考え上段・下段・幅を作成する。

※一覧画面作成は実践１と同じで続きを作成する。

```
<div class="row">
  <ul class="nav nav-pills rmenu-ul"> <!-- menu-ul スタート -->
    <li class="rmenu-li"> <!-- menu-li スタート -->
    
            <div id="mainTableDiv">
              <table id="mainTable" name="mainTable" class="table rmenuTable">
                <thead>
                  <tr>
                    <th class="width50">
                      <div>印刷</div>
                      <input name="H印刷" id="H印刷" type="checkbox" class="rfocusblue rmenuCenter">
                    </th>
                    <th class="width180"><div>システム機能ＩＤ</div><div>システム機能名称</div></th>
                    <th class="width180"><div>外部システムＩ／ＦＩＤ</div><div>外部システムＩ／Ｆ名称</div></th>
                    <th class="width120"><div>送受信</div><div>接続先システム</div></th>
                    <th class="width120"><div>サイクル</div><div>処理タイミング</div></th>
                    <th class="width120"><div>データ量(平均件数)</div><div>データ量(最大件数)</div></th>
                    <th class="width120"><div>文字コード</div><div>方式</div></th>
                    <th class="width150"><div>送受信手段</div><div>ＨＵＬＦＴ＿ＩＤ</div></th>
                    <th class="width150"><div>ＡＣＭＳ伝送手順</div><div>ＥＤＩ共通キー</div></th>
                    <th class="width150"><div>ＪＰ１イベント</div><div>伝送パターン</div></th>
                    <th class="width200">備考</th>
                    <th class="width150"><div>バッチＩＤ</div><div>バッチ名称</div></th>
                  </tr>
                </thead>
                <tbody>
                  <tr>
                    <td class="width50 rmenuCenter">
                      <input name="印刷" type="checkbox" class="rfocusblue 印刷">
                    </td>
                    <td class="width180">
                      <div class="rmenuLeft システム機能ＩＤ"></div>
                      <div class="rmenuLeft システム機能名称"></div>
                    </td>
                    <td class="width180">
                      <div class="rmenuLeft 外部システムＩ／ＦＩＤ"></div>
                      <div class="rmenuLeft 外部システムＩ／Ｆ名"></div>
                    </td>
                    <td class="width120">
                      <div class="rmenuLeft 送受信"></div>
                      <div class="rmenuLeft 接続先システム"></div>
                    </td>
                    <td class="width120">
                      <div class="rmenuLeft サイクル"></div>
                      <div class="rmenuLeft 処理タイミング"></div>
                    </td>
                    <td class="width120">
                      <div class="rmenuLeft データ量平均件数"></div>
                      <div class="rmenuLeft データ量最大件数"></div>
                    </td>
                    <td class="width120">
                      <div class="rmenuLeft 文字コード"></div>
                      <div class="rmenuLeft 方式"></div>
                    </td>
                    <td class="width150">
                      <div class="rmenuLeft 送受信手段"></div>
                      <div class="rmenuLeft ｈｕｌｆｔＩＤ"></div>
                    </td>
                    <td class="width150">
                      <div class="rmenuLeft ａｃｍｓ伝送手順"></div>
                      <div class="rmenuLeft ｅｄｉ共通キー"></div>
                    </td>
                    <td class="width150">
                      <div class="rmenuLeft ｊｐ１イベント"></div>
                      <div class="rmenuLeft 伝送パターン"></div>
                    </td>
                    <td class="width200 rmenuLeft 備考"></td>
                    <td class="width180">
                      <div class=" rmenuLeft バッチＩＤ"></div>
                      <div class=" rmenuLeft バッチ名称"></div>
                    </td>
                    </tr>
                  </tbody>
                </table>
              </div>
```

Html（登録画面を 2 段表示を作成する）

①点線内を一覧と同じ様に二段表示にする。

②インプットを 2 段で作成する。

③セレクト BOX の項目名を作成する。

```
<ul class="nav nav-pills rmenu-ul"> <!-- menu-ul スタート -->
  <li class="rmenu-li"> <!-- menu-li スタート -->

      <div id="scrollable-HTable"> <!-- テーブル スクロール スタート -->
      <table id="mainTableH" class="table table-condensed rmenuTable">
        <thead>
          <tr>
            ①
            <th class="width50"><div> </div><div>追加</div><div> </th>
            <th class="width40">削除</th>
            <th class="width180"><div>外部システムＩ／ＦＩＤ</div><div>外部システムＩ／Ｆ名称</div></th>
            <th class="width120"><div>送受信</div><div>接続先システム</div></th>
            <th class="width120"><div>サイクル</div><div>処理タイミング</div></th>
            <th class="width140"><div>データ量(平均件数)</div><div>データ量(最大件数)</div></th>
            <th class="width140"><div>文字コード</div><div>方式</div></th>
            <th class="width200"><div>送受信手段</div><div>ＨＵＬＦＴ　ＩＤ</div></th>
            <th class="width200"><div>ＡＣＭＳ伝送手順</div><div>ＥＤＩ共通キー</div></th>
            <th class="width200"><div>ＪＰ１イベント</div><div>伝送パターン</div></th>
            <th class="width220">備考</th>
            <th class="width180"><div>バッチＩＤ</div><div>バッチ名称</div></th>
          </tr>
        </thead>
      </table>
      </div> <!-- テーブル スクロール エンド -->

      <div id="scrollable-DTable"> <!-- テーブル スクロール スタート -->
      <table id="mainTable" class="table table-condensed rmenuTable">
        <tbody>
          ②
          <td class="width50">
          <input name="行追加" type="button" class=" 行追加 btn color4 btn-sm rfocusblue" value="追加">
          </td>
          <td class="width40">
            <input name="削除" type="checkbox" class=" 削除 rfocusblue" value="削除">
          </td>
          <td class="width180">
              <div>
            <input name="外部システムＩ／ＦＩＤ" type="text" value="" class=" form-control input-sm rfocusblue rmenuLeft 外部システムＩ／ＦＩＤ">
              </div>
              <div>
            <input name="外部システムＩ／Ｆ名" type="text" value="" class=" form-control input-sm rfocusblue rmenuLeft 外部システムＩ／Ｆ名">
              </div>
          </td>
          <td class="width120">
            <div>
            <select name="送受信" class="form-control input-sm rfocusblue 送受信">
              <option value="">送受信選択</option>
              <option value="1">受信</option>
              <option value="2">送信</option>
            </select>
            </div>
            <div>
          <input name="接続先システム" type="text" value="" class="form-control input-sm rfocusblue rmenuLeft 接続先システム">
            </div>
          ③
          class="width120">
          iv>
            <select name="サイクル" class="form-control input-sm rfocusblue サイクル">
              <option value="">サイクル選択</option>
              <option value="1">日次</option>
              <option value="2">週次</option>
              <option value="3">月次</option>
              <option value="4">半期</option>
              <option value="5">年次</option>
              <option value="6">随時</option>
            </select>
            </div>
            <div>
          <input name="処理タイミング" type="text" class="form-control input-sm rfocusblue rmenuLeft 処理タイミング">
            </div>
          </td>
          <td class="width140">
            <div>
          <input name="データ量平均件数" type="text" class="form-control input-sm rfocusblue rmenuLeft データ量平均件数">
            </div>
```

CSS

実践 1 と同じ

Appspec

①②セレクト BOX の項目を作成する。上段をコピーし、項目名を変更して行を追加作成します。

```
    //-------------------------------------------------------------------------
    //  権限情報の設定（使用不可のロールを設定する）
    //-------------------------------------------------------------------------
    ,roleInfo: [
        ["#戻る",                   [""]]
        ,["#実行",                  [10, 50]]
        //ここから追加処理
        ,["#システム機能紐付け",    [""]]
    ]
    //バリデーションのイベントを定義する
    ,validationEvent: [
        ["#mainForm", "focus", "onFocus"]
        ,["#mainForm", "blur", "onBlur"]
    ]
    //その他セレクターイベントを定義する
    ,selectorEvent: [
        //ここから追加処理
        ["#検索サブシステム選択",   "change", "onChange検索サブシステム選択"]
        ,["#検索管理単位選択",      "change", "onChange検索管理単位選択"]
        ,[".削除",                  "change", "onChange削除"]
        ,[".送受信",                "change", "onChange送受信"]
        ,[".サイクル",              "change", "onChangeサイクル"]
        ,[".文字コード",            "change", "onChange文字コード"]
        ,[".送受信手段",            "change", "onChange送受信手段"]
        ,[".ａｃｍｓ伝送手順",      "change", "onChangeａｃｍｓ伝送手順"]
    ]
```

①

```
//-------------------------------------------------------------------------
//セレクトボックス□定義
//selectorid　　:HTMLのIDを指定する
//selectorname:HTMLのnameを指定する（テーブル内で使用する時に定義する）
//init　　　　　:
//initvalue　　:初期値（0）
//inithtml　　:初期表示（～選択）
//datasetid　　:datasetのID名を指定する
//valuename　　:datasetに定義されているキー値の項目名
//htmlname　　:datasetに定義されている表示用データの項目名
//change　　　:
//datasetid　　:datasetのID名を指定する
//valuename　　:選択したキー値を格納するdatasetの項目名
//htmlname　　:選択した表示値を格納するdatasetの項目名
//-------------------------------------------------------------------------
,selectbox1: [
    {
        selectorid: "検索サブシステム選択"
        ,selectorname: ""
        ,init:      {initvalue: "", inithtml: "サブシステム選択", dataset: "選択サブシステム", valuename: "選択サブシステムＮＯ", htmlname: "選択サブシステム名称"}
        ,change:    {dataset: "header", valuename: "検索サブシステムＮＯ", htmlname: "検索サブシステム名称"}
    }
]
,selectbox2: [
    {
        selectorid: "検索管理単位選択"
        ,selectorname: ""
        ,init:      {initvalue: "", inithtml: "管理単位選択", dataset: "選択管理単位", valuename: "選択管理単位ＮＯ", htmlname: "選択管理単位名"}
        ,change:    {dataset: "header", valuename: "検索管理単位ＮＯ", htmlname: "検索管理単位名"}
    }
]
```

←1と2はヘッダーのセレクトBOX

②

```
,selectbox3: [
    {
        selectorid: ""
        ,selectorname: "送受信"
        ,init:      {initvalue: "", inithtml: "送受信選択", dataset: "選択送受信", valuename: "選択送受信ＮＯ", htmlname: "選択送受信名"}
        ,change:    {dataset: "detail", valuename: "送受信", htmlname: ""}
    }
]
,selectbox4: [
    {
        selectorid: ""
        ,selectorname: "サイクル"
        ,init:      {initvalue: "", inithtml: "サイクル選択", dataset: "選択サイクル", valuename: "選択サイクルＮＯ", htmlname: "選択サイクル名"}
        ,change:    {dataset: "detail", valuename: "サイクル", htmlname: ""}
    }
]
,selectbox5: [
    {
        selectorid: ""
        ,selectorname: "文字コード"
        ,init:      {initvalue: "", inithtml: "文字コード選択", dataset: "選択文字コード", valuename: "選択文字コードＮＯ", htmlname: "選択文字コード名"}
        ,change:    {dataset: "detail", valuename: "文字コード", htmlname: ""}
    }
]
,selectbox6: [
    {
        selectorid: ""
        ,selectorname: "送受信手段"
        ,init:      {initvalue: "", inithtml: "送受信手段選択", dataset: "選択送受信手段", valuename: "選択送受信手段ＮＯ", htmlname: "選択送受信手段名"}
        ,change:    {dataset: "detail", valuename: "送受信手段", htmlname: ""}
    }
]
,selectbox7: [
    {
        selectorid: ""
        ,selectorname: "ａｃｍｓ伝送手順"
        ,init:      {initvalue: "", inithtml: "ａｃｍｓ伝送手順選択", dataset: "選択ａｃｍｓ伝送手順", valuename: "選択ａｃｍｓ伝送手順ＮＯ", htmlname: "選択ａｃｍｓ伝送手順名"}
        ,change:    {dataset: "detail", valuename: "ａｃｍｓ伝送手順", htmlname: ""}
    }
]
```

←3～7がインプットテーブルのセレクトBOX

コントローラ

Appspec で作成したイベントをコントローラで動かします。

①チェンジ イベント処理を作成する。

　チェンジ イベント処理をコピーし、［項目名］［selectbox の数字］を作成する。

②点線部分に［parentNode］をコピペし 1 つ増やす。

（2 段表示の場合 3 つ必要になる。）

［完成コントローラ］

```
//-------------------------------------------------↓
//送受信□チェンジイベント処理（追加処理）↓
//-------------------------------------------------↓
,onChange送受信: function(event) {↓
    $R.log("Controller onChange送受信 :start");↓

    //Win7・IE11の判定↓
    var status = this.isIE11OfWin7();↓
    if (status) {↓
        //var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex+1;↓
        var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓
    }↓
    else {↓
        var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓
    }↓
    this.model.onChangeTableSelectBox(event, row, this.appspec.selectbox3);↓

    $R.log("Controller onChange送受信 :end");↓
}↓
//-------------------------------------------------↓
//サイクル□チェンジイベント処理（追加処理）↓
//-------------------------------------------------↓
,onChangeサイクル: function(event) {↓
    $R.log("Controller onChangeサイクル :start");↓

    //Win7・IE11の判定↓
    var status = this.isIE11OfWin7();↓
    if (status) {↓
        //var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex+1;↓
        var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓
    }↓
    else {↓
        var row = event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓
    }↓
    this.model.onChangeTableSelectBox(event, row, this.appspec.selectbox4);↓

    $R.log("Controller onChangeサイクル :end");↓
}↓
//-------------------------------------------------↓
//文字コード□チェンジイベント処理（追加処理）↓
//-------------------------------------------------↓
,onChange文字コード: function(event) {↓
    $R.log("Controller onChange文字コード :start");↓
```

［コピー元コントローラ］

［parentNode］が一つ少ない状態。

コピー元の画面が 1 段表示の為 2 つ

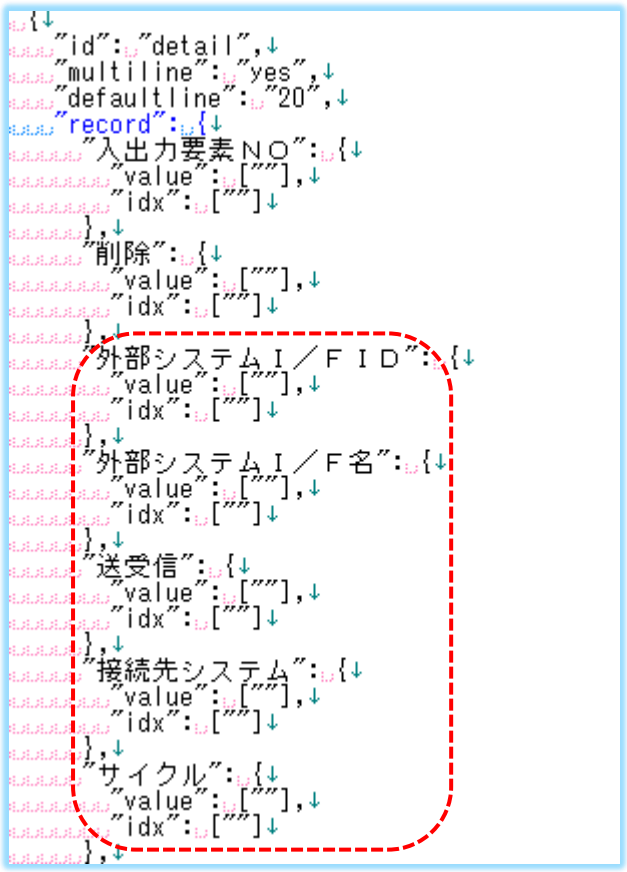
2 段表示の画面の場合は 3 つ必要になるので忘れないように注意

```
Controller.include({

    //----------------------------------------------------
    // 画面種類 チェンジ イベント処理 (追加処理)
    //----------------------------------------------------
    onChange画面種類: function(event) {
        $R.log("Controller onChange画面種類 : start");

        //Win7・IE11の判定
        var status = this.isIE11OfWin7();
        if (status) {
            //var row = event.currentTarget.parentNode.parentNode.rowIndex + 1;
            var row = event.currentTarget.parentNode.parentNode.rowIndex;
        }
        else {
        var row = event.currentTarget.parentNode.parentNode.rowIndex;
        }

        this.model.onChangeTableSelectBox(event, row, this.appspec.selectbox3);

        $R.log("Controller onChange画面種類 : end");
    }

```

サーバー側の作成

データセット JSON

①Ｈｔｍｌの項目名を作成（テーブルに項目名を作成する）



```
{
  "id": "detail",
  "multiline": "yes",
  "defaultline": "20",
  "record": {
    "入出力要素ＮＯ": {
      "value": [""],
      "idx": [""]
    },
    "削除": {
      "value": [""],
      "idx": [""]
    },
    "外部システムＩ／ＦＩＤ": {
      "value": [""],
      "idx": [""]
    },
    "外部システムＩ／Ｆ名": {
      "value": [""],
      "idx": [""]
    },
    "送受信": {
      "value": [""],
      "idx": [""]
    },
    "接続先システム": {
      "value": [""],
      "idx": [""]
    },
    "サイクル": {
      "value": [""],
      "idx": [""]
    },
```

②Appspec で作成した項目名を作成する

（ヘッダーのセレクト BOX の下へセレクト BOX を追加する。）

［**"comment"**］項目名+選択情報を作成

［**"id"**］項目名を作成

［**“record”**］Appspec②で作成した項目名を作成

『**ＮＯ**』にはセレクト BOX 内の**項目の数**を用意します

『**名**』にはセレクト BOX で選択できる**項目名を**用意します・

※『名』は画面で確認できる部分。『ＮＯ』はサーバー側でデータのやり取りは画面では見えない。

```
  },
  {
    "comment": "選択管理単位情報",
    "id": "選択管理単位",
    "multiline": "yes",
    "record": {
      "選択管理単位ＮＯ": {
        "value": [""],
        "idx": [""]
      },
      "選択管理単位名": {
        "value": [""],
        "idx": [""]
      }
    }
  },
  {
    "comment": "処理タイプ選択情報",
    "id": "選択処理タイプ",
    "multiline": "yes",
    "record": {
      "選択処理タイプＮＯ": {
        "value": ["1", "2", "3"],
        "idx": [""]
      },
      "選択処理タイプ名": {
        "value": ["オンライン", "バッチ", "ＲＥＳＴ"],
        "idx": [""]
      }
    }
  },
  {
    "comment": "送受信選択情報",
    "id": "選択送受信",
    "multiline": "yes",
    "record": {
      "選択送受信ＮＯ": {
        "value": ["1", "2"],
        "idx": [""]
      },
      "選択送受信名": {
        "value": ["受信", "送信"],
        "idx": [""]
      }
    }
  },
  {
    "comment": "サイクル選択情報",
    "id": "選択サイクル",
    "multiline": "yes",
    "record": {
      "選択サイクルＮＯ": {
        "value": ["1", "2", "3", "4", "5", "6"],
        "idx": [""]
      },
      "選択サイクル名": {
        "value": ["日次", "週次", "月次", "半期", "年次", "随時"],
        "idx": [""]
      }
    }
  }
```

← ヘッダーのセレクトBOX

←選択する項目名はDBから持ってくるタイプ

↓テーブル内のセレクトBOX

← 選択する項目名のＮＯを作成

↓ 選択する項目名を作成

← 選択する項目名のＮＯを作成

← 選択する項目名を作成

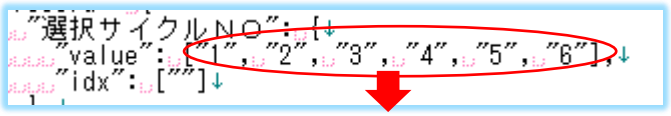
↓ 選択する項目名のＮＯを作成

↓ 選択する項目名を作成

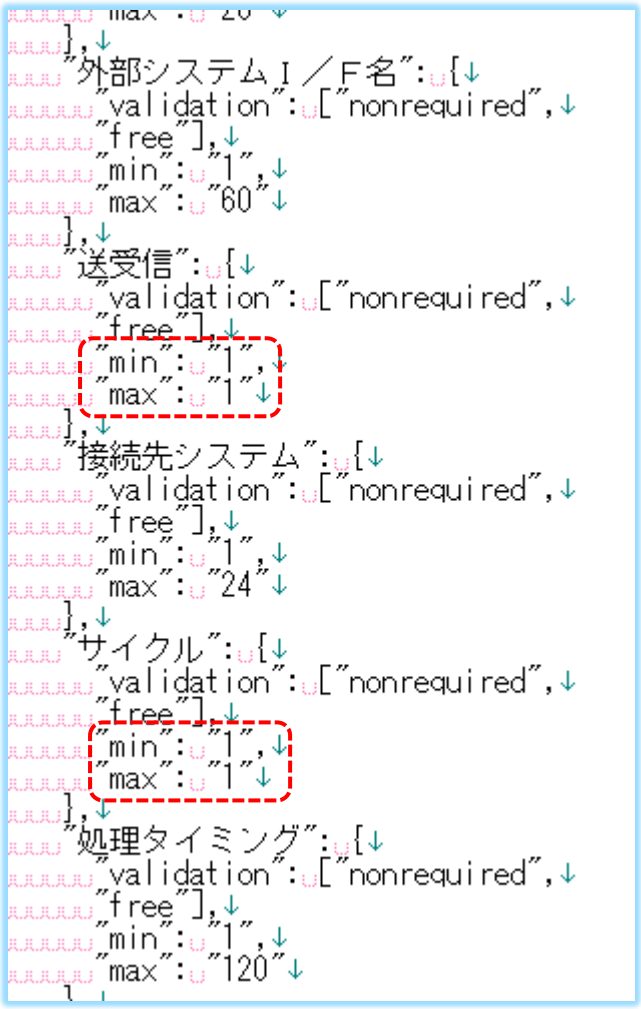
バリデーション JSON

①実践 1 と同じ（ポスグレで文字数を確認して作成する）

②セレクト BOX の項目名は文字が入らず、データセット JSON で作成した『ＮＯ』が入ります。



1 桁の数字で項目名を管理しているため、ここでは［ｍｉｎ］［ｍａｘ］ともに 1 が入ります。



※テーブルのセレクト BOX で変更が必要になるのは Html～データセット JSON・バリデーション JSON までです。

他は実践 2 と同じ手順です。

トラン JSON

①実践 1 と同じ

データセット JSON から項目名を作成する

※リクエストデータ・レスポンスデータに注意

SQLJSON

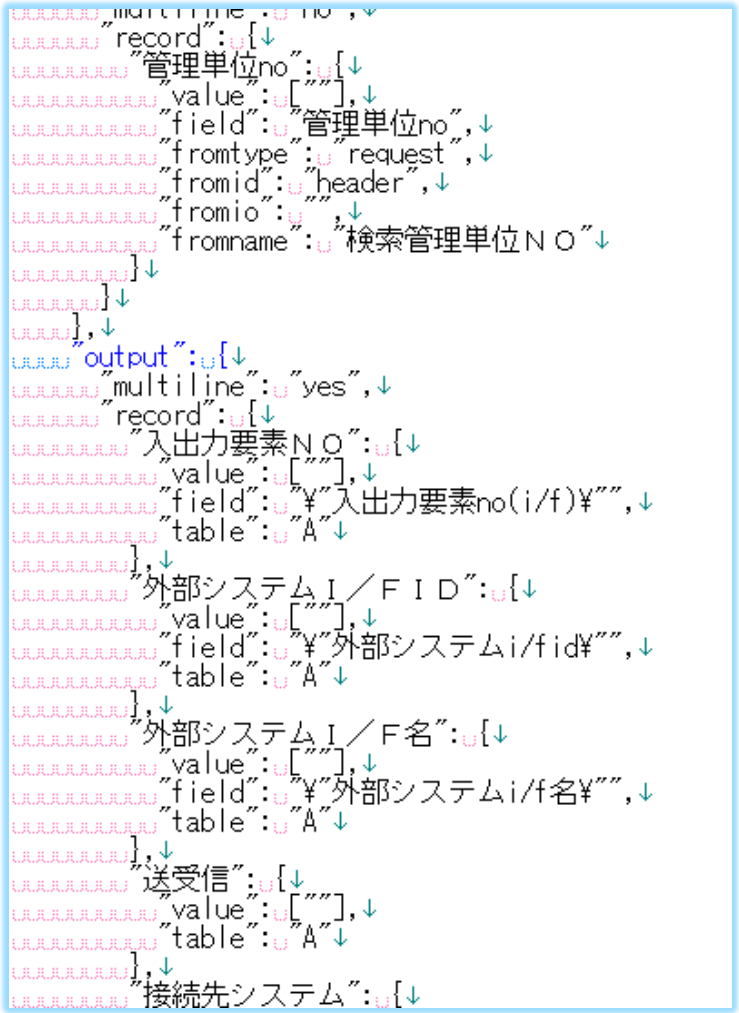
①selectSQL

※実践 2 と同じ

1.アウトプットの項目名をデータセットから作成する

2.ジェネレート SQL の為、［**"field"**］［**"table"**］を作成する

3.SQL の項目名を作成する



②updateSQL

※実践 2 と同じ

updateSQL の構成

```
"comment": "システム機能_入出力要素　削除処理",　（紐付けボタン先用）ここでは触れません
"comment": "外部システムＩ／Ｆ　削除処理",
"comment": "外部システムＩ／Ｆ　更新処理",
"comment": "外部システムＩ／Ｆ　新規処理",
"comment": "外部システムＩ／Ｆ　照会処理",
```

1.［照会処理］はアウトプットを作成する。

［更新・新規処理］はインプットの項目名を作成する。

［削除処理］は削除チェックで入る「入出力要素 NO」をインプットに作成する。

2.ジェネレート SQL の為、［**"field"**］［**"table"**］を作成する

3.SQL の項目名を作成する

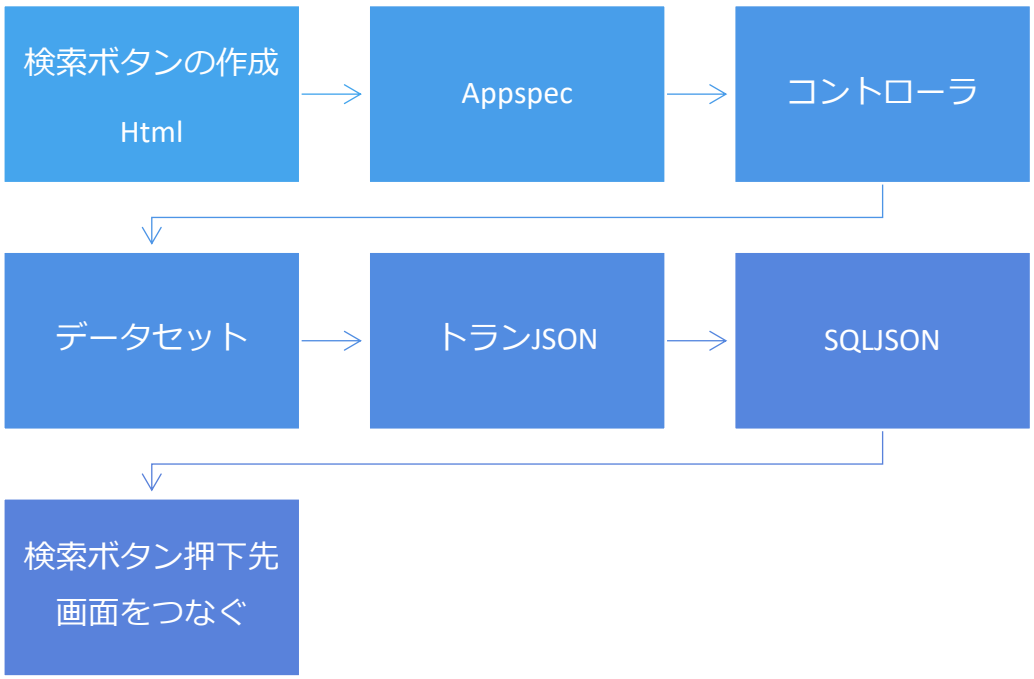
Ruby

※実践 2 と同じ

SQLJSON の before を確認し、使用している時のみ作成が必要。

検索ボタンを作成

手順


検索ボタンの作成
Html
Appspec
コントローラ
データセット
トランJSON
SQLJSON
検索ボタン押下先
画面をつなぐ

画面の確認

テーブル内に検索ボタンを作成します。

ボタンを押下すると次画面で持ってくるデータを選択し、紐付けることができます。

［次画面　紐付け選択画面］

Html

検索ボタンを作成します。

※実践 1 のボタン追加方法と同じです。

赤枠の様に［button］を作成します。

```
↓
        <div id="scrollable-HTable"> <!-- テーブル　スクロール　スタート -->↓
        <table id="mainTableH" class="table table-condensed rmenuTable">↓
          <thead>↓
            <tr>↓
              <th class="width50"><div>　</div><div>追加</div><div>　</th>↓
              <th class="width40">削除</th>↓
              <th class="width180"><div>外部システムＩ／ＦＩＤ</div><div>外部システムＩ／Ｆ名称</div></th>↓
              <th class="width120"><div>送受信</div><div>接続先システム</div></th>↓
              <th class="width120"><div>サイクル</div><div>処理タイミング</div></th>↓
              <th class="width140"><div>データ量(平均件数)</div><div>データ量(最大件数)</div></th>↓
              <th class="width140"><div>文字コード</div><div>方式</div></th>↓
              <th class="width200"><div>送受信手段</div><div>ＨＵＬＦＴ＿ＩＤ</div></th>↓
              <th class="width200"><div>ＡＣＭＳ伝送手順</div><div>ＥＤＩ共通キー</div></th>↓
              <th class="width200"><div>ＪＰ１イベント</div><div>伝送パターン</div></th>↓
              <th class="width220">備考</th>↓
              <th class="width180"><div>バッチＩＤ</div><div>バッチ名称</div></th>↓
            </tr>↓
          </thead>↓
        </table>↓
        </div> <!-- テーブル　スクロール　エンド -->↓
      ↓
        <div id="scrollable-DTable"> <!-- テーブル　スクロール　スタート -->↓
        <table id="mainTable" class="table table-condensed rmenuTable">↓
          <tbody>↓
```

```
          </div>↓
          <div>↓
        <input name="ｅｄｉ共通キー" type="text" class="form-control input-sm rfocusblue rmenuLeft　ｅｄｉ共通キー">↓
          </div>↓
        </td>↓
        <td class="width200">↓
          <div>↓
        <input name="ｊｐ１イベント" type="text" class="form-control input-sm rfocusblue rmenuLeft　ｊｐ１イベント">↓
          </div>↓
          <div>↓
        <input name="伝送パターン" type="text" class="form-control input-sm rfocusblue rmenuLeft 伝送パターン">↓
          </div>↓
        </td>↓
      <td class="width220">↓
        <textarea name="備考" rows="4" class="form-control input-sm rfocusblue rmenuLeft 備考"></textarea>↓
      </td>↓
        <td class="width180">↓
          <div class="width50">↓
            <button name="検索" class="検索 btn color4 btn-sm" type="button">検索</button>↓
          </div>↓
          <div>↓
        <input name="バッチＩＤ" type="text" class="form-control input-sm rfocusblue rmenuLeft バッチＩＤ">↓
          </div>↓
          <div>↓
        <input name="バッチ名称" type="text" class="form-control input-sm rfocusblue rmenuLeft バッチ名称">↓
          </div>↓
        </td>↓
      </tr>↓
</tbody>↓
```

Appspec

①ボタンのイベントを作成する。

```
//----------------------------------------------------------------↓
//イベント設定はセレクタ・イベント・コールバック関数の順に指定する↓
//----------------------------------------------------------------↓
//NAVボタンのイベントを定義する↓
,navButtonEvent: [↓
    ["#戻る",                "click", "on戻る"]↓
   ,["#実行",                "click", "on実行"]↓
   ,["#システム機能紐付け",  "click", "onシステム機能紐付け"]↓
   ,[".行追加",              "click", "on行追加クリック"]↓
   ,[".検索",                "click", "on検索"]↓
  ]↓
```

②Appspec でヘッダーの情報を引き継ぐ

［引き渡し］は次画面 DG_720R_serect へヘッダーで選択している情報を引き渡す。

［引き継ぎ］は次画面で紐付けたデータを持って帰ってテーブルに表示させます。

```
]↓
//----------------------------------------------------------------------
//次画面への引き継ぎデータ定義↓
//（次画面のセッションストレージに設定される）↓
//----------------------------------------------------------------------
,nextStorageData:[↓
  {↓
    datasetid:"header"↓
    ,dataname:["検索サブシステムＮＯ","検索管理単位ＮＯ"]↓
    ,validation:["required","required"]↓
    ,titlename:["サブシステム","管理単位"]↓
  }↓
]↓
//----------------------------------------------------------------------
//前画面からの引き継ぎデータ定義↓
//----------------------------------------------------------------------
,beforeStorageData:[↓
  {↓
    datasetid:"header"↓
    ,dataname:["検索サブシステムＮＯ","検索管理単位ＮＯ"]↓
    ,classname:["検索サブシステム選択","検索管理単位選択"]↓
    ,typename:["select","select"]↓
  }↓
]↓
//----------------------------------------------------------------------
//選択画面への引き渡しデータ定義（バッチ□選択画面）↓
//----------------------------------------------------------------------
,selectStorageRequestData:[↓
  {↓
    requestname:"DG_720R_SELECT"↓   ←次の選択画面のプログラム名
    ,datasetid:"header"↓            ←引き渡したいデータの場所はヘッダー
    ,dataname:["検索サブシステムＮＯ","検索管理単位ＮＯ←ヘッダーのセレクトBOXの項目2つを次の画面へ引き渡す
    ,value:["",""]↓                 ←上の引き渡すデータを移送する箱「""」の間に入る
  }↓
]↓
//----------------------------------------------------------------------
//選択画面からの引き継ぎデータ定義（バッチ□選択画面）↓
//----------------------------------------------------------------------
,selectStorageResponseData:[↓
  {↓
    responsename:"DG_750U"↓        ←元の画面に戻るためプログラム名を指示
    ,datasetid:"detail"↓           ←持って帰るデータの場所はディティールのデータ
    ,dataname:["入出力要素ＮＯバッチ","バッチＩＤ","バッチ名称"]↓
    ,value:["","",""]↓             ←持って帰るでーたの項目は3つ
  }↓                               ←持って帰るでーたの項目は3つあるので「""」が3つ必要
```

［検索ボタン使用時のデータの流れ方］

DG_740R一覧画面	DG_750U登録画面	DG_720R_select選択画面
登録ボタンを押下 →	検索ボタンを押下 →	選択し （ヘッダーのデータを引き継ぐ）
	データが入力されて表示 ←	選択ボタンを押下
登録内容が表示できる ←	実行を押下し確定	（ディティールデータを返す）

コントローラ

追加したボタンの動きを作成する。

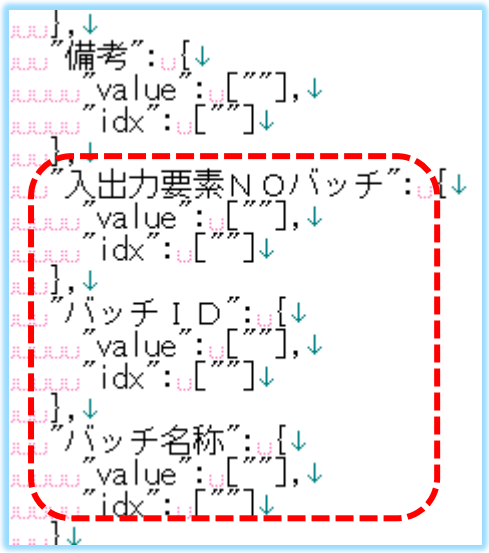
［ｏｎ検索］を作成する。

```
//----------------------------------------------↓
//　バッチ画面□検索処理（追加処理）↓
//----------------------------------------------↓
on検索:　function(event)　{↓
　　$R.log("Controller　on検索　:　start");↓

　　//　二行表示の為、parentNodeが一階層深い（注）↓
　　//　Win7・IE11の判定↓
　　var　status　=　this.isIE11OfWin7();↓
　　if　(status)　{↓
　　　　//var　row　=　event.currentTarget.parentNode.parentNode.parentNode.rowIndex　+　1;↓
　　　　var　row　=　event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓　←何段目に戻るかを指示している
　　}↓
　　else　{↓
　　　　var　row　=　event.currentTarget.parentNode.parentNode.parentNode.rowIndex;↓
　　}↓
　　↓
　　//　JOBフロー単位□選択画面へ遷移する↓
　　this.model.on選択画面遷移(this.appspec.selectStorageRequestData[0],　this.appspec.selectStorageResponseData[0],　row);↓

　　$R.log("Controller　on検索　:　end");↓
}↓
```

```
//----------------------------------------↓
//初期処理↓
//----------------------------------------↓
on初期処理: function() {↓
    $R.log("Controller on初期処理 : start");↓

    var arg = this.model.on初期処理();↓
    this.view.on初期処理(arg);↓
    ↓
    var status = this.model.on選択画面戻り();↓
    ↓
    //初期処理を実行するか、選択画面戻り処理を実行する判定する↓
    if (!status) {↓
        //初期処理：前画面の引き継ぎデータをデータセットに設定する↓
        var status1 = this.model.setFromBeforeStorageDataToDataset();↓
        if (status1 == "OK") {↓
            this.ajaxExecute("init");↓
        }↓
        ↓
        $R.log("Controller on初期処理 : end");↓
        return;↓
    }↓
    ↓
    //選択画面の戻り処理↓
    //データセットを使って、遷移前の画面を復元する↓
    var dataSet = this.model.dataset.getData();↓ ←初期処理に一行追加
    ↓
    //セレクトボックスの一括初期表示（検索サブシステム選択）↓
    this.onShowSelectBoxAll(dataSet, this.appspec.selectbox1);↓
    ↓
    //セレクトボックスの一括初期表示（検索管理単位選択）↓
    this.onShowSelectBoxAll(dataSet, this.appspec.selectbox2);↓
    ↓
    //ヘッダー部□復元↓
    this.view.on初期処理OfEditResponseData(dataSet, "", "");↓
    ↓
    //明細部□復元↓
    this.view.on照会OfEditResponseData(dataSet, "", "");↓

    $R.log("Controller on初期処理 : end");↓
}↓
```

データセット JSON

データセットＪＳＯＮへ待って帰ってきたデータが入るように項目名を作成する。



トラン JSON

データセット JSON を参考に作成する。

SQLJSON

①セレクトＪＳＯＮ

1.アウトプットに項目を作成する。

2.ジェネレート SQL の為、［**"field"**］［**"table"**］を作成する。

```
],↓
"入出力要素ＮＯバッチ":[↓
"value":[""],↓
"field":"¥"入出力要素no(バッチ)¥"",↓
"table":"E"↓
},↓
"バッチＩＤ":[↓
"value":[""],↓
"field":"バッチid",↓
"table":"E"↓
},↓
"バッチ名称":[↓
"value":[""],↓
"field":"バッチ名称",↓
"table":"E"↓
],↓
"更新日時":[↓
"value":[""],↓
"funct":"TO_CHAR(A.更新日時,'YYYY/MM/DD HH24:MI:SS.MS')",↓
"table":"A"↓
]↓
```

3.ＳＱＬを作成する。

次画面のテーブルを［"LEFT OUTER JOIN］でくっつける。

これがＡＳ→Ｅになる為、2.で作成した［**"table"**］は「**E**」になります。

```
},↓
"sqls":[[↓
"comment":"外部システムＩ／Ｆ照会処理",↓
"id":"detail",↓
"before":"",↓
"after":"",↓
"sql":[↓
"type":"select",↓
"freesql":"",↓
"genesql":[↓
"dist":"",↓
"from":↓
[
"¥"外部システムi/f¥"                    AS A"↓
,"LEFT OUTER JOIN バッチ AS E ON (A.¥"入出力要素no(バッチ)¥" = E.¥"入出力要素no(バッチ)¥")"↓
],↓
"where":"A.管理単位no = ?",↓
"order":"A.¥"外部システムi/fid¥""↓
]↓
},↓
"input":[↓
"multiline":"no",↓
"record":[↓
"管理単位no":[↓
"value":[""],↓
"field":"管理単位no",↓
"fromtype":"request",↓
"fromid":"header",↓
"fromio":"",↓
"fromname":"検索管理単位No"↓
```

※Ｅを忘れるとＡには項目名がないためエラーになります。

②アップデートＳＱＬ

アップデートＳＱＬでも赤字の照会処理は①と同じ作業を作成します。

updateSQL の構成

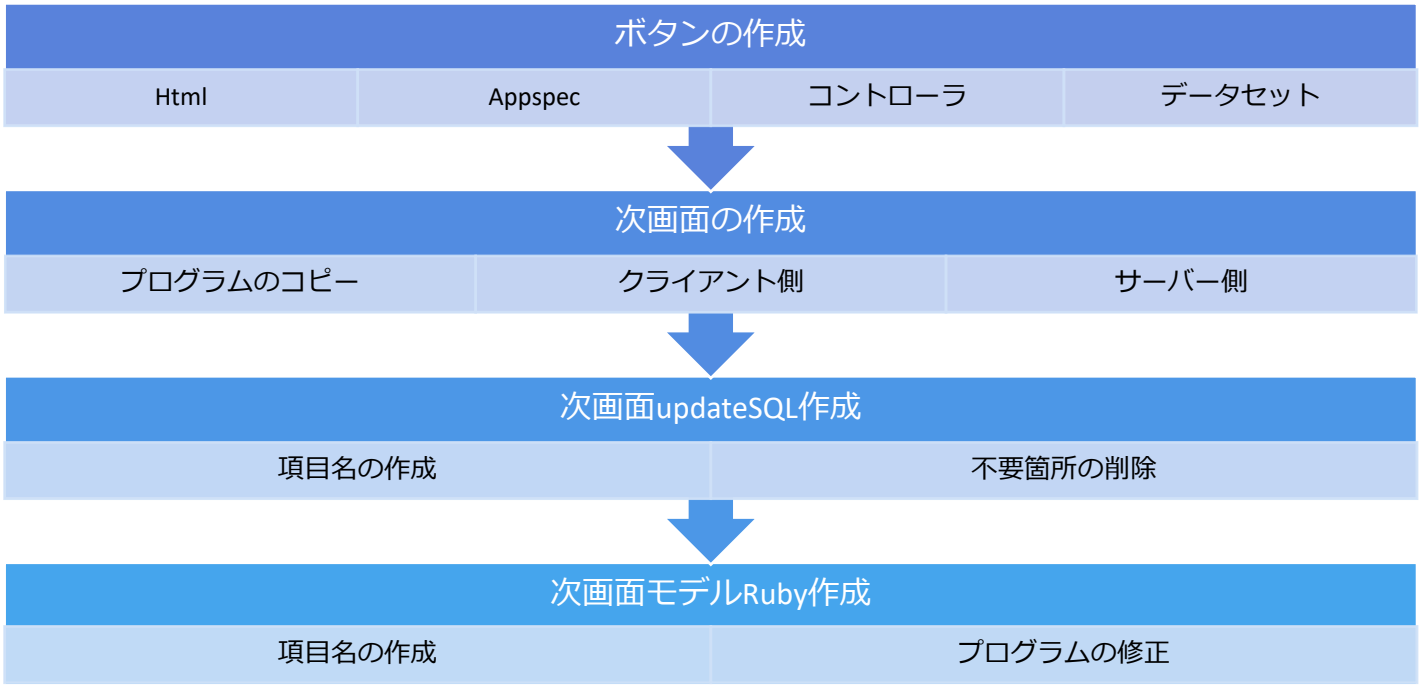
"comment": "システム機能_入出力要素　削除処理",　（紐付けボタン先用）ここでは触れません
"comment": "外部システムＩ／Ｆ　削除処理",
"comment": "外部システムＩ／Ｆ　更新処理",
"comment": "外部システムＩ／Ｆ　新規処理",
"comment": "外部システムＩ／Ｆ　照会処理",

持って帰ってきたデータを表示するため

（青字の処理には不要。インプットすることがないから）

紐付けボタンを作成

手順



画面の確認

実践 3 を作成します

違う部分はテーブル 1 は更新しず、赤枠内だけを更新するように、

実践 4 では updateSQL とモデル ruby を作成します。

［ボタンの確認］

作成中

［次画面　紐付け画面］

ボタンの作成

※手順は「検索ボタン作成」と同じです。

ここでは箇所の確認をします。

①Html

②Appspec

③コントローラ

④データセット

次画面の作成（DG_751U）

①プログラムのコピー

②クライアント側

③サーバー側

次画面の updateSQL 作成（DG_751U）

①項目名を作成する

②コピー元はテーブルが 2 つあり、テーブルのデータ更新と同時に紐付けができた

このプログラムは紐付けデータだけを更新したい為

テーブルのデータを更新してしまう updateSQL の不要部分を削除します。

工事中

次画面モデル ruby の作成（DG_751U）

工事中