DOWAホールディングス株式会社様基幹業務システム再構築(Step2)  
画面開発ガイドライン  
0.50版

改訂履歴

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版 | 改訂日 | 改訂箇所 | 改訂内容 | 作成者 | 承認者 |
| 0.50 | 2015/12/01 |  | 新規作成 | 駒田 |  |

目次

[1. はじめに 5](#_Toc441235522)

[1.1. 目的 5](#_Toc441235523)

[1.2. 対象読者 5](#_Toc441235524)

[1.3. 前提知識 5](#_Toc441235525)

[2. 開発環境 6](#_Toc441235526)

[2.1. 開発環境 6](#_Toc441235527)

[2.1.1. Intra-mart 7.2 Debug Server 6](#_Toc441235528)

[2.1.2. Orcale 11g 6](#_Toc441235529)

[2.1.3. Eclipse 6](#_Toc441235530)

[2.1.4. JDK 6](#_Toc441235531)

[2.2. ブラウザ 6](#_Toc441235532)

[3. アプリケーションの構成 7](#_Toc441235533)

[3.1. フレームワークの利用 7](#_Toc441235534)

[3.2. アプリケーション構成のレイヤー 7](#_Toc441235535)

[3.3. クライアント層の構成 7](#_Toc441235536)

[3.3.1. CSS 8](#_Toc441235537)

[3.3.2. CSJS 8](#_Toc441235538)

[3.4. プレゼンテーション層の構成 8](#_Toc441235539)

[3.4.1. JSP 10](#_Toc441235540)

[3.5. ビジネス層の構成 10](#_Toc441235541)

[3.6. インテグレーション層の構成 12](#_Toc441235542)

[4. アプリケーションの実装 14](#_Toc441235543)

[4.1. クライアント層 14](#_Toc441235544)

[4.1.1. 画面遷移の方法 14](#_Toc441235545)

[4.1.2. 登録や更新時のサーバー送信 15](#_Toc441235546)

[4.1.3. JsRenderの利用 15](#_Toc441235547)

[4.2. プレゼンテーション層 15](#_Toc441235548)

[4.2.1. Formクラスの実装について 15](#_Toc441235549)

[4.2.2. 基本マスタ、基本マスタ(取引先)の補完 17](#_Toc441235550)

[4.3. ロジック層 19](#_Toc441235551)

[4.4. サービス層 19](#_Toc441235552)

[4.4.1. サービス実装の構成 19](#_Toc441235553)

[4.4.2. 登録、更新者コード、登録更新日時等の補完 21](#_Toc441235554)

[4.4.3. Oracle Char型に対する対応 21](#_Toc441235555)

[4.4.4. Entity、ServiceGeneratedの自動生成 24](#_Toc441235556)

[4.4.5. 排他制御(SELECT ～ FOR UPDATE)メソッド 24](#_Toc441235557)

[4.5. 層をまたいだ制御 24](#_Toc441235558)

[4.5.1. Form、Dto、ModelのFiledのデータ型について 24](#_Toc441235559)

[4.5.2. データの移送について 25](#_Toc441235560)

[4.5.3. 一覧のページングについて 25](#_Toc441235561)

[4.5.4. ログインユーザ情報の利用について 31](#_Toc441235562)

[4.6. セキュリティ 31](#_Toc441235563)

[4.6.1. SQLインジェクション対策 31](#_Toc441235564)

[4.6.2. クロスサイト・スクリプティング対策 32](#_Toc441235565)

[4.6.3. クロスサイト・リクエストフォージェリ対策 32](#_Toc441235566)

[5. 実装イメージの解説 35](#_Toc441235567)

[5.1. 画面の初期表示 35](#_Toc441235568)

[5.1.1. 画面遷移イメージ 35](#_Toc441235569)

[5.1.2. ソースの構成 35](#_Toc441235570)

[5.1.3. プレゼンテーション層 36](#_Toc441235571)

[5.1.4. ロジック層 51](#_Toc441235572)

[5.1.5. サービス層 57](#_Toc441235573)

[5.2. 画面の登録／更新／削除機能 58](#_Toc441235574)

[5.2.1. 画面遷移イメージ 58](#_Toc441235575)

[5.2.2. ソースの構成 59](#_Toc441235576)

[5.2.3. クライアント層 59](#_Toc441235577)

[5.2.4. プレゼンテーション層 60](#_Toc441235578)

[5.2.5. ロジック層 68](#_Toc441235579)

[5.2.6. サービス層 72](#_Toc441235580)

[5.3. 単票画面の一覧表示機能 72](#_Toc441235581)

[5.3.1. 実装イメージ 72](#_Toc441235582)

[5.3.2. プレゼンテーション層(値の取得と、templateの展開) 72](#_Toc441235583)

[5.3.3. クライアント層 76](#_Toc441235584)

[5.3.4. プレゼンテーション層(一覧の更新データ取得) 77](#_Toc441235585)

[5.4. 一覧画面の検索機能 77](#_Toc441235586)

[5.4.1. 画面遷移イメージ 77](#_Toc441235587)

[5.4.2. ソースの構成 78](#_Toc441235588)

[5.4.3. クライアント層 79](#_Toc441235589)

[5.4.4. プレゼンテーション層 81](#_Toc441235590)

[5.4.5. ロジック層 89](#_Toc441235591)

[5.4.6. サービス層 93](#_Toc441235592)

# Mở đầu

Ở mục này ghi về mục đích của tài liệu này/đối tượng độc giả/kiến thức tiền đề.

## Mục đích

Tài liệu này nhằm mục đích giải thích việc implement tiêu chuẩn của chức năng màn hình sao cho có thể duy trì chất lượng của việc tái cấu trúc system nghiệp vụ then chốt (Step 2) (bên dưới sẽ gọi là system này) của cty TNHH Dowa Holdings (bên dưới sẽ gọi là phía DOWA), cũng như duy trì tính maintenance ở mức cố định, và từng developer có thể chú tâm vào việc implement chức năng nghiệp vụ.

Ở tài liệu này, tuy sẽ định nghĩa việc implement mang tính tiêu chuẩn của chức năng màn hình nhưng do những điều khoản nằm ngoài dự tính của tiêu chuẩn này mà việc implement bị ảnh hưởng có thể không được đánh giá đúng. Tuy không cần phải làm theo toàn bộ implement mang tính tiêu chuẩn nhưng trường hợp thực hiện implement khác với implement tiêu chuẩn thì sẽ tiến hành sau khi đã thảo luận đầy đủ. Ngoài ra, người phụ trách review code phải có kiến thức rộng hơn mức tiêu chuẩn để có thể tiến hành review một cách thận trọng.

## Đối tượng độc giả

Đối tượng của tài liệu này là: Application engineers, programmers, application architect mà sẽ cấu trúc system này.

## Kiến thức tiền đề

Tài liệu này sẽ lấy việc có những kiến thức liên quan tới cái ghi bên dưới làm tiền đề

　・intra-mart

　・Java

　・JavaScript

　・HTML

# Môi trường develop

Ở mục này ghi về môi trường develop khi develop màn hình.

## Môi trường develop

### Intra-mart 7.2 Debug Server

Khi develop sẽ sử dụng DebugServer của Intra-mart 7.2 với tư cách là server của AP. Đã cài vào pack môi trường develop và được sử dụng như tiền đề của pack môi trường develop.

### Orcale 11g

Khi develop thì dùng Oracle 11g. Download từ site Oracle, và install vào PC dùng để develop. Instance cấu trúc bằng UTF8.

### Eclipse

Khi develop dùng Eclipse với tư cách là IDE. Đã cài vào pack môi trường develop và được sử dụng như tiền đề của pack môi trường develop. Ngoài ra, đang ghi thiết lập CheckStyle (chẳng hạn) vào tài liệu khác (eclipse環境設定.xlsx). Nhất định phải xác nhận thiết lập rồi thực hiện mô tả.

### JDK

Khi develop thì dùng JDK 1.6.0\_30. Bản thân Eclipse sẽ thao tác trên luồng Java 8, build sẽ tiến hành trên luồng JDK 6. Cần chú ý kỹ về version JDK của cấu trúc run, và path build.

## Browser

Khi develop, test thì phải tiến hành xác nhận ở browser bên dưới

・Internet Explorer 11

※Những browser không ghi ở trên và browser dùng cho mobile thì không phải là đối tượng

# Cấu trúc application

Ở mục này sẽ ghi về cấu trúc màn hình khi develop màn hình.

## Sử dụng framework

Ở system này sẽ dùng framework SAStruts và S2JDBC (project Seaser) trên nền intra-mart 7.2 để develop màn hình.

Tham khảo (Công thức SAStruts)

<http://sastruts.seasar.org/>

Tham khảo (Công thức S2JDBC)

<http://s2container.seasar.org/2.4/ja/s2jdbc.html>

## Layer của cấu trúc application

Ở system này sẽ chia xử lý bên phía server thành 3 layer 「presentation layer」, 「business layer」, 「integration layer」 rồi quản lý. Ở system này sẽ thêm 「client layer」 vào đó để quản lý cấu trúc application với tư cách là 4 layer.



Hình 3‑1　Layer của cấu trúc application

## Cấu trúc của client layer

Client layer sẽ cấu trúc từ CSS(Cascading Style Sheets), CSJS(Client Side JavaScript). Phụ trách vai trò bên dưới.

・Vẽ màn hình và control màn hình

Client layer sẽ develop bằng ngôn ngữ bên dưới.

・HTML

・CSS

・JavaScript

Ở client layer, về mặt nguyên tắc là chỉ tiến hành control màn hình mang tính động. Còn control màn hình mang tính tĩnh (vd: convert mode hiển thị) thì thực hiện ở presentation layer.

### CSS

Về cơ bản là dùng CSS tiêu chuẩn intra-mart, or CSS common (common.css)

Lưu ý là design thống nhất trong toàn bộ system. Ngoài ra, nhờ vào việc chuẩn bị sắp xếp common CSS mà hầu như ở các màn hình không cần phải implement. Reference CSS common thì đang được commit vào Repository SVN. Vừa tham khảo reference vừa tiến hành implement.

Tham khảo (Reference CSS common)

<http://localhost:8081/imart/dowa/css_ref/index.jsp>

Trường hợp có design vốn có của màn hình thì tạo 1 file CSS ở 1 màn hình cũng được.

Mockup đã được tạo ở giai đoạn BD, DD thì chỉ là cái được tạo ra để xác nhận chức năng, design chứ không phải là cái được tạo ra để develop. Do đó, tuy nên lấy mockup làm tham khảo nhưng cũng phải implement theo design đúng.

### CSJS

Về mặt quy tắc là dùng Jquery để implement. Việc có thể thực hiện bằng JQuery là implement bằng Jquery. Do thống nhất thao tác ở toàn bộ system nên sẽ chuẩn bị CSJS common. Ngoài ra, khi cần thiết sẽ request việc add vào CSJS common. Về cơ bản là không ghi CSJS trên JSP. Phải tạo CSJS với tư cách là file ngoại bộ, rồi tiến hành implement. File ngoại bộ thì set 1 file đặt ở 1 màn hình. Ngoài ra, sẽ dùng namespace để tránh trùng lặp tên function trong các chức năng.

Đang chuẩn bị sắp xếp

Tham khảo (Reference CSJS common)

// TODO:

## Cấu trúc của presentation layer

Presentation layer sẽ cấu trúc từ Action. Phụ trách vai trò bên dưới.

・Tiến hành control mang tính common đối với request.

・Tiến hành kiểm chứng tính đúng đắn (validation) đối với giá trị input của user.

・Nhận yêu cầu từ user, rồi khởi động logic

・Cấu trúc màn hình thích hợp căn cứ vào kết quả logic

Presentation layer sẽ develop bằng ngôn ngữ bên dưới.

・Java



Hình 3‑2　Cấu trúc presentation layer

Presentation layer sẽ cấu trúc bằng Form, Action, Model, JSP như bên trên.

Đơn vị tạo sẽ như bên dưới.

Bảng 3‑1　Vai trò và đơn vị của presentation layer

|  |  |  |
| --- | --- | --- |
| Class chủ yếu | Vai trò | Đơn vị |
| Form | Class truyền giá trị input từ màn hình vào action | 1 đối với 1 action |
| Action | Xử lý request, yêu cầu xử lý đến business layer. Nhận kết quả xử lý rồi trả về | Nguyên tắc 1 đối với 1 màn hình |
| Model | Class truyền giá trị input từ action vào JSP | 1 đối với 1 màn hình |
| JSP | Định nghĩa layout màn hình | 1 đối với 1 màn hình |

Xử lý thật của presentation layer sẽ ghi vào class Action.

Do business layer, integration layer không phải là cái phụ thuộc vào màn hình nên về nguyên tắc là không access vào module JavaEE (HttpSession, HttpServletRequest, HttpServletResponse). Do đó, việc access vào module của những cái này sẽ là vai trò của 「Action」 và 「JSP」.

Về nguyên tắc, class Action sẽ implement theo tỷ lệ 1:1 đối với màn hình, còn xử lý button của màn hình (chẳng hạn) thì sẽ implement theo cách của class Action nhưng trường hợp class Action nhiều quá thì phân chia class Action cũng được.

### JSP

Dùng HTML để implement. Implement bằng HTML đúng.

Mockup đã được tạo ở giai đoạn BD, DD thì chỉ là cái được tạo ra để xác nhận chức năng, design chứ không phải là cái được tạo ra để develop. Do đó, tuy nên lấy mockup làm tham khảo nhưng cũng phải implement bằng markup đúng.

JSP dùng Apache Tiles để tiến hành cấu trúc hóa. Về việc dùng Apache Tiles để implement cấu trúc thì tham khảo 「xx\_JSPの実装について\_0.50.docx」

Ở system này có nhiều control mang tính common (Control hiển thị và dialog search master cơ bản chẳng hạn). Phần này thì thực hiện common hóa bằng taglib. Tham khảo reference rồi tiến hành implement.

Tham khảo (Reference Taglib common)

http://52.68.179.228/svn/1510\_hanbai\_sanpai/document/40\_製造（M）/01\_調査・検討/開発ガイドライン/タグライブラリ一覧.xlsx

## Cấu trúc business layer

Business layer cấu trúc từ Logic, Component. Phụ trách vai trò bên dưới.

・Tiến hành business logic (kiểm chứng và tính toán giá trị)

・Công khai interface của data access

Business layer sẽ develop bằng ngôn ngữ bên dưới.

・Java



Hình 3‑3 Cấu trúc logic layer

Logic layer sẽ cấu trúc bằng Dto, Logic, Component, ResultDto như trên.

Bên dưới là ghi chép về đơn vị tạo.

Bảng 3‑2 Vai trò và đơn vị của logic layer

|  |  |  |
| --- | --- | --- |
| Class chủ yếu | Vai trò | Đơn vị |
| Dto | Class truyền giá trị input từ action vào logic | 1 đối với 1 method của logic |
| Logic | Xử lý business logic. Gọi integration layer khi cần thiết. | ※1 |
| Component | Component khi cấu trúc business logic | Tùy ý |
| ResultDto | Class truyền giá trị trả về từ logic vào action. | 1 đối với 1 method của logic |

※Đơn vị của 1 Logic là 「1 nghiệp vụ」. Đơn vị của method Logic là 「1 thao tác」 đối với 1 nghiệp vụ. Ví dụ, trường hợp nghiệp vụ 「車両登録」, set là 1 Logic ở nghiệp vụ 「車両登録」. Set 「車両登録」 là method trong Logic khi 「登録する (đăng ký)」, 「削除する (delete)」, 「検索する (search)」

Logic sẽ tạo bằng đơn vị nghiệp vụ. Do đó, Logic sẽ dễ mở rộng class.

Class Logic sẽ được nghĩ như cổng vào của business logic, còn xử lý thật thì implement trong class Component. Class Component thì tạo bằng đơn vị tùy ý của developer cũng được.

## Cấu trúc của integration layer

Integration layer thì được cấu trúc từ Service, và phụ trách vai trò bên dưới.

・Access vào database

Integration layer thì develop bằng ngôn ngữ bên dưới.

・Java



Hình 3‑4 Cấu trúc của integration layer

Integration layer sẽ cấu trúc bằng Entity, Condition, Service như bên trên

Bên dưới là ghi chép về đơn vị tạo.

Bảng 3‑3 Vai trò và đơn vị của integration layer

|  |  |  |
| --- | --- | --- |
| Class chủ yếu | Vai trò | Đơn vị |
| Entity | Class truyền giá trị input từ logic vào service (cấu trúc tương tự table, data get) | 1 đối với 1 table  1 đối với 1 SQL |
| Condition | Class truyền giá trị input từ logic vào service (Conditional phrase...) | 1 đối với 1 SQL |
| Service | Interface cung cấp database access | 1 đối với 1 table |

Việc truy vấn vào DB tuy dùng S2JDBC nhưng ngoài class Service ra thì sẽ không dùng JDBCManager. Ngoài ra, sẽ cho phép 「Interface giống như flow」 ở class Service bằng cấu trúc đơn giản khi run trong vòng 10 method. Tuy nhiên, sẽ không phải là việc đánh giá đơn giản. Ngoài ra, trường hợp vượt quá 10 dòng thì tiến hành implement bằng cách tạo file SQL.

Tham khảo (Interface giống như flow và de-CoC)

<http://s2container.seasar.org/2.4/ja/s2jdbc_abstract.html#%E6%B5%81%E3%82%8C%E3%82%8B%E3%82%88%E3%81%86%E3%81%AA%E3%82%A4%E3%83%B3%E3%82%BF%E3%83%BC%E3%83%95%E3%82%A7%E3%83%BC%E3%82%B9%E3%81%A8%E8%84%B1CoC>

# Implement application

Ở mục này sẽ ghi về cách implement application.

## Client layer

Ở mục này sẽ ghi về cách implement vốn có của client layer.

### Cách di chuyển màn hình

Di chuyển màn hình sẽ dùng submit của Form hoặc CSJS mà về cơ bản đã được implement với tư cách common part để di chuyển.

Bên dưới là ví dụ về việc di chuyển màn hình bằng submit của Form

<script>

// 本来は、CSJSファイルに記載する

　function transfer() {

$(‘#form’).submit();

}

</scrip>

<form method=”POST” action=”URL”id=”form”>

</form>

Bên dưới là di chuyển màn hình bằng CSJS đã được implement với tư cách common part.

<script>

// 本来は、CSJSファイルに記載する

　function transfer() {

var data = {};

data.parameterA = ‘AAAA’;

dowa.sd.core.request.transitionTo($(‘#form’).attr(‘action’), data);

}

</scrip>

<form method=”POST” action=”URL”id=”form”>

</form>

### Việc gửi server khi đăng ký và update

Việc gửi server khi đăng ký và update sẽ dùng CSJS mà về cơ bản đã được implement với tư cách common part để di chuyển.

<script>

// 本来は、CSJSファイルに記載する

　function onClickRegisterBtn() {

// 登録用FORMのactionをFormにコピーする

$('#form).attr('action', $('#editForm').attr('action'));

// FORMと登録後の遷移先を引数に、共通部品を呼び出す

dowa.sd.core.request.ajaxSubmit($('#form'), $('#readForm').attr('action')); }

</scrip>

<form method=”POST” id=”form”>

<input type=”text”name=”aaaa” value=”AAAA”/>

</form>

<form method=”POST” action=”登録ActionURL”id=”editForm”>

</form>

<form method=”POST” action=”遷移先ActionURL”id=”readForm”>

</form>

### Sử dụng JsRender

Phần list sẽ dùng JsRender (Template engine của CSJS) để implement

Tham khảo (Công thức JsRender)

<http://www.jsviews.com/>

Cách implement thì tham khảo giải thích của image implement.

## Presentation layer

Ở mục này sẽ ghi về cách implement vốn có của presentation implement.

### Liên quan đến việc impement class Form

Form sẽ tiến hành implement theo tiêu chuẩn bên dưới.

■Field ứng với item màn hình thì định nghĩa trong class Form theo trình tự image bên dưới.



Hình 4‑1　Thứ tự field của class Form

Lí do là vì Validator được chạy theo thứ tự Field của class Form cho nên sẽ output theo thứ tự đã quyết định error của field.

■Annotation của Validator sẽ theo thứ tự tăng dần của list Validator

/\*\* 製品部門 . \*/

Xếp theo thứ tự tăng dần của list Validator

@Required(arg0 = @Arg(key = "jp.co.dowa.sd.sd04211.caption.003") )

@Maxbytelength(maxbytelength = 3, arg0 = @Arg(key = "jp.co.dowa.sd.sd04211.caption.003") )

public String spart;

Lí do là vì Validator được chạy theo thứ tự đã chỉ định trong Field cho nên sẽ thống nhất thứ tự theo từng phân loại Validator như 「Bắt buộc」「Check format」 (chẳng hạn)

■ Field không ứng với item màn hình (item hidden) thì sẽ ghi chép tổng hợp ở bên dưới cái tương ứng với item màn hình. Lý do là sẽ dễ chỉnh sửa khi maintenance item màn hình.

■Cung cấp annotation ToString của lombok vào class Form

@ToString

public class EditForm implements Serializable {

Do sẽ output log khi phát sinh error.

### Bổ sung master cơ bản, master cơ bản (Khách hàng)

Có thể dùng chức năng 「基本マスタ補完」 sao cho có thể dùng Master cơ bản, master cơ bản (Khách hàng) một cách đơn giản.

Bổ sung master cơ bản, master cơ bản (khách hàng) sẽ được tiến hành ở thời điểm bên dưới

・Sau khi thao tác method public ở **Interceptor** trước và sau khi gọi class Action.

・Ngay trước khi convert tham số thành chuỗi ký tự JSON, rồi ghi vào *HttpServletResponse* ở *ResponseUtils#write*

Chức năng 基本マスタ補完 (bổ sung master cơ bản) sẽ bổ sung nội dung của基本マスタ, 基本マスタ(取引先) bằng annotation bên dưới.

Bảng 4‑1 Annotation dùng để bổ sung 基本マスタ, 基本マスタ(取引先)

|  |  |  |
| --- | --- | --- |
| Tên annotation | Kiểu data mà annotation có thể cung cấp | Ứng dụng |
| BaseMasterSupplementInfo | List<HashMap<String, String>> | Pulldown list, checkbox, radio button |
| BaseMasterSupplementName | String | Tên gọi khi input search (chẳng hạn) |

#### BaseMasterSupplementInfo

Nội dung của master cơ bản sẽ được bổ sung bằng cách thiết lập annotation như bên dưới.

/\*\* 運搬事業者プルダウン. \*/

@BaseMasterSupplementInfo(cdtyp = "001", spart = "01", targetkbn = "1", valueField = "ujscd",

nmflg = true)

private List<Map<String, String>> ujsCdSelect;

Annotation này chỉ có thể áp dụng kiểu List<Map<String, String>>. Kiểu này có thể thiết lập y nguyên vào dowaui:select、dowaui:checkbox、dowaui:radio

BaseMasterSupplementInfo có tham số như bên dưới.

Bảng 4‑2　Tham số của annotation BaseMasterSupplementInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc／Tùy ý |
| targetkbn | 取得区分(1:基本マスタ、2:基本マスタ(得意先) | String | Bắt buộc |
| spart | 製品部門 | String | Bắt buộc |
| kunnr | 得意先コード (Trường hợp 取得区分 là 2:基本マスタ(得意先) thì bắt buộc) | String | Tùy ý |
| cdtyp | マスタ種別 | String | Tùy ý |
| ms1kb | 区分1 | String | Tùy ý |
| ms2kb | 区分2 | String | Tùy ý |
| ms3kb | 区分3 | String | Tùy ý |
| ms4kb | 区分4 | String | Tùy ý |
| ms5kb | 区分5 | String | Tùy ý |
| bsdtField | 検索基準日 (trường hợp chưa input thì set date system là 検索基準日) | String | Tùy ý |
| nmflg | 名称フラグ (true: get tên gọi、false: get tên tắt、default là false) | boolean | Tùy ý |
| valueField | Tên Field có giá trị được thiết lập (dùng để thêm vào selected...) | String | Bắt buộc |

#### BaseMasterSupplementName

Nội dung của master cơ bản sẽ được bổ sung bằng cách thiết lập annotation như bên dưới.

/\*\* 車種名称. \*/

@BaseMasterSupplementName(spart = "ZZ", cdtyp = "12", ms1kb = "5", targetkbn = "1",

valueField = "syscd")

private String mstnm;

Annotation này chỉ áp dụng kiểu String. BaseMasterSupplementName có tham số như bên dưới

Bảng 4‑3 Tham số của annotation BaseMasterSupplementName

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc／Tùy ý |
| targetkbn | 取得区分(1:基本マスタ、2:基本マスタ(得意先) | String | Bắt buộc |
| Spart | 製品部門 | String | Bắt buộc |
| Kunnr | 得意先コード (trường hợp 取得区分 là 2:基本マスタ(得意先) thì bắt buộc) | String | Tùy ý |
| Cdtyp | マスタ種別 | String | Tùy ý |
| ms1kb | 区分1 | String | Tùy ý |
| ms2kb | 区分2 | String | Tùy ý |
| ms3kb | 区分3 | String | Tùy ý |
| ms4kb | 区分4 | String | Tùy ý |
| ms5kb | 区分5 | String | Tùy ý |
| bsdtField | 検索基準日 (trường hợp chưa input thì setsystem date là 検索基準日) | String | Tùy ý |
| Nmflg | 名称フラグ (true: get tên gọi、false: get tên tắt、default là false) | boolean | Tùy ý |
| valueField | Tên Field có giá trị được thiết lập (thiết lập với tư cách là mstCd rồi get giá trị) | String | Bắt buộc |

## Logic layer

Ở mục này ghi về cách implement vốn có của logic layer.

## Service layer

Ở mục này sẽ ghi về cách implement vốn có của service layer

### Cấu trúc implement service

Bên dưới là quan hệ kế thừa của Service

Hình 4‑2　Cấu trúc implement service

Class Service của các table sẽ implement theo dạng kế thừa ServiceGenerated. Tự động generate method bên dưới vào ServiceGenerated

Bảng 4‑3　Method được tự động generate vào ServiceGenerated

|  |  |
| --- | --- |
| Method | Giải thích |
| findById | Tiến hành Select bằng PrimaryKey |
| lockByIdUpdateDatetime | Tiến hành Select For Update bằng PrimaryKey và date update. Không trả về kết quả Select. |
| findAndLockByIdUpdateDatetime | Tiến hành Select For Update bằng PrimaryKey và date update. Trả về kết quả Select. |
| findAllOrderById | Set PrimaryKey là thứ tự sort, rồi get toàn bộ record |

ServiceGenerated sẽ xếp vào bên dưới src/main/generated sao cho có thể ghi đè khi column của table đã bị thay đổi. Các Service thì sẽ xếp vào bên dưới src/main/java sao cho không được ghi đè.

Bảng 4‑4　Method AbstractServiceGenerated

|  |  |
| --- | --- |
| Method | Giải thích |
| updateExcludesNull | Tiến hành Update theo kiểu đã loại bỏ item Null |
| insertExcludesNull | Tiến hành Insert theo kiểu đã loại bỏ item Null |
| update | Tiến hành Update bao gồm cả item Null |
| insert | Tiến hành Insert bao gồm cả item Null |
| getPagingResultList | ページング用一覧取得 |

### Bổ sung 登録 (đăng ký), 更新者コード (code người update), 登録更新日時 (ngày giờ update đăng ký)

Item bên dưới sẽ tự động bổ sung giá trị của Entity khi dùng method đăng ký, update của AbstractServiceGenerated.

Bảng 4‑5　Field đối tượng bổ sung

|  |  |
| --- | --- |
| Field | Giải thích |
| trscd | 登録者コード |
| trkid | 登録機能ＩＤ |
| trdat | 登録日時 |
| ksscd | 更新者コード |
| kskid | 更新機能ＩＤ |
| ksdat | 更新日時 |

Trường hợp không dùng method đăng ký, update của AbstractServiceGenerated thì vui lòng dùng jp.co.dowa.sd.core.helper.DowaSdEntityHelper để bổ sung.

Ở xử lý batch (chẳng hạn) sẽ có trường hợp ログインユーザ (user login) ≠ 更新者コード (code người update)

### Đối ứng đối với kiểu Oracle Char

Kiểu Oracle Char là kiểu chuỗi ký tự có chiều dài cố định. Đối với item đăng ký, update, trường hợp chiều dài chưa đạt chiều dài cố định được chỉ định thì sẽ được padding space ở phía Database nhưng khi Bind giá trị vào cụm Where, nếu chỉ định giá trị chưa đạt chiều dài cố định thì space sẽ không được padding, và sẽ không match với tư cách điều kiện.

Do đó, sẽ tiến hành padding space đối với giá trị input ở Interceptor trước khi run method của Service, hoặc, tiến hành xử lý trim ở Interceptor sau khi run method của Service.



Hình 4‑3　Thao tác của OracleCharInterceptor

OracleCharInterceptor sẽ tiến hành padding, trim space đối với Field bên dưới. Lưu ý khi implement, rồi thực hiện chỉ định.

#### Ở class cung cấp annotation Entity, thì đây là Field cung cấp annotation Column, và chỉ định columnDefinition=”char(X)”

Trường hợp column của Database là kiểu Char thì sẽ được generate tự động khi output Entity. Chú ý trường hợp tạo class Condition mà không được generate tự động.

/\*\*

\* 車両マスタエンティティクラス

\*

\* @author $Author: $

\* @version $Revision: $ $Date: $

\*/

@Entity

@Generated(value = {"S2JDBC-Gen 2.4.46", "org.seasar.extension.jdbc.gen.internal.model.EntityModelFactoryImpl"})

public class Zswma0 implements Serializable {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\* 製品部門 \*/

@Id

@Column(name = "SPART", columnDefinition = "char(2)", nullable = false, unique = false)

public String spart;

#### Kiểu String cung cấp annotation CharType ở tham số Service

Trường hợp column Database là kiểu Char thì sẽ được generate tự động khi output ServiceGenerated.

Lưu ý khi add method vào class Service.

public abstract class Zswma0ServiceGenerated extends AbstractServiceGenerated<Zswma0> {

/\*\*

\* 識別子でエンティティを検索します。

\*

\* @param spart

\* 識別子

\* @param ujscd

\* 識別子

\* @param syacd

\* 識別子

\* @param vehic

\* 識別子

\* @return エンティティ

\*/

public Zswma0 findById(@CharType(2) final String spart, @CharType(10) final String ujscd, @CharType(4) final String syacd, @CharType(1) final String vehic) {

return this.select().id(spart, ujscd, syacd, vehic).getSingleResult();

}

### Generate tự động Entity, ServiceGenerated

Tuy Entity, ServiceGeneraterd dùng 「s2jdbc-gen」 để output nhưng vui lòng output từ project bên dưới. Cho customize vào chính S2jdbc-gen. Ngoài ra, đang chỉnh sửa file template.

<http://52.68.179.228/svn/1510_hanbai_sanpai/product/dowa_generate>

### Method exclusive control (SELECT ～ FOR UPDATE)

Trong ServiceGeneraterd dùng method bên dưới để exclusive control. Dùng khi cần thiết.

・lockByIdUpdateDatetime(Không có giá trị trả về)

・findAndLockByIdUpdateDatetime(Có giá trị trả về)

Method này sẽ set key chủ của các table và ngày giờ update làm điều kiện, rồi tiến hành loại trừ dòng. Trường hợp không tồn tại record (đã update bởi người khác), trường hợp record đã được loại trừ, (error SQL: ORA-61000 sẽ trả về, 他者 (người khác) thì đang Lock) thì sẽ throw exception với tư cách là application exception.

Set exception này như là cơ cấu xuất ra cảnh báo trên màn hình bằng cách trả về y nguyên trên màn hình.

## Cái khác

### Liên quan tới kiểu data của Field Form, Dto, Model

Liên quan đến kiểu data của Field bean mà dùng để di chuyển data thì cơ bản là sẽ implement theo cách nghĩ bên dưới.

#### Form

Do Form sẽ duy trì giá trị đã được truyền từ màn hình nên sẽ định nghĩa toàn bộ Field bằng kiểu String

#### Model

Do Model sẽ duy trì giá trị truyền vào JSP nên sẽ định nghĩa toàn bộ Field bằng kiểu String.

Tuy nhiên, trường hợp bên dưới không nên dùng kiểu String.

* Thông tin dùng để control màn hình (dùng trong cụm điều kiện của if trên JSP) thì nên dùng kiểu Java primitive mà không phải là kiểu String.
* Thông tin mà sẽ Input vào tag library thì không nên dùng kiểu String.

#### Dto, ResultDto

Do Dto, ResultDto sẽ input/output Logic nên phải dùng kiểu data có nghiệp vụ. Thông tin của giá trị số thì dùng kiểu BigDecimal (chẳng hạn).

Tuy nhiên, do date input của màn hình biểu hiện bằng Char(8) trên database nên sẽ duy trì theo dạng yyyyMMdd ở Dto layer.

Convert format ở Action layer, rồi di chuyển vào Dto.

### Liên quan đến việc di chuyển data

Việc di chuyển data về cơ bản là dùng class org.seasar.framework.beans.util.Beans để tiến hành. Do Converter của date, giá trị số đang được implement nên sẽ dùng khi cần thiết. Ngoài ra, do đã chuẩn bị sẵn Converter để convert kiểu date input màn hình (yyyy/MM/dd) thành kiểu date định nghĩa database (yyyyMMdd) nên sẽ sử dụng.

Tham khảo (Seaser Beans)

<http://s2container.seasar.org/2.4/ja/beans.html>

Bảng 4‑6　 Converter đã implement ở common

|  |  |
| --- | --- |
| Converter | Giải thích |
| dateInputFormatConverter | Convert yyyyMMdd thành yyyy/MM/dd |
| dateSystemFormatConverter | Convert yyyy/MM/dd thành yyyyMMdd |

　Image dùng Converter mà đã implement ở common

// DtoをModelへ移送

Beans.copy(findResultDto, this.sd04211Model).dateConverter("yyyy/MM/dd hh:mm:ss", "updata")

.dateInputFormatConverter("yksdt", "ykedt").execute();

// Formから、Dtoへ移送

final Sd04211RegisterDto registerDto =

Beans.createAndCopy(Sd04211RegisterDto.class, this.editForm)

.dateSystemFormatConverter("yksdt", "ykedt").execute();

### Liên quan đến paging list

Paging list sẽ thực hiện theo cấu trúc bên dưới



Hình 4‑4　Cơ cấu paging list (Màn hình→DB)

Ở màn hình (JSP、CSJS), dùng phần taglib, csjs của common để sắp xếp button 「次へ」「前へ」. Ngoài ra, thiết lập giá trị vào hidden dùng để gửi thông tin paging.

Ở Action, nhận thông tin hidden dùng để gửi thông tin paging rồi di chuyển đến Service. Ở Service, hàm số common dùng để paging đang được thiết lập vào AbstractServiceGenerated, tiến hành get data bằng cách gọi hàm số common.

Ở class Action cần thiết lập số record hiển thị trong page.



Hình 4‑5　Cơ cấu paging của list (DB→màn hình)

Tuy data đã get từ DB sẽ di chuyển và trả về màn hình nhưng thông tin kết quả control paging thì sẽ trả về màn hình bằng cách thiết lập với tư cách là thuộc tính trong request. (Có thể trả về với tư cách parameter. Màn hình có nhiều list thì sẽ trả về bằng parameter)

Thông tin kết quả control page đã được trả về ở màn hình sẽ được setting vào trong common taglib hoặc là common CSJS, nhờ vậy mà có thể sử dụng lại.

Bên dưới là giải thích liên quan đến phần common.

#### Tablib　dowaui:pagingControl

Bố trí button 「次へ」(Next) 「前へ」(Previous). Set pagingParameterId rồi cho liên kết với dowaui:pagingParameter.

<dowaui:pagingControl onClick="dowa.sd.sd04222.search();" id="pagingCongorlTop" pagingParameterId="pagingParameter" />

Bảng 4‑7　Tham số input của dowaui:pagingControl

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc/Tùy ý |
| id | Thuộc tính id của Html | String | Bắt buộc |
| onClick | Event JavaScript khi nhấn button (「次へ」「前へ」 được setting event tương tự) | String | Bắt buộc |
| parameter | Thông tin kết quả paging  (Nếu chưa được setting thì sẽ dùng giá trị thuộc tính của request) | PagingControlResult | Tùy ý |
| pagingParameterId | id của PagingParameter sẽ liên kết | String | Bắt buộc |

#### Taglib　dowaui:pagingDisplayNumber

Hiển thị số lượng page

<dowaui:pagingDisplayNumber id="pagingNumberTop"/>

Bảng 4‑8　Tham số input của dowaui:pagingDisplay

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc/Tùy ý |
| id | Thuộc tính id của Html | String | Bắt buộc |
| parameter | Thông tin kết quả paging  (Nếu chưa được setting thì sẽ dùng giá trị thuộc tính của request) | PagingControlResult | Tùy ý |

#### Taglib dowaui:pagingParameter

Tạo ra field gửi parameter hidden dùng cho paging.

Parameter sẽ được gửi dưới dạng 「pagingControl.currentPage」, nhưng bằng cách setting prefix mà có thế gắn tiếp đầu ngữ (Sử dụng trong trường hợp cần nhiều paging trong 1 màn hình)

<dowaui:pagingParameter id="pagingParameter" />

Bảng 4‑9 Tham số input của dowaui:pagingParameter

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc/Tùy ý |
| id | Thuộc tính id của Html | String | Bắt buộc |
| prefix | Tiếp đầu ngữ của thuộc tính name của Htm | String | Bắt buộc |
| parameter | Thông tin kết quả paging(Nếu chưa được setting thì dùng giá trị thuộc tính của request) | PagingControlResult | Tùy ý |

#### CSJS(JQuery UI Widget) pagingControl

Implement CSJS(JQuery UI Widget) của dowaui:pagingControl

Constructor

$('#pagingCongorlTop').pagingControl({

pagingParameterId : "pagingParameter",

total : "0",

from : "0",

to : "0",

currentPage : "0",

sortKey : "null",

sortOrder : "null"

});

Bảng 4‑10 Tham số input của CSJS pagingControl

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc/Tùy ý |
| total | Tổng số record | String | Bắt buộc |
| from | Số lượng record của From của pager | String | Bắt buộc |
| to | Số lượng record của To của pager | String | Bắt buộc |
| currentPage | Số lượng page hiện tại | String | Bắt buộc |
| sortKey | Sort key | String | Bắt buộc |
| sortOrder | Loại sort (Asc or Desc) | String | Bắt buộc |

Hàm số setControl

$('#pagingCongorlTop').pagingControl('setControl', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

"to" : resultObject.pagingControl.numTo,

"currentPage" : resultObject.pagingControl.currentPage,

"sortKey" : resultObject.pagingControl.sortKey,

"sortOrder" : resultObject.pagingControl.sortOrder,

});

Tham số input thì giống với constructor.

#### CSJS(JQuery UI Widget)　pagingNumber

Implement CSJS(JQuery UI Widget) của dowaui:pagingNumber

Constructor

$('#pagingNumberTop').pagingNumber({

total : "0",

from : "0",

to : "0"

});

Bảng 4‑11　 Tham số input của CSJS pagingNumber

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | Giải thích | Kiểu data | Bắt buộc/Tùy ý |
| total | Tổng số record | String | Bắt buộc |
| from | Số lượng record của From của pager | String | Bắt buộc |
| to | Số lượng record của To của pager | String | Bắt buộc |

Hàm số setNumber

$('#pagingNumberTop').pagingNumber('setNumber', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

"to" : resultObject.pagingControl.numTo

});

Tham số input thì giống với constructor

#### jp.co.dowa.sd.core.dto.PagingControl

Là class Dto để chuyển giá trị từ màn hình vào DB. Có ở Class Dto nhưng trường hợp ngoại lệ thì dùng ở bất kì Layer nào cũng được.

#### jp.co.dowa.sd.core.dto.PagingControlResult

Là class Dto để chuyển giá trị từ DB vào màn hình. Có ở Class Dto nhưng trường hợp ngoại lệ thì dùng ở bất kì Layer nào cũng được.

#### jp.co.dowa.sd.service.AbstractServiceGenerated#getPagingResultList

Dựa vào thông tin của class PagingControl đã chuyển sẽ sinh ra class PagingControlResult rồi get data từ tổng số record, DB và trả về.

### Liên quan tới việc sử dụng thông tin user login

Thông tin user login sẽ get từ jp.co.dowa.sd.util.ContextUtils rồi sử dụng. Thông tin user login thì đang get ở LoginInterCeptor rồi thiết lập.

Tuy cũng có thể get từ API của intra-mart nhưng sẽ tồn tại data cần load trong trường hợp xử lý chỉ định trước (xử lý batch chẳng hạn). Do sẽ bổ sung nội dung đó trong LoginInterceptor và API, nên nhất định phải get thông qua API.

Có thể get thông tin bên dưới từ ContextsUtils

Bảng 4‑12　Data có thể get từ ContextsUtils

|  |  |  |
| --- | --- | --- |
| Tham số | Giải thích | Kiểu data |
| userId | ログインユーザID | String |
| locale | ロケール | Locale |

Trường hợp có thông tin cần phải add thì vui lòng liên lạc với Komada.

## Security

### Biện pháp injection SQL

Dùng JdbcManager của S2JDBC để tiến hành access data. Ngoài ra, khi dùng JdbcManager, không nên kết hợp giá trị đã được gửi từ màn hình như là chuỗi ký tự, rồi generate như là câu SQL.

OK

this.select().id(spart, mtmcd, mesno).where(

"name = ?",

name); // nameが画面送信値

NG

this.select().id(spart, mtmcd, mesno).where(

"name = " + name); // nameが画面送信値

### Biện pháp Cross-site scripting

Phải tiến hành vô hiệu hóa (sanitizing) ở chỗ sử dụng giá trị gửi từ server khi implement JSP.

VD vô hiệu hóa：Thực hiện vô hiệu hóa ở f:h() kiểu EL

<input type="text" id="matnr" name="matnr" value="${f:h(sd02101Model.matnr)}" maxlength="18"/>

※Tuy có phần chưa được vô hiệu hóa trong nội bộ logic taglib nhưng sẽ sửa sớm

### Biện pháp Cross-Site Request Forgery

Biện pháp Cross-Site Request Forgery chỉ set xử lý ứng với việc update database như 「登録」「更新」 làm đối tượng. Xử lý không ứng với việc update database thì không tiến hành biện pháp này.

Biện pháp Cross-Site Request Forgery sẽ tiến hành theo khái niệm bên dưới.



Hình 4‑6　Khái niệm của biện pháp Cross-Site Request Forgery

Khi thực hiện khái niệm bên trên thì sẽ dùng component bên dưới. Dùng cái đó để implement.

#### Issue Token rồi lưu vào Session

Dùng Intercetor để control Token. Nếu gắn annotation SaveToken vào class Action thì sẽ issue tự động Token rồi lưu vào Session.

/\*\*

\* 車両台帳登録初期表示 Action.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

@SaveToken

public class IndexAction {

#### Thiết lập Token bằng hidden ở phần form của màn hình

Dùng tag dowaui:form. Dùng tag bên trái để generate form. Tuy nhiên, chỉ dùng form dùng để gửi màn hình

Dùng tag dowaui:form

<dowaui:form id="sd04211Form" name="sd04211Form" method="post">

・・・入力項目等

</dowaui:form>

<form id="sd04211EditForm" name="sd04211EditForm" method="post" action="${f:url('/sd04211/edit/')}">

Vì bên phía tôi chỉ dùng để generate URL nên không dùng dowaui:form

</form>

<form id="sd04211ReadForm" name="sd04211ReadForm" method="post" action="${f:url('/sd04211/')}">

</form>

#### So sánh Token mà đã được lưu trong Session và Token mà đã được điều tra từ màn hình, rồi check

Đang dùng Intercetor để control Token. Nếu gắn annotation TokenValid vào class Action thì sẽ check Token một cách tự động. Trường hợp error thì sẽ di chuyển đến màn hình system error.

/\*\*

\* 車両台帳登録 登録 Action.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

@TokenValid

public class EditAction {

# Giải thích về hình ảnh implement

Ở mục này sẽ giải thích chi tiết về các layer đã nêu ở mục trước.

## Hiển thị khởi tạo màn hình

Nêu ra ví dụ về hình ảnh implement cho phần hiển thị khởi tạo của màn hình車両台帳登録

### Hình ảnh di chuyển màn hình

Implement với dự định di chuyển như bên dưới

Nhấn menu

Chọn từ list

Menu

List

車両台帳登録

Hình 5‑1　Di chuyển màn hình khi hiển thị khởi tạo màn hình 車両台帳登録

### Cấu trúc source

Cấu trúc source như bên dưới. Ngoài ra, khi hiển thị khởi tạo thì sẽ di chuyển từ màn hình khác cho nên không dùng lớp client. Và, đối với lớp service thì vì những cái không phải là sevice mới được sinh ra tự động cho nên sẽ không mô tả.

|  |  |  |
| --- | --- | --- |
| Layer | Class chính | Source |
| Lớp presentation | Action | jp.co.dowa.sd.action.sd04211.IndexAction |
| Form | jp.co.dowa.sd.form.sd04211.IndexForm |
| Model | jp.co.dowa.sd.model.sd04211.Sd04211Model |
| JSP | /jp/co/dowa/sd/sd04211/index.jsp |
| Lớp logic | Logic | jp.co.dowa.sd.logic.sd04211.Sd04211Logic |
| Component | jp.co.dowa.sd.component.sd04211.Sd04211DbAccessComponent |
| Dto | jp.co.dowa.sd.dto.sd04211.Sd04211FindDto |
| ResultDto | jp.co.dowa.sd.dto.sd04211.Sd04211FindResultDto |
| Lớp service | Service | jp.co.dowa.sd.service. Zswma0Service |

### Lớp presentation

#### Action

/\*

Phải thêm Copyright

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.action.sd04211;

import jp.co.dowa.sd.core.exception.DowaSdApplicationException;

import jp.co.dowa.sd.core.exception.DowaSdValidateException;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindResultDto;

import jp.co.dowa.sd.enums.ScreenTypeEnum;

import jp.co.dowa.sd.form.sd04211.IndexForm;

Nếu setting đúng chính xác Eclipse thì sẽ tự động được sắp xếp rất đẹp

import jp.co.dowa.sd.logic.sd04211.Sd04211Logic;

import jp.co.dowa.sd.model.sd04211.Sd04211Model;

import org.apache.commons.lang.StringUtils;

import org.seasar.framework.beans.util.Beans;

import org.seasar.struts.annotation.ActionForm;

import org.seasar.struts.annotation.Execute;

import javax.annotation.Resource;

/\*\*

\* 車両台帳登録 初期表示 Action.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

Việc chỉ định JSP thì sẽ mô tả ở sau webapps/WEB-INF/view/

\*/

public class IndexAction {

private static final String INDEX = "/jp/co/dowa/sd/sd04211/index.jsp";

@ActionForm

@Resource

public IndexForm indexForm;

@Resource

public Sd04211Logic sd04211Logic;

public Sd04211Model sd04211Model;

/\*\*

\* indexメソッド.

\*

\* @return 遷移先

Trường hợp hiển thị khởi tạo, khi phát sinh error vì ko có nơi để back về cho nên sẽ di chuyển đến màn hình system error

\* @throws DowaSdValidateException 選択Bắt buộc入力チェックエラー

\* @throws DowaSdApplicationException アプリケーション例外

\*/

@Execute(input = "/jp/co/dowa/sd/core/systemerror.jsp")

public String index() throws DowaSdValidateException, DowaSdApplicationException {

// 選択Bắt buộc入力チェック

if (!StringUtils.equals(this.indexForm.screenType, ScreenTypeEnum.REGIST.getValue())) {

// 製品部門

if (StringUtils.isEmpty(this.indexForm.spart)) {

// TODO:例外メッセージの送信方法

throw new DowaSdValidateException();

Check liên quan đến màn hình, chẳng hạn như bắt buộc chọn thì sẽ check ở Action

}

// 運搬事業者コード

if (StringUtils.isEmpty(this.indexForm.ujscd)) {

// TODO:例外メッセージの送信方法

Cách gửi message trường hợp error thì đang thảo luận.

Dự định là chèn message vào class ngoại lệ, rồi triển khai tự động bằng Interceptor.

throw new DowaSdValidateException();

}

// 車両コード

if (StringUtils.isEmpty(this.indexForm.syacd)) {

// TODO:例外メッセージの送信方法

throw new DowaSdValidateException();

}

// 車両枝番

if (StringUtils.isEmpty(this.indexForm.vehic)) {

// TODO:例外メッセージの送信方法

throw new DowaSdValidateException();

Check common thao tác màn hình (check quyền access) thì sẽ implement như là xử lý common. Bây giờ không implement cũng được

}

}

// 画面操作共通チェック

// TODO:未実装

// 画面返却値の作成(引継ぎ項目コピー)

this.sd04211Model = Beans.createAndCopy(Sd04211Model.class, this.indexForm).execute();

// 更新モードの場合、初期表示情報を取得する

if (!StringUtils.equals(this.indexForm.screenType, ScreenTypeEnum.REGIST.getValue())) {

final Sd04211FindDto findDto =

org.seasar.framework.beans.util.Ở Beans, có thể copy giá trị của cùng tên filed. Hãy tích cực sử dụng.

Ngoài ra, cũng có thể setting loại bỏ, convert giá trị. Tham khảo bên dưới.

<http://s2container.seasar.org/2.4/ja/beans.html>

Beans.createAndCopy(Sd04211FindDto.class, this.indexForm).execute();

final Sd04211FindResultDto findResultDto = this.sd04211Logic.find(findDto);

Beans.copy(findResultDto, this.sd04211Model).dateConverter("yyyy/MM/dd hh:mm:ss", "updata")

.dateInputFormatConverter("yksdt", "ykedt").execute();

}

return INDEX;

}

}

#### Form

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.form.sd04211;

import org.seasar.struts.annotation.Required;

import java.io.Serializable;

public class Sd04211Form implements Serializable {

/\*\*

\* コンストラクタ.

\*/

public IndexForm() {}

/\*\* 製品部門 . \*/

public String spart;

/\*\* 運搬事業者コード. \*/

Có danh sách Validator bên dưới.

http://52.68.179.228/svn/1510\_hanbai\_sanpai/document/40\_製造（M）/01\_調査・検討/開発ガイドライン/Validator一覧.xlsx

Nếu còn thiếu cái nào thì liên hệ Komada

public String ujscd;

/\*\* 車両コード. \*/

public String syacd;

/\*\* 車両枝番号 // TODO:項目名未定. \*/

public String syaSeqno;

/\*\* 画面モード. // TODO:項目名未定. \*/

@Required

public String screenType;

/\*\* 前画面検索条件. TODO:項目名未定. \*/

public String preSearchConditions;

}

#### Model

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.model.sd04211;

import jp.co.dowa.sd.interceptor.anotation.BaseMasterSupplementInfo;

import jp.co.dowa.sd.interceptor.anotation.BaseMasterSupplementName;

import lombok.Data;

import java.io.Serializable;

import java.util.List;

import java.util.Map;

/\*\*

\* 車両台帳登録 Model.

\*

Annotation để get basic master thì dùng như bên dưới.

Gán Data annotation vào Class. Nếu làm như vậy thì có thể tạo được getter、setter ở những chỗ mà ko thể nhìn thấy Lombok. (Có thể nhìn thấy ở outline)

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

@Data

public class Sd04211Model implements Serializable {

/\*\*

\* コンストラクタ.

\*/

public Sd04211Model() {}

/\*\* 運搬事業者プルダウン. \*/

@BaseMasterSupplementInfo(cdtyp = "001", spart = "01", targetkbn = "1", valueField = "ujscd",

nmflg = true)

private List<Map<String, String>> ujsCdSelect;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 運搬事業者名称. \*/

private String name6;

/\*\* 製品部門名称. \*/

private String spart;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号. \*/

private String vehic;

/\*\* 最終更新日. \*/

private String updata;

/\*\* 車両番号コード. \*/

private String sytno;

/\*\* 販売組織コード. \*/

private String nyrvk;

/\*\* 販売組織名称. \*/

private String name1;

/\*\* 車種コード. \*/

private String syscd;

/\*\* 車種名称. \*/

@BaseMasterSupplementName(spart = "ZZ", cdtyp = "12", ms1kb = "5", targetkbn = "1",

valueField = "syscd")

private String mstnm;

/\*\* 最終更新者. \*/

private String hksha;

/\*\* 有効期間\_有効開始日. \*/

private String yksdt;

/\*\* 有効期間\_有効終了日. \*/

private String ykedt;

/\*\* 削除フラグ\_フラグ. \*/

private boolean ykedtOver;

/\*\* 最大積載量\_重量. \*/

private String sdsam;

/\*\* 車両重量\_重量. \*/

private String syaam;

/\*\* 車両総重量\_重量. \*/

private String srsam;

/\*\* 指Hình 番号\_指Hình . \*/

private String aufnr;

/\*\* 画面モード. // TODO:項目名未定. \*/

private String screenType;

/\*\* 前画面検索条件. TODO:項目名未定. \*/

private String preSearchConditions;

}

#### JSP

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

<%@ taglib prefix="imarttag" uri="http://www.intra-mart.co.jp/taglib/foundation/imarttag"%>

<%@ taglib prefix="imart" uri="http://www.intra-mart.co.jp/taglib/core/standard"%>

<%@ taglib prefix="html" uri="http://struts.apache.org/tags-html"%>

Đọc layout.jsp ở Tiles.

Common.css、Jquery sẽ được load tự động

<%@ taglib prefix="f" uri="http://sastruts.seasar.org/functions"%>

<%@ taglib prefix="tiles" uri="http://struts.apache.org/tags-tiles"%>

<%@ taglib prefix="dowaui" uri="http://dowa.co.jp/core/tags" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

Sẽ thành tag title. Cần setting popup (chẳng hạn)

<tiles:insert template="/WEB-INF/view/jp/co/dowa/sd/layout/layout.jsp" flush="true">

<tiles:put name="title" value="${dowaui:message('jp.co.dowa.sd.sd04211.caption.001')}${dowaui:message('jp.co.dowa.sd.common.caption.001')}"/>

<tiles:put name="header" type="string">

Trong Tiles:put name=”header” thì sẽ được setting trong tag header

<script src="${f:url('/dowa/csjs/sd04211/sd04211.js')}"></script>

<script type="text/javascript">

$(function() {

});

function test() {

alert("test");

}

Trong Tiles:put name=”content” thì được setting trong tag body

</script>

</tiles:put>

<tiles:put name="content" type="string">

<div class="ui-form-container">

Việc assign event cho button thì về cơ bản sẽ làm ở file js nhưng vì chỗ này ko thể cho nên cho trực tiếp vào href cũng được

<div class="ui-title">

<imarttag:imartTitleBar title="${dowaui:message('jp.co.dowa.sd.sd04211.caption.001')}" icon="/images/standard/title.gif"/>

<imarttag:imartToolbarFrame>

<imarttag:imartToolbarLeft>

<c:if test="${sd04211Model.screenType=='regist'}">

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.001')}" icon="/images/icons/16x16/fugue-icons/shadow/application-task.png"

href="Javascript:dowa.sd.sd04211.onClickRegisterBtn();" />

<span class="separate\_bar"></span>

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.003')}" icon="/images/icons/16x16/fugue-icons/shadow/application-task.png"

href="javascript:dowa.sd.sd04211.onClickConfirmBtn();" />

</c:if>

<c:if test="${sd04211Model.screenType=='edit'}">

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.002')}" icon="/images/icons/16x16/fugue-icons/shadow/application-task.png"

href="Javascript:dowa.sd.sd04211.onClickUpdateBtn();"/>

<span class="separate\_bar"></span>

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.003')}" icon="/images/icons/16x16/fugue-icons/shadow/application-task.png"

href="javascript:dowa.sd.sd04211.onClickConfirmBtn();" />

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<span class="separate\_bar"></span>

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.004')}" icon="/images/icons/16x16/fugue-icons/shadow/application-task.png"

href="javascript:dowa.sd.sd04211.onClickCorrectionBtn();" />

</c:if>

</imarttag:imartToolbarLeft>

<imarttag:imartToolbarRight>

<imarttag:imartIcon name="${dowaui:message('jp.co.dowa.sd.common.caption.005')}" icon="/images/icons/16x16/agt\_back.png" href="javascript:sd.mockup.sd04211.back()" />

</imarttag:imartToolbarRight>

</imarttag:imartToolbarFrame>

</div>

<form id="sd04211Form" name="sd04211Form" method="post">

<div id=errorDiv>

<span id=msg>

<html:errors/>

CSS thì sẽ vừa xem common CSS reference vừa setting nhen

Taglib thì vừa xem danh sách taglib vừa setting nhen.

</span>

</div>

<input type="hidden" id="screenType" name="screenType" value="${sd04211Model.screenType}" />

<table class="ui-form">

<colgroup>

<col class="wd-12">

<col class="wd-38">

<col class="wd-12">

<col class="wd-38">

</colgroup>

<tr>

<c:if test="${sd04211Model.screenType=='regist'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.002')}</label></th>

<td>

<dowaui:generalPurposeSearchSingle value="${sd04211Model.ujscd}" id="ujscd" name="ujscd" maxlength="10" class="wd-23"

nameValue="${sd04211Model.name6}" namearea\_name="name6" namearea\_class="wd-60" namearea\_maxlength="40"

searchiconOnclick="test();" screenType="${sd04211Model.screenType}" />

</td>

</c:if>

<c:if test="${sd04211Model.screenType!='regist'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.002')}</label></th>

<td>

<dowaui:generalPurposeSearchSingle value="${sd04211Model.ujscd}" id="ujscd" name="ujscd" maxlength="10" class="wd-23"

nameValue="${sd04211Model.name6}" namearea\_name="name6" namearea\_class="wd-60" namearea\_maxlength="40"

searchiconOnclick="test();" screenType="read" />

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='regist'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.003')}</label></th>

</c:if>

<c:if test="${sd04211Model.screenType!='regist'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.003')}</label></th>

</c:if>

<td>

<dowaui:productDeptSelect value="${sd04211Model.spart}" iphkb="" gmkcd="" id="spart" name="spart" width="200px"/>

</td>

</tr>

<tr>

<c:if test="${sd04211Model.screenType=='regist'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.004')}</label></th>

<td>

<input type="text" class="ui-text wd-10" name="syacd" id="syacd" maxlength="4" value="${sd04211Model.syacd}"/>

<input type="text" class="ui-text wd-5" name="vehic" id="vehic" maxlength="1" value="${sd04211Model.vehic}"/>

</td>

</c:if>

<c:if test="${sd04211Model.screenType!='regist'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.004')}</label></th>

<td>

<input type="text" class="ui-text-readonly wd-10" name="syacd" id="syacd" maxlength="4" value="${sd04211Model.syacd}" readonly />

<input type="text" class="ui-text-readonly wd-5" name="vehic" id="vehic" maxlength="1" value="${sd04211Model.vehic}" readonly />

</td>

</c:if>

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.005')}</label></th>

<td>

<input type="text" class="ui-text-readonly wd-27" name="updata" id="updata" maxlength="20" value="${sd04211Model.updata}" readonly />

</td>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.006')}</label></th>

<td>

<input type="text" class="ui-text wd-42" name="sytno" id="sytno" maxlength="20" value="${sd04211Model.sytno}" />

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.006')}</label></th>

<td>

<input type="text" class="ui-text-readonly wd-42" name="sytno" id="sytno" maxlength="20" value="${sd04211Model.sytno}" readonly />

</td>

</c:if>

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.007')}</label></th>

<td>

<dowaui:salesDeptSearchSingle value="${sd04211Model.nyrvk}" id="nyrvk" name="nyrvk" class="wd-10" maxlength="4"

nameValue="${sd04211Model.name1}" namearea\_class="wd-34" namearea\_maxlength="16"

paternKbn="" spart="" vkorgdivision="" screenType="read" />

</td>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.008')}</label></th>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.008')}</label></th>

</c:if>

<td>

<dowaui:baseMasterSearchMulti id="syscd" name="syscd" value="${sd04211Model.syscd}" class="wd-10" maxlength="3"

nameValue="${sd04211Model.mstnm}" namearea\_class="wd-35" namearea\_maxlength="40" namearea\_name="mstnm"

spart="" cdtyp="" screenType="${sd04211Model.screenType}" targetType="1"/>

</td>

<th>${dowaui:message('jp.co.dowa.sd.sd04211.caption.009')}</th>

<td><input type="text" class="ui-text-readonly wd-27" name="hksha" id="hksha" maxlength="12" value="${sd04211Model.hksha}" readonly/></td>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.010')}</label></th>

<td>

<div id="floatingCalendarFrom" style="display:inline">

<input type="text" class="ui-text wd-22" name="yksdt" id="yksdt" maxlength="10" value="${sd04211Model.yksdt}">

<imart:floatingCalendar element="document.sd04211Form.yksdt">

<imarttag:imartIcon icon="/images/icons/16x16/fugue-icons/shadow/calendar-blue.png" />

</imart:floatingCalendar>

<img id="clearicon\_floatingCalendarFrom" class="ml-5" border="0" src="/imart/images/icons/16x16/deletecell.png" align="absmiddle">

<label>～</label>

</div>

<div id="floatingCalendarTo" style="display:inline">

<input type="text" class="ui-text wd-22" name="ykedt" id="ykedt" maxlength="10" value="${sd04211Model.ykedt}">

<imart:floatingCalendar element="document.sd04211Form.ykedt">

<imarttag:imartIcon icon="/images/icons/16x16/fugue-icons/shadow/calendar-blue.png" />

</imart:floatingCalendar>

<img id="clearicon\_floatingCalendarTo" class="ml-5" border="0" src="/imart/images/icons/16x16/deletecell.png" align="absmiddle">

</div>

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.010')}</label></th>

<td>

<div id="floatingCalendarFrom" style="display:inline">

<input type="text" class="ui-text-readonly wd-22" name="yksdt" id="yksdt" maxlength="10" value="${sd04211Model.yksdt}" readonly>

<label>～</label>

</div>

<div id="floatingCalendarTo" style="display:inline">

<input type="text" class="ui-text-readonly wd-22" name="ykedt" maxlength="10" value="${sd04211Model.ykedt}" readonly>

</div>

</td>

</c:if>

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.011')}</label></th>

<td>

<input type="checkbox" name="ykedtOver" id="ykedtOver" disabled

<c:if test="${sd04211Model.ykedtOver}">checked</c:if>

/>

</td>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.012')}</label></th>

<td colspan="3">

<input type="text" class="ui-text wd-6" name="sdsam" id="sdsam" maxlength="5" value="${sd04211Model.sdsam}"><label>kg</label>

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.012')}</label></th>

<td colspan="3">

<input type="text" class="ui-text-readonly wd-6" name="sdsam" id="sdsam" maxlength="5" value="${sd04211Model.sdsam}"><label>kg</label>

</td>

</c:if>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.013')}</label></th>

<td colspan="3">

<input type="text" class="ui-text wd-6" name="syaam" id="syaam" maxlength="5" value="${sd04211Model.syaam}"><label>kg</label>

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.013')}</label></th>

<td colspan="3">

<input type="text" class="ui-text-readonly wd-6" name="syaam" id="syaam" maxlength="5" value="${sd04211Model.syaam}" readonly><label>kg</label>

</td>

</c:if>

</tr>

<tr>

<c:if test="${sd04211Model.screenType!='read'}">

<th><label class="ui-require">${dowaui:message('jp.co.dowa.sd.sd04211.caption.014')}</label></th>

<td colspan="3">

<input type="text" class="ui-text wd-6" name="srsam" id="srsam" maxlength="5" value="${sd04211Model.srsam}"><label>kg</label>

</td>

</c:if>

<c:if test="${sd04211Model.screenType=='read'}">

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.014')}</label></th>

<td colspan="3">

<input type="text" class="ui-text-readonly wd-6" name="srsam" id="srsam" maxlength="5" value="${sd04211Model.srsam}" readonly><label>kg</label>

</td>

</c:if>

</tr>

<tr>

<th><label>${dowaui:message('jp.co.dowa.sd.sd04211.caption.015')}</label></th>

<td colspan="3">

<dowaui:generalPurposeSearchSingle value="${sd04211Model.aufnr}" id="aufnr" name="aufnr" maxlength="12" class="wd-12"

searchiconOnclick="test();" screenType="${sd04211Model.screenType}" displayMode="2"/>

</td>

</tr>

</tbody>

</table>

</form>

<form id="sd04211EditForm" name="sd04211EditForm" method="post" action="${f:url('/sd04211/edit/')}">

</form>

<form id="sd04211ReadForm" name="sd04211ReadForm" method="post" action="${f:url('/sd04211/')}">

</form>

</div>

</tiles:put>

</tiles:insert>

### Lớp logic

#### Logic

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.logic.sd04211;

import jp.co.dowa.sd.component.sd04211.Sd04211DbAccessComponent;

import jp.co.dowa.sd.core.exception.DowaSdSystemException;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindResultDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211RegisterDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211UpdateDto;

import jp.co.dowa.sd.entity.Zswma0;

import jp.co.dowa.sd.service.Zswma0Service;

import jp.co.dowa.sd.util.DateUtils;

import org.seasar.framework.beans.util.Beans;

import javax.annotation.Resource;

/\*\*

\* 車両台帳登録 業務ロジック.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

public class Sd04211Logic {

@Resource

Sd04211DbAccessComponent sd04211DbAccessComponent;

@Resource

public Zswma0Service zswma0Service;

/\*\*

\* 車両台帳を取得する.

\*

\* @param findDto パラメータ

\* @return 処理結果

\*/

public Sd04211FindResultDto find(final Sd04211FindDto findDto) {

final Zswma0 zswma0 = this.zswma0Service.findById(findDto.getSpart(), findDto.getUjscd(),

findDto.getSyacd(), findDto.getVehic());

if (null == zswma0) {

// TODO:エラー処理

throw new DowaSdSystemException();

}

final Sd04211FindResultDto result =

Beans.createAndCopy(Sd04211FindResultDto.class, zswma0).execute();

final int periodCheckResult =

DateUtils.periodCheck(DateUtils.getSystemDate(), result.getYksdt(), result.getYkedt());

if (periodCheckResult != 0) {

result.setYkedtOver(true);

}

return result;

}

#### Component

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.component.sd04211;

import jp.co.dowa.sd.entity.Zswma0;

import jp.co.dowa.sd.util.DateUtils;

/\*\*

\* 車両台帳登録 DBアクセス関連.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

public class Sd04211DbAccessComponent {

/\*\*

\* 表示用更新情報を補完する.

\*

\* @param entity 車両台帳Entity

\*/

public void completeUpdateInfo(final Zswma0 entity) {

// TODO:表示用更新組織、更新者の取得

entity.hjkvk = "dep1";

entity.hksha = "aoyagi";

entity.updata = DateUtils.getSystemDate();

}

}

#### Dto

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.dto.sd04211;

import lombok.Data;

import java.io.Serializable;

Gán Data annotation cho Class. Nếu làm như vậy thì sẽ tạo getter、setter ở những chỗ không thể nhìn thấy Lombok. (Có thể thấy ở outline)

/\*\*

\* 車両台帳登録 find用Dto.

\*

\*/

@Data

public class Sd04211FindDto implements Serializable {

/\*\*

\* コンストラクタ.

\*/

public Sd04211FindDto() {}

/\*\* 製品部門 . \*/

private String spart;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号 // TODO:項目名未定. \*/

private String syaSeqno;

}

#### ResultDto

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.dto.sd04211;

import lombok.Data;

import java.io.Serializable;

import java.sql.Timestamp;

/\*\*

\* 車両台帳登録 find用Dto.

\*

\*/

@Data

public class Sd04211FindResultDto implements Serializable {

/\*\* 製品部門 . \*/

private String spart;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号. \*/

private String vehic;

/\*\* 更新日時 . \*/

private Timestamp updata;

/\*\* 車両登録番号. \*/

private String sytno;

/\*\* 販売組織. \*/

private String nyrvk;

/\*\* 車種コード. \*/

private String syscd;

/\*\* 最終更新者 . \*/

private String hksha;

/\*\* 有効開始日. \*/

private String yksdt;

/\*\* 有効終了日. \*/

private String ykedt;

/\*\* 削除フラグ. TODO:DB項目にない。 \*/

private boolean ykedtOver;

/\*\* 最大積載量. \*/

private String sdsam;

/\*\* 車両重量. \*/

private String syaam;

/\*\* 車両総重量. \*/

private String srsam;

/\*\* 指Hình 番号. \*/

private String aufnr;

}

### Lớp service

#### Service

package jp.co.dowa.sd.service;

import jp.co.dowa.sd.condition.Zswma0.Zswma0R01Condition;

import jp.co.dowa.sd.entity.Zswma0;

import javax.annotation.Generated;

/\*\*

\* {@link Zswma0}のサービスクラスです。

\*

\*/

@Generated(

Kế thừa ServiceGenerated

value = {"S2JDBC-Gen 2.4.46",

"org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl"},

date = "2015/12/03 19:16:42")

public class Zswma0Service extends Zswma0ServiceGenerated {

private static final String ZSWMA\_R001 = "ZSWMA\_R001.sql";

/\*\*

\* 車両台帳を取得します.

\*

\* @param condition 取得条件

\* @return 車両台帳データ

Cơ bản là dùng file SQL để chạy SQL.

\*/

public Zswma0 r001(final Zswma0R01Condition condition) {

return this.selectBySqlFile(Zswma0.class, Zswma0Service.ZSWMA\_R001, condition)

.getSingleResult();

}

}

## Chức năng đăng kí/update/delete màn hình

Nêu ví dụ về hình ảnh thực hiện phần đăng kí của màn hình 車両台帳登録.

### Hình ảnh di chuyển màn hình

Implement với dự định di chuyển như bên dưới.

Đăng kí, update, xác định…

Menu

List

車両台帳登録

Hình 5‑2　Di chuyển màn hình khi đăng kí ở màn hình 車両台帳登録

### Cấu trúc source

Cấu trúc source như bên dưới. Ngoài ra, khi hiển thị khởi tạo thì sẽ di chuyển từ màn hình khác cho nên không dùng lớp client. Và, đối với lớp service thì vì những cái không phải là sevice mới được sinh ra tự động cho nên sẽ không mô tả.

Bảng 5‑1 Cấu trúc source khi hiển thị khởi tạo màn hình 車両台帳登録

|  |  |  |
| --- | --- | --- |
| Layer | Class chính | Source |
| Lớp client | CSJS | /dowa/csjs/sd04211/sd04211.js |
| Lớp presentation | Action | jp.co.dowa.sd.action.sd04211.EditAction |
| Form | jp.co.dowa.sd.form.sd04211.IndexForm |
| Form | jp.co.dowa.sd.form.sd04211.EditForm |
| JSP | /jp/co/dowa/sd/sd04211/index.jsp |
| Lớp logic | Logic | jp.co.dowa.sd.logic.sd04211.Sd04211Logic |
| Dto | jp.co.dowa.sd.dto.sd04211.Sd04211RegisterDto |
| Lớp service | Service | jp.co.dowa.sd.service. Zswma0Service |

### Lớp client

#### CSJS

var dowa = dowa || {};

Khai báo NameSpace.

dowa.sd = dowa.sd || {};

dowa.sd.sd04211 = {};

Vì ko thể hoàn thiện ở dạng EL cho nên hãy viết URL vào phía JSP

(function($) {

dowa.sd.sd04211.onClickEditBtn = function() {

$('#sd04211Form').attr('action', $('#sd04211EditForm').attr('action'));

Liên quan đến phần common thì trong nội bộ sẽ serialize Form rồi gửi đến server

dowa.sd.core.request.ajaxSubmit($('#sd04211Form'), $('#sd04211ReadForm').attr('action'));

};

dowa.sd.sd04211.onClickComfirmBtn = function() {

dowa.sd.core.request.ajaxSubmit($('#sd04211Form'));

};

})(jQuery);

### Lớp presentation

#### Action

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.action.sd04211;

import jp.co.dowa.sd.core.exception.DowaSdValidateException;

import jp.co.dowa.sd.dto.sd04211.Sd04211RegisterDto;

import jp.co.dowa.sd.enums.ScreenTypeEnum;

import jp.co.dowa.sd.form.sd04211.EditForm;

import jp.co.dowa.sd.form.sd04211.IndexForm;

import jp.co.dowa.sd.logic.sd04211.Sd04211Logic;

import net.arnx.jsonic.JSON;

import org.apache.commons.lang.StringUtils;

import org.seasar.framework.beans.util.Beans;

import org.seasar.struts.annotation.ActionForm;

import org.seasar.struts.annotation.Execute;

import org.seasar.struts.util.ResponseUtil;

import java.util.HashMap;

import java.util.Map;

import javax.annotation.Resource;

/\*\*

\* 車両台帳登録 登録 Action.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

public class EditAction {

@ActionForm

@Resource

public EditForm editForm;

@Resource

public Sd04211Logic sd04211Logic;

/\*\*

Nếu có lỗi input thì sẽ cho di chuyển đến URL bên trái. JSP này sẽ format kết quả thành dạng JSON rồi trả về màn hình.

\* indexメソッド.

\*

\* @return 遷移先

\* @throws DowaSdValidateException 選択Bắt buộc入力チェックエラー

\*/

@Execute(input = "/jp/co/dowa/sd/core/jsonerror.jsp")

public String index() throws DowaSdValidateException {

// 共通権限チェック

// TODO:未実装

// RQ-3 4の業務処理を行う

// TODO:未実装

if (StringUtils.equals(this.editForm.screenType, ScreenTypeEnum.REGIST.getValue())) {

final Sd04211RegisterDto registerDto =

Di chuyển data của Form vào Dto, rồi gọi Logic.

Beans.createAndCopy(Sd04211RegisterDto.class, this.editForm).execute();

this.sd04211Logic.register(registerDto);

} else if (StringUtils.equals(this.editForm.screenType, ScreenTypeEnum.EDIT.getValue())) {

Lưu data input màn hình dùng cho màn hình tiếp theo (Mở hiển thị khởi tạo của cùng màn hình ở mode tham khảo)

System.out.println("更新モード");

}

// 画面返却値の作成(引継ぎ項目コピー)

final IndexForm indexForm = Beans.createAndCopy(IndexForm.class, this.editForm).execute();

indexForm.screenType = ScreenTypeEnum.READ.getValue();

final Map<String, Object> responseData = new HashMap<String, Object>();

responseData.put("error", false);

responseData.put("data", indexForm);

Chèn data vào Response rồi trả về. Chỗ này định sẽ làm common.

final String jsonText = JSON.encode(responseData);

ResponseUtil.write(jsonText, "application/json");

return null;

}

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.action.sd04211;

import jp.co.dowa.sd.core.exception.DowaSdValidateException;

import jp.co.dowa.sd.dto.sd04211.Sd04211RegisterDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211UpdateDto;

import jp.co.dowa.sd.enums.ScreenTypeEnum;

import jp.co.dowa.sd.form.sd04211.EditForm;

import jp.co.dowa.sd.form.sd04211.IndexForm;

import jp.co.dowa.sd.logic.sd04211.Sd04211Logic;

import jp.co.dowa.sd.util.ResponseUtils;

import org.apache.commons.lang.StringUtils;

import org.seasar.framework.beans.util.Beans;

import org.seasar.struts.annotation.ActionForm;

import org.seasar.struts.annotation.Execute;

import javax.annotation.Resource;

/\*\*

\* 車両台帳登録 登録 Action.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

public class EditAction {

@ActionForm

@Resource

public EditForm editForm;

@Resource

public Sd04211Logic sd04211Logic;

/\*\*

Nếu có lỗi input thì sẽ cho di chuyển đến URL bên trái. JSP này sẽ format kết quả thành dạng JSON rồi trả về màn hình.

\* indexメソッド.

\*

\* @return 遷移先

\* @throws DowaSdValidateException 選択Bắt buộc入力チェックエラー

\*/

@Execute(input = "/jp/co/dowa/sd/core/jsonerror.jsp")

public String index() throws DowaSdValidateException {

// 共通権限チェック

// TODO:未実装

// RQ-3 4の業務処理を行う

// TODO:未実装

if (StringUtils.equals(this.editForm.screenType, ScreenTypeEnum.REGIST.getValue())) {

Di chuyển data của Form vào Dto, rồi gọi Logic.

final Sd04211RegisterDto registerDto =

Beans.createAndCopy(Sd04211RegisterDto.class, this.editForm)

.dateSystemFormatConverter("yksdt", "ykedt").execute();

this.sd04211Logic.register(registerDto);

} else if (StringUtils.equals(this.editForm.screenType, ScreenTypeEnum.EDIT.getValue())) {

final Sd04211UpdateDto updateDto = Beans.createAndCopy(Sd04211UpdateDto.class, this.editForm)

.dateSystemFormatConverter("yksdt", "ykedt").execute();

Lưu data input màn hình dùng cho màn hình tiếp theo (Mở hiển thị khởi tạo của cùng màn hình ở mode tham khảo )

this.sd04211Logic.update(updateDto);

}

// 画面返却値の作成(引継ぎ項目コピー)

final IndexForm indexForm = Beans.createAndCopy(IndexForm.class, this.editForm).execute();

indexForm.screenType = ScreenTypeEnum.READ.getValue();

ResponseUtils.write(false, indexForm);

Chèn data vào Response rồi trả về. Nếu là JSON thì return là null cho nên cần chú ý

return null;

}

}

}

#### Form

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

Có danh sách Validator bên dưới

http://52.68.179.228/svn/1510\_hanbai\_sanpai/document/40\_製造（M）/01\_調査・検討/開発ガイドライン/Validator一覧.xlsx

Nếu còn thiếu cái nào thì liên hệ Komada

\*/

package jp.co.dowa.sd.form.sd04211;

import org.seasar.struts.annotation.Required;

import java.io.Serializable;

/\*\*

\* 車両台帳登録 登録Form.

\*

\* @version $Revision$

\* @author t.komada

\* @since 1.0

\*/

public class EditForm implements Serializable {

/\*\*

\* コンストラクタ.

\*/

public EditForm() {}

/\*\* 製品部門 . \*/

@Required

public String spart;

/\*\* 運搬事業者コード. \*/

@Required

public String ujscd;

/\*\* 車両コード. \*/

@Required

public String syacd;

/\*\* 車両枝番号 // TODO:項目名未定. \*/

public String syaSeqno;

/\*\* 更新日時 // TODO:項目名未定. \*/

public String upddate;

/\*\* 車両番号. \*/

@Required

public String sytno;

/\*\* 販売組織. \*/

public String nyrvk;

/\*\* 車種. \*/

@Required

public String syscd;

/\*\* 最終更新者 //TODO:項目名未定. \*/

public String userName;

/\*\* 有効開始日. \*/

@Required

public String yksdt;

/\*\* 有効終了日. \*/

@Required

public String ykedt;

/\*\* 削除フラグ. \*/

public String ykedtOver;

/\*\* 最大積載量. \*/

@Required

public String sdsam;

/\*\* 車両重量. \*/

@Required

public String syaam;

/\*\* 車両総重量. \*/

@Required

public String srsam;

/\*\* 指Hình 番号. \*/

@Required

public String aufnr;

/\*\* 画面モード. // TODO:項目名未定. \*/

@Required

public String screenType;

/\*\* 前画面検索条件. TODO:項目名未定. \*/

public String preSearchConditions;

}

#### JSP

Dùng những cái giống với hiển thị khởi tạo. Tham khảo 5.1.3.4)JSP

### Lớp logic

#### Logic

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.logic.sd04211;

import jp.co.dowa.sd.component.sd04211.Sd04211DbAccessComponent;

import jp.co.dowa.sd.core.exception.DowaSdSystemException;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211FindResultDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211RegisterDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211UpdateDto;

import jp.co.dowa.sd.entity.Zswma0;

import jp.co.dowa.sd.service.Zswma0Service;

import jp.co.dowa.sd.util.DateUtils;

import org.seasar.framework.beans.util.Beans;

import javax.annotation.Resource;

/\*\*

\* 車両台帳登録 業務ロジック.

\*

\* @version $Revision$

\* @author ndha

\* @since 1.0

\*/

public class Sd04211Logic {

@Resource

Sd04211DbAccessComponent sd04211DbAccessComponent;

@Resource

public Zswma0Service zswma0Service;

/\*\*

\* 車両台帳を登録する.

\*

\* @param registerDto パラメータ

Di chuyển data từ DTO vào Entity rồi gọi Service.

\*/

public void register(final Sd04211RegisterDto registerDto) {

final Zswma0 entity = Beans.createAndCopy(Zswma0.class, registerDto).execute();

this.sd04211DbAccessComponent.completeUpdateInfo(entity);

final int result = this.zswma0Service.insert(entity);

if (result != 1) {

throw new DowaSdSystemException();

}

return;

}

/\*\*

\* 車両台帳を更新する.

\*

\* @param updaterDto パラメータ

\*/

public void update(final Sd04211UpdateDto updaterDto) {

final Zswma0 entity = Beans.createAndCopy(Zswma0.class, updaterDto).execute();

this.sd04211DbAccessComponent.completeUpdateInfo(entity);

final int result = this.zswma0Service.updateExcludesNull(entity);

if (result != 1) {

throw new DowaSdSystemException();

}

return;

}

}

#### Dto

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.dto.sd04211;

import lombok.Data;

import java.io.Serializable;

/\*\*

\* 車両台帳登録 登録Form.

\*

Gán Data annotation cho Class. Nếu làm như vậy thì sẽ tạo getter、setter ở những chỗ không thể nhìn thấy Lombok. (Có thể thấy ở outline)

\* @version $Revision$

\* @author t.komada

\* @since 1.0

\*/

@Data

public class Sd04211RegisterDto implements Serializable {

/\*\* 製品部門 . \*/

private String spart;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号 // TODO:項目名未定. \*/

private String syaSeqno;

/\*\* 車両番号. \*/

private String sytno;

/\*\* 販売組織. \*/

private String nyrvk;

/\*\* 車種. \*/

private String syscd;

/\*\* 有効開始日. \*/

private String yksdt;

/\*\* 有効終了日. \*/

private String ykedt;

/\*\* 最大積載量. \*/

private String sdsam;

/\*\* 車両重量. \*/

private String syaam;

/\*\* 車両総重量. \*/

private String srsam;

/\*\* 指Hình 番号. \*/

private String aufnr;

}

### Lớp service

Vì sẽ dùng implement AbstractServiceGenerated mà đang kế thừa Zswma0Service cho nên không mô tả.

## Chức năng hiển thị danh sách của màn hình report đơn

Nêu lên ví dụ về implement của phần 「処理品・その他・明細」 của màn hình受注伝票入力.

### Hình ảnh implement

Trường hợp là report đơn thì không phải implement phần danh sách bằng Ajax access mà sẽ bố trí chuỗi JSON vào màn hình rồi bind giá trị đó vào template của JsRender.



Hình 5‑3 Hình ảnh implement phần danh sách của màn hình report đơn

### Lớp presentation(Get giá trị, triển khai template)

#### Action

Extract IndexAction#index

Beans.copy(findResultDto, this.sd04211Model).dateConverter("yyyy/MM/dd hh:mm:ss", "updata")

.dateInputFormatConverter("yksdt", "ykedt").execute();

final List<OrderModel> orderModelList = new ArrayList<OrderModel>();

for (int i = 0; i < 3; i++) {

final OrderModel orderModel = new OrderModel();

orderModel.setSpart("ZZ");

orderModel.setNsgam(i + "-1");

orderModel.setSekam(i + "-2");

orderModel.setMenge(i + "-3");

orderModel.setMeins(i + "-4");

orderModel.setJpckb(i + "-5");

orderModel.setPayme(i + "-6");

Dùng JSONIC để chuyển object thành JSON. Setting chuỗi JSON vào Model

orderModel.setBill(i + "-7");

orderModel.setMnfno(i + "-8");

orderModel.setAufnr(i + "-9");

orderModelList.add(orderModel);

}

final String disposalPermitJsonString =

JSON.escapeScript(orderModelList.toArray(new OrderModel[orderModelList.size()]));

this.sd04211Model.setDisposalPermitJsonString(disposalPermitJsonString);

#### JSP

Extract index.jsp

Phần template

<script id="disposalPermitTemplate" type="text/x-jsrender">

<tr>

<td rowspan="2">

Setting Type=”text/x-jsrender”. Phải chỉ rõ là nó là Temlpate

<input type="checkbox" />

</td>

<td>

<input type="checkbox" />

</td>

<td>

<dowaui:generalPurposeSearchSingle id="orderForm[{{>idx}}].spart" name="orderForm[{{>idx}}].spart" value="{{>spart}}" maxlength="10" class="wd-23"

nameValue="" namearea\_class="wd-60" namearea\_maxlength="40"

searchiconOnclick="test();" screenType="" inJsRender="true" />

</td>

<td colspan="2">

Ngay cả tham số của Taglib, cũng có thể thay thế JsRender. Trong Taglib, hãy setting inJsRender=”true”. (Đang có đối ứng trùng lặp tag Script)

<dowaui:itemSearchSingle value="" id="" name="" maxlength="10" class="wd-23"

nameValue="" namearea\_name="" namearea\_class="wd-60" namearea\_maxlength="40"

searchiconOnclick="test();" screenType="" inJsRender="true" />

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].nsgam" name="orderForm[{{>idx}}].nsgam" value="{{>nsgam}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].nsgam" name="orderForm[{{>idx}}].nsgam" value="{{>nsgam}}" class="ui-text-readonly wd-60" readonly />

Có thể dùng câu if của JsRender. Hãy dùng khi cần thiết

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].sekam" name="orderForm[{{>idx}}].sekam" value="{{>sekam}}" class="ui-text wd-60"/>

{{else}}

<input type="text" id="orderForm[{{>idx}}].sekam" name="orderForm[{{>idx}}].sekam" value="{{>sekam}}" class="ui-text-readonly wd-60" readonly />

Ở member của field name [配列添え字].field class trong actionForm thì có thể truyền kiểu mảng vào ActionForm

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].menge" name="orderForm[{{>idx}}].menge" value="{{>menge}}" class="ui-text wd-60" />/<input type="text" id="orderForm[{{>idx}}].meins" name="orderForm[{{>idx}}].meins" value="{{>meins}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].menge" name="orderForm[{{>idx}}].menge" value="{{>menge}}" class="ui-text-readonly wd-60" readonly />/<input type="text" id="orderForm[{{>idx}}].meins" name="orderForm[{{>idx}}].meins" value="{{>meins}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td rowspan="2">

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].jpckb" name="orderForm[{{>idx}}].jpckb" value="{{>jpckb}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].jpckb" name="orderForm[{{>idx}}].jpckb" value="{{>jpckb}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].payme" name="orderForm[{{>idx}}].payme" value="{{>payme}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].payme" name="orderForm[{{>idx}}].payme" value="{{>payme}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].bill" name="orderForm[{{>idx}}].bill" value="{{>bill}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].bill" name="orderForm[{{>idx}}].bill" value="{{>bill}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].mnfno" name="orderForm[{{>idx}}].mnfno" value="{{>mnfno}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].mnfno" name="orderForm[{{>idx}}].mnfno" value="{{>mnfno}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td>

{{if screenType == 1}}

<input type="text" id="orderForm[{{>idx}}].aufnr" name="orderForm[{{>idx}}].aufnr" value="{{>aufnr}}" class="ui-text wd-60" />

{{else}}

<input type="text" id="orderForm[{{>idx}}].aufnr" name="orderForm[{{>idx}}].aufnr" value="{{>aufnr}}" class="ui-text-readonly wd-60" readonly />

{{/if}}

</td>

<td rowspan="2">

<input type="text" id="" name="" />

</td>

</tr>

</script>

Extract index.jsp

Phần lưu JSON

<input type="hidden" id="disposalPermitJsonString" value="${f:h(sd04211Model.disposalPermitJsonString)}" />

### Lớp client

#### CSJS

Parse chuỗi JSON

$(document).ready(function() {

var disposalPermitData = ImJson.parseJSON($('#disposalPermitJsonString').val());

for (var i = 0; i < disposalPermitData.length; i++ ) {

disposalPermitData[i].idx = i;

disposalPermitData[i].screenType = ($('#screenType').val() == "read") ? 0 : 1;

var template = $.templates('#disposalPermitTemplate');

Convert screenType thành 0 or 1. Vì lí do gì đó không thể so sánh string ở phía JsRender Template.

var htmlOutput = template.render(disposalPermitData[i]);

$('#disposalPermitRowTbody').append(htmlOutput);

}

});

Get Template rồi bind giá trị.

Append html đã sinh ra

### Lớp presentation(Get data update danh sách)

#### Form

Extract EditForm

public OrderForm[] orderForm;

## Chức năng search của màn hình danh sách

Nêu ví dụ về hình ảnh implement phần search của màn hình 車両台帳.

### Hình ảnh di chuyển màn hình

Implement với dự định di chuyển như bên dưới.

Nhấn menu

Nhấn menu

Chọn từ danh sách

Menu

Danh sách

車両台帳登録

Hình 5‑4　Di chuyển màn hình 車両台帳

### Cấu trúc source

Cấu trúc source như bên dưới。

Bảng 5‑2　Cấu trúc source khi search màn hình danh sách

|  |  |  |
| --- | --- | --- |
| Layer | Class chính | Source |
| Lớp presentation | Action | jp.co.dowa.sd.action.sd04222.SearchAction |
| Form | jp.co.dowa.sd.form.sd04222.SearchForm |
| Model | jp.co.dowa.sd.model.sd04222.SearchResultModel |
| JSP | /jp/co/dowa/sd/sd04222/index.jsp |
| Lớp logic | Logic | jp.co.dowa.sd.logic.sd04211.Sd04211Logic |
| Dto | jp.co.dowa.sd.dto.sd04211.Sd04211SearchDto |
| Dto | jp.co.dowa.sd.dto.sd04211.Sd04211SearchResultDto |
| Lớp service | Service | jp.co.dowa.sd.service.Zswma0Service |
| Lớp client | CSJS | /dowa/csjs/sd04222/sd04222.js |

### Lớp client

#### CSJS

var dowa = dowa || {};

Chỉ định hàm số Callback trong tham số thứ 2 của ajaxSen.

Khi tham số thứ 3 error, tham số thứ 4 bình thường thì cả 2 là error, nhưng hình như hầu như là không có dùng.

dowa.sd = dowa.sd || {};

dowa.sd.sd04222 = {};

$(function() {

dowa.sd.sd04222.search = function() {

$('#sd04222Form').attr('action', $('#sd04222SearchForm').attr('action'));

dowa.sd.core.request.ajaxSend($('#sd04222Form')

// done

,function(resultObject, textStatus, jqXHR) {

Tạm thời clear phần danh sách

$('#searchResultListTbody').empty();

for (var i = 0; i < resultObject.data.length; i++ ) {

resultObject.data[i].idx = i;

var template = $.templates('#searchResultTemplate');

var htmlOutput = template.render(resultObject.data[i]);

Triển khai giá trị trả về vào trong Template rồi Aappend.

$('#searchResultListTbody').append(htmlOutput);

$('#syacd\_' + i + '\_link').click({'idx' : i}, function (event) {

var transferInfo = {

'spart' : $('#spart\_' + event.data.idx).val(),

'ujscd' : $('#ujscd\_' + event.data.idx).val(),

'syacd' : $('#syacd\_' + event.data.idx).val(),

Add event của link trong danh sách

'vehic' : $('#vehic\_' + event.data.idx).val(),

'screenType' : 'read',

'preSearchConditions' : ''

}

dowa.sd.core.request.transitionTo($('#sd04222TransferForm').attr('action'), transferInfo);

});

}

$('#pagingNumberTop').pagingNumber('setNumber', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

Setting giá trị vào phần liên quan đến Paging

"to" : resultObject.pagingControl.numTo

});

$('#pagingNumberBottom').pagingNumber('setNumber', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

"to" : resultObject.pagingControl.numTo

});

$('#pagingCongorlTop').pagingControl('setControl', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

"to" : resultObject.pagingControl.numTo,

"currentPage" : resultObject.pagingControl.currentPage,

"sortKey" : resultObject.pagingControl.sortKey,

"sortOrder" : resultObject.pagingControl.sortOrder,

});

$('#pagingCongorlBottom').pagingControl('setControl', {

"total" : resultObject.pagingControl.total,

"from" : resultObject.pagingControl.numFrom,

"to" : resultObject.pagingControl.numTo,

"currentPage" : resultObject.pagingControl.currentPage,

"sortKey" : resultObject.pagingControl.sortKey,

"sortOrder" : resultObject.pagingControl.sortOrder,

});

});

};

$('#searchBtn').click(function() {

dowa.sd.sd04222.search();

});

});

### Lớp presentation

#### Action

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.action.sd04222;

import jp.co.dowa.sd.core.exception.DowaSdValidateException;

import jp.co.dowa.sd.dto.sd04211.Sd04211SearchDto;

import jp.co.dowa.sd.dto.sd04211.Sd04211SearchResultDto;

import jp.co.dowa.sd.form.sd04222.SearchForm;

import jp.co.dowa.sd.logic.sd04211.Sd04211Logic;

import jp.co.dowa.sd.model.sd04222.SearchResultModel;

import jp.co.dowa.sd.util.ResponseUtils;

import org.seasar.framework.beans.util.Beans;

import org.seasar.struts.annotation.ActionForm;

import org.seasar.struts.annotation.Execute;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

/\*\*

\* 車両台帳 検索 Action.

\*

\* @version $Revision$

\* @author t.komada

\* @since 1.0

\*/

public class SearchAction {

@ActionForm

@Resource

public SearchForm searchForm;

@Resource

public Sd04211Logic sd04211Logic;

/\*\*

\* indexメソッド.

\*

\* @return 遷移先

\* @throws DowaSdValidateException 選択Bắt buộc入力チェックエラー

\*/

@Execute(input = "/jp/co/dowa/sd/core/jsonerror.jsp")

public String index() throws DowaSdValidateException {

// 共通権限チェック

// TODO:未実装

final Sd04211SearchDto sd04211SearchDto =

Beans.createAndCopy(Sd04211SearchDto.class, this.searchForm).execute();

sd04211SearchDto.getPagingControl().limit = 10;

Đối với tham số màn hình thì setting limit. Vì chưa quyết định cách lưu giữ Limit cho nên tạm thời chỉ định trực tiếp cũng OK. Nếu có thể thì tôi muốn hoàn thiện (hoàn thành) ở Interceptor

final List<Sd04211SearchResultDto> sd04211SearchResultDtoList =

this.sd04211Logic.search(sd04211SearchDto);

final List<SearchResultModel> searchResultModelList = new ArrayList<SearchResultModel>();

for (final Sd04211SearchResultDto sd04211SearchResultDto : sd04211SearchResultDtoList) {

final SearchResultModel searchResultModel =

Beans.createAndCopy(SearchResultModel.class, sd04211SearchResultDto).execute();

searchResultModelList.add(searchResultModel);

}

// 画面返却値の作成

ResponseUtils.write(false, searchResultModelList);

return null;

}

}

#### Form

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.form.sd04222;

import jp.co.dowa.sd.core.dto.PagingControl;

import java.io.Serializable;

/\*\*

\* 車両台帳 検索Form.

\*

\* @version $Revision$

\* @author t.komada

\* @since 1.0

\*/

public class SearchForm implements Serializable {

/\*\* 製品部門 . \*/

public String spart;

/\*\* 運搬事業者コード. \*/

public String ujscd;

/\*\* 車両コード. \*/

public String syacd;

/\*\* 有効.TODO:項目名未定 \*/

Đối với tham số màn hình thì setting limit. Vì chưa quyết định cách lưu giữ Limit cho nên tạm thời chỉ định trực tiếp cũng OK. Nếu có thể thì tôi muốn hoàn thiện (hoàn thành) ở Interceptor

public String enable;

/\*\* 削除済み.TODO:項目名未定 \*/

public String deleteFlag;

/\*\* ページング制御 情報. \*/

public PagingControl pagingControl;

}

#### Model

package jp.co.dowa.sd.model.sd04222;

import jp.co.dowa.sd.core.dto.PagingControlResult;

import lombok.Data;

/\*\*

\* 車両台帳 検索結果Model.

\*

\* @version $Revision$

\* @author t.komadas

\* @since 1.0

\*/

@Data

public class SearchResultModel {

/\*\* 製品部門. \*/

private String spart;

/\*\* 運搬事業者コード. \*/

public String ujscd;

/\*\* 運搬事業者名称. \*/

private String name6;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号. \*/

private String vehic;

/\*\* 車両番号コード. \*/

private String sytno;

/\*\* 有効期間\_有効開始日. \*/

private String yksdt;

/\*\* 有効期間\_有効終了日. \*/

private String ykedt;

/\*\* 最大積載量\_重量. \*/

private String sdsam;

/\*\* 車両重量\_重量. \*/

private String syaam;

/\*\* 車両総重量\_重量. \*/

private String srsam;

/\*\* 指Hình 番号\_指Hình . \*/

private String aufnr;

/\*\* 削除フラグ\_フラグ. \*/

private boolean ykedtOver;

/\*\* ページング制御パラメータ. \*/

private PagingControlResult pagingControlResult;

}

#### JSP

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

<%@ taglib prefix="imarttag" uri="http://www.intra-mart.co.jp/taglib/foundation/imarttag"%>

<%@ taglib prefix="imart" uri="http://www.intra-mart.co.jp/taglib/core/standard"%>

<%@ taglib prefix="html" uri="http://struts.apache.org/tags-html"%>

<%@ taglib prefix="f" uri="http://sastruts.seasar.org/functions"%>

<%@ taglib prefix="tiles" uri="http://struts.apache.org/tags-tiles"%>

<%@ taglib prefix="dowaui" uri="http://dowa.co.jp/core/tags" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<tiles:insert template="/WEB-INF/view/jp/co/dowa/sd/layout/layout.jsp" flush="true">

<tiles:put name="title" value="収集運搬車両台帳一覧"/>

<tiles:put name="header" type="string">

<script src="${f:url('/dowa/csjs/sd04222/sd04222.js')}"></script>

<script id="searchResultTemplate" type="text/x-jsrender">

<tr>

<td><input type="text" class="ui-text-readonly" value="{{>name6}}" readonly /></td>

<td><a href="javascript:void;" id="syacd\_{{>idx}}\_link">{{>syacd}}</a>

<input type="hidden" id="spart\_{{>idx}}" value="{{>spart}}" />

<input type="hidden" id="ujscd\_{{>idx}}" value="{{>ujscd}}" />

<input type="hidden" id="syacd\_{{>idx}}" value="{{>syacd}}" />

<input type="hidden" id="vehic\_{{>idx}}" value="{{>vehic}}" />

</td>

<td><input type="text" class="ui-text-readonly" value="{{>vehic}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>sytno}}" readonly /></td>

<td></td>

<td><input type="text" class="ui-text-readonly" value="{{>yksdt}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>ykedt}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>sdsam}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>syaam}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>srsam}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>aufnr}}" readonly /></td>

<td><input type="text" class="ui-text-readonly" value="{{>ykedtOver}}" readonly /></td>

</tr>

</script>

</tiles:put>

<tiles:put name="content" type="string">

<div class="ui-form-container">

<div class="ui-title">

<imarttag:imartTitleBar title="収集運搬車両台帳一覧" icon="/images/standard/title.gif"/>

</div>

<form id="sd04222Form" name="sd04222Form" method="post">

<div id=errorDiv>

<span id=msg>

<html:errors/>

</span>

</div>

<table class="ui-form">

<colgroup>

<col class="wd-12">

<col class="wd-38">

<col class="wd-12">

<col class="wd-38">

</colgroup>

<tr>

<th><label class="ui-require">製品部門</label></th>

<td>

<dowaui:productDeptSelect value="${sd04222Model.spart}" iphkb="" gmkcd="" id="spart" name="spart" width="200px"/>

</td>

<th><label>状態</label></th>

<td>

<input type="checkbox" name="enable">有効&nbsp;<input type="checkbox" name="deleteflag" >削除済み

</td>

</tr>

<tr>

<th><label>運搬事業者コード</label></th>

<td>

<dowaui:generalPurposeSearchSingle value="${sd04222Model.ujscd}" id="ujscd" name="ujscd" maxlength="10" class="wd-23"

nameValue="" searchiconOnclick="test();" screenType="regist" displayType="2" />

</td>

<th><label>車両コード</label></th>

<td>

<dowaui:generalPurposeSearchSingle value="${sd04222Model.syacd}" id="syacd" name="syacd" maxlength="10" class="wd-23"

nameValue="" searchiconOnclick="test();" screenType="regist" displayType="2" />

</td>

</tr>

</table>

<div style="text-align:center;"><input type="button" value="検索" id="searchBtn"/><input type="button" value="クリア" class="ml-10" id="clearBtn"/></div>

<dowaui:pagingControl onClick="dowa.sd.sd04222.search();" id="pagingCongorlTop" pagingParameterId="pagingParameter" />

<dowaui:pagingDisplayNumber id="pagingNumberTop"/>

<table class="ui-table mt-10">

Setting part dùng cho paging

<thead>

<tr>

<th><label>運搬事業者名称</label></th>

<th><label>車両コード</label></th>

<th><label>車両枝番</label></th>

<th><label>車両登録番号</label></th>

<th><label>車種名称</label></th>

<th><label>有効開始日</label></th>

<th><label>有効終了日</label></th>

<th><label>最大積載量</label></th>

<th><label>車両重量</label></th>

<th><label>車両総重量</label></th>

<th><label>指Hình 番号</label></th>

<th><label>削除フラグ</label></th>

</tr>

</thead>

Setting part dùng cho paging

<tbody id="searchResultListTbody">

</tbody>

</table>

<dowaui:pagingControl onClick="dowa.sd.sd04222.search()" id="pagingCongorlBottom" pagingParameterId="pagingParameter" />

<dowaui:pagingDisplayNumber id="pagingNumberBottom"/>

<dowaui:pagingParameter id="pagingParameter" />

</form>

<form id="sd04222SearchForm" name="sd04222SearchForm" method="post" action="${f:url('/sd04222/search/')}">

</form>

<form id="sd04222TransferForm" name="sd04222TransferForm" method="post" action="${f:url('/sd04211/')}">

</form>

</div>

</tiles:put>

</tiles:insert>

### Lớp logic

#### Logic

Sd04211Logic#search

/\*\*

\* 車両台帳を検索する.

\*

\* @param searchDto パラメータ

\* @return 検索された車両台帳

\*/

public List<Sd04211SearchResultDto> search(final Sd04211SearchDto searchDto) {

final Zswma0R001Condition condition = new Zswma0R001Condition();

condition.spart = searchDto.getSpart();

condition.ujscd = StringUtils.isEmpty(searchDto.getUjscd()) ? null : searchDto.getUjscd();

condition.syacd = StringUtils.isEmpty(searchDto.getSyacd()) ? null : searchDto.getSyacd();

final List<Zswma0> resultList =

this.zswma0Service.r001(condition, searchDto.getPagingControl());

final List<Sd04211SearchResultDto> resultDtoList = new ArrayList<Sd04211SearchResultDto>();

for (final Zswma0 result : resultList) {

final Sd04211SearchResultDto resultDto =

Beans.createAndCopy(Sd04211SearchResultDto.class, result).execute();

resultDtoList.add(resultDto);

}

return resultDtoList;

}

#### Dto

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.dto.sd04211;

import jp.co.dowa.sd.core.dto.PagingControl;

import lombok.Data;

import java.io.Serializable;

/\*\*

\* 車両台帳 Seach用Dto.

\*

\* @version $Revision$

\* @author t.komada

\* @since 1.0

\*/

@Data

public class Sd04211SearchDto implements Serializable {

/\*\* 製品部門 . \*/

private String spart;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 有効.TODO:項目名未定 \*/

private String enable;

/\*\* 削除済み.TODO:項目名未定 \*/

private String deleteFlag;

/\*\* ページング制御情報. \*/

Chuyển thông tin paging

private PagingControl pagingControl;

}

#### ResultDto

/\*

\* (C) DOWA HOLDINGS Co., Ltd. 2015

\*/

package jp.co.dowa.sd.dto.sd04211;

import jp.co.dowa.sd.core.dto.PagingControlResult;

import lombok.Data;

import java.io.Serializable;

import java.sql.Timestamp;

/\*\*

\* 車両台帳登録 find用Dto.

\*

\*/

@Data

public class Sd04211SearchResultDto implements Serializable {

/\*\* 製品部門 . \*/

private String spart;

/\*\* 運搬事業者コード. \*/

private String ujscd;

/\*\* 車両コード. \*/

private String syacd;

/\*\* 車両枝番号. \*/

private String vehic;

/\*\* 更新日時 . \*/

private Timestamp updata;

/\*\* 車両登録番号. \*/

private String sytno;

/\*\* 販売組織. \*/

private String nyrvk;

/\*\* 車種コード. \*/

private String syscd;

/\*\* 最終更新者 . \*/

private String hksha;

/\*\* 有効開始日. \*/

private String yksdt;

/\*\* 有効終了日. \*/

private String ykedt;

/\*\* 削除フラグ. TODO:DB項目にない。 \*/

private boolean ykedtOver;

/\*\* 最大積載量. \*/

private String sdsam;

/\*\* 車両重量. \*/

private String syaam;

/\*\* 車両総重量. \*/

private String srsam;

/\*\* 指Hình 番号. \*/

private String aufnr;

}

### Lớp service

#### Service

package jp.co.dowa.sd.service;

import jp.co.dowa.sd.condition.zswma0.Zswma0R001Condition;

import jp.co.dowa.sd.core.dto.PagingControl;

import jp.co.dowa.sd.entity.Zswma0;

import java.util.List;

import javax.annotation.Generated;

/\*\*

\* {@link Zswma0}のサービスクラスです。

\*

\*/

@Generated(

value = {"S2JDBC-Gen 2.4.46",

"org.seasar.extension.jdbc.gen.internal.model.ServiceModelFactoryImpl"},

date = "2015/12/03 19:16:42")

public class Zswma0Service extends Zswma0ServiceGenerated {

private static final String R002 = "zswma0\_r002.sql";

/\*\*

\* 車両台帳を取得します.

\*

\* @param condition 取得条件

\* @return 車両台帳データ

\*/

public List<Zswma0> r001(final Zswma0R001Condition condition, final PagingControl pagingControl) {

return this.getPagingResultList(Zswma0.class, Zswma0Service.R002, condition, pagingControl);

}

Hãy gọi getPagingResultList mà đã implement trong AbstractServiceGenerated

}