

使用性向上のための カスタマイズ内容

2022年10月26日～28日

Wagby Developer Days 2022
in 幕張メッセ (Japan IT Week 秋)

 **Arise Innovation, Inc.**

AIとローコード開発を活用し
世の中をより良い世界に

- 会社概要
- Wagbyとは
- カスタマイズ実績
- まとめ

会社概要

AIとローコード開発を活用し 世の中をより良い世界に



会社名 アライズイノベーション株式会社



Arise Innovation, Inc.

設立 2016年7月

所在地 東京都中央区勝どき

資本金 8,000万円

代表者 代表取締役社長 CEO 清水 真

○自社開発 AI OCR



○国産型ロー・ノーコード開発ツール



○国産型 RPAツール

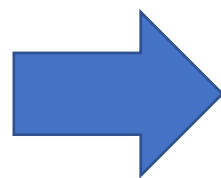


Wagby とは

Wagbyとは

- Webベースの業務アプリケーションを構築できる
国産のローコード・ノーコード開発プラットフォームのことを指します。

画面で設計



アプリケーション完成



Wagbyとは

- Wagbyは生成されたコードを自由にカスタマイズ出来ることが特徴です。



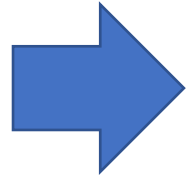
独自行言語なし
知識さえあれば
誰でも
カスタマイズ可能

カスタマイズ実績

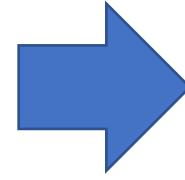
政府機関 勤怠管理システムのWagby導入・カスタマイズ事例



記載、申請



内容確認、不備
確認・承認



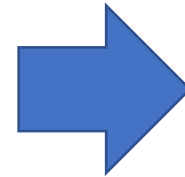
最終結果を人事課が
全て紙で確認



データ入力、エラー
チェック、メール送信



内容確認・承認



人事課はデータで全
データを確認

★カスタマイズポイント（サーバサイドJavaScript）

- ・日付設定ボタンを1回押下することで適用開始日から日数分の明細データが生成されます。

勤務時間割振申請 新規登録 人事次郎

保存 キャンセル **日付設定**

申請フロー 人事課フロー(勤務割振申請)

ヘッダ

提出年月日 2022-10-14

申請者	人事次郎	検索...	課名	<input checked="" type="checkbox"/> 人事課
育短用週休日	<input type="checkbox"/> 月 <input type="checkbox"/> 火 <input type="checkbox"/> 水 <input type="checkbox"/> 木 <input type="checkbox"/> 金		休憩時間短縮	0

勤務区分	フレ2	検索...		
始業時刻			終業時刻	
休憩開始	12:00		休憩終了	13:00

コアタイム開始	10:00	コアタイム終了	15:00
フレックス始業時刻開始	07:00	フレックス始業時刻終了	10:00
フレックス終業時刻開始	16:00	フレックス終業時刻終了	22:00

<明細一括入力欄>

適用開始日	2022-10-15	日数	14
明細用始業時刻	9時0分	明細用終業時刻	17時45分
明細用昼休憩開始	12時	明細用昼休憩終了	13時
明細用夜休憩開始		明細用夜休憩終了	

カスタマイズ実績

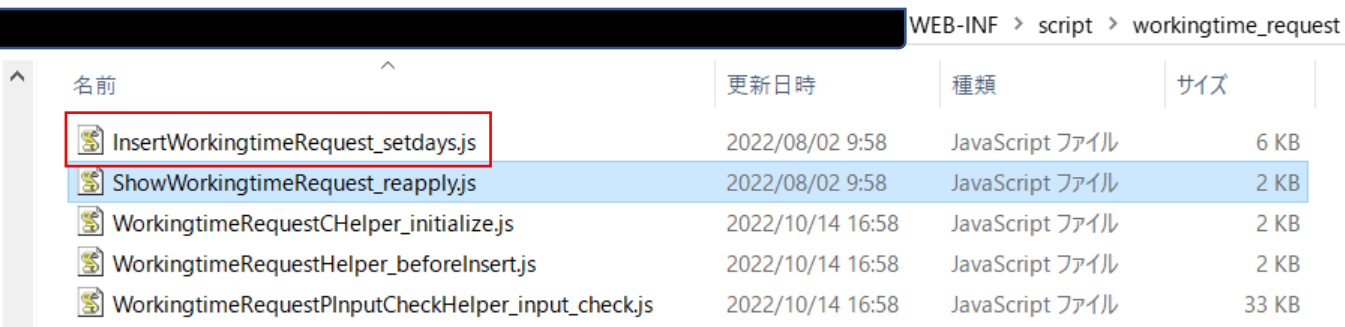
★カスタマイズポイント（サーバサイドJavaScript）

- ・日付設定ボタン押下後
15日～28日（14日分）の明細が生成されました。

総勤務時間数		77時間30分/77時間30分		最短勤務時間		4 時間	
<small>勤務区分がフレ3以外の場合は15分単位で設定してください(フレ3は5分単位で設定が可能です)</small>							
No.	日付	始業時刻	昼休憩開始	夜休憩開始	特例日	休憩	
	勤務時間	終業時刻	昼休憩終了	夜休憩終了	週休日		
1	2022-10-15 (土)	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	0時間0分	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
2	2022-10-16 (日)	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	0時間0分	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
3	2022-10-17 (月)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
4	2022-10-18 (火)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
5	2022-10-19 (水)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
6	2022-10-20 (木)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
7	2022-10-21 (金)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
8	2022-10-22 (土)	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	0時間0分	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
9	2022-10-23 (日)	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	0時間0分	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
10	2022-10-24 (月)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
	7時間45分	17時45分	13時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	
11	2022-10-25 (火)	9時0分	12時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>	

- ・ 業務処理コードとなるためサーバサイドJavaScriptとなります。

サーバサイドJavaScriptはWEB-INF < script < モデル名
にDesignerで設定した独自ボタンイベント名で
処理を記載します。



名前	更新日時	種類	サイズ
InsertWorkingtimeRequest_setdays.js	2022/08/02 9:58	JavaScript ファイル	6 KB
ShowWorkingtimeRequest_reapplyjs	2022/08/02 9:58	JavaScript ファイル	2 KB
WorkingtimeRequestCHelper_initialize.js	2022/10/14 16:58	JavaScript ファイル	2 KB
WorkingtimeRequestHelper_beforeInsert.js	2022/10/14 16:58	JavaScript ファイル	2 KB
WorkingtimeRequestPInputCheckHelper_input_check.js	2022/10/14 16:58	JavaScript ファイル	33 KB

```
InsertWorkingtimeRequest_setdays.js - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

function process() {
    var ExcelFunction = Java.type("jp.jasminesoft.util.ExcelFunction");
    var Jfcerror = Java.type("jp.jasminesoft.jfc.error.Jfcerror");
    var Jfcwarn = Java.type("jp.jasminesoft.jfc.error.Jfcwarn");
    var Jfcinfo = Java.type("jp.jasminesoft.jfc.error.Jfcinfo");
    var HolidayManager = Java.type("jp.jasminesoft.util.HolidayManager");

    //日数が設定されていない場合はエラーとする。
    var days = workingtime_request.checkdays;
    if(days == -1 ){
        var error = new Jfcerror();
        error.content = "明細を追加する為には日数を設定してください。";
        p.errors.addJfcerror(error);
        retuen;
    }
}
//現在の繰り返しコンテナの行を全て削除
var detailLength = workingtime_request.detail.length;
for(var i = 0; i < detailLength; i++){
    workingtime_request_p.removeDetail(0);
}
//取得したストアモデルデータの繰り返しコンテナをremoveする
var WTR = p.request.getAttribute("workingtime_request");
var detailLength_s = WTR.detail.length;
for(var j = 0; j < detailLength_s; j++){
    workingtime_request.removeDetail(0);
}
//繰り返しコンテナを追加する
var PHelperClass = Java.type("jp.jasminesoft.jfc.IPresentationHelper");
var WTRHelper = p.appctx.getBean("WorkingtimeRequestHelper");
var WTRPHelper = p.appctx.getBean("WorkingtimeRequestPHelper");

for (var i = 0; i < days; i++) {
    WTRHelper.addDetail(WTR, -1, true, p, errorManager);
}

//追加した繰り返しコンテナにデータを代入する。
var breaktimeshort = workingtime_request.breaktimeshort;
var checkworkdivision = workingtime_request.checkworkdivision;
var adaptationstartday = workingtime_request.adaptationstartday;
var workstarttime = workingtime_request.workstarttimeDetail;
```

★カスタマイズポイント (JavaScript)

- ・コピーボタンを押下すると押した行のデータを右の時刻入力欄にコピーします

	日付	始業時刻	昼休憩開始	夜休憩開始	特例日	休短	始業時刻	昼休憩開始	夜休憩開始	特例日	休短
	勤務時間	終業時刻	昼休憩終了	夜休憩終了	週休日	.	終業時刻	昼休憩終了	夜休憩終了	週休日	休短
1	2021-03-11 (木)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
2	2021-03-12 (金)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
3	2021-03-13 (土)				○	0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	○	<input type="text"/>
	0時間0分					<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
4	2021-03-14 (日)				○	0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	○	<input type="text"/>
	0時間0分					<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
5	2021-03-15 (月)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
6	2021-03-16 (火)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
7	2021-03-17 (水)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
8	2021-03-18 (木)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>
9	2021-03-19 (金)	09:00	12:00			0	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/> 時	<input type="text"/> 時 <input type="text"/> 分	<input type="text"/>	<input type="text"/>

★カスタマイズポイント (JavaScript)

- ・ 押下したボタンの行がコピーされました。

サーバサイドではないJavaScriptの為、押下した瞬間にコピーされます。

	日付	始業時刻	昼休憩開始	夜休憩開始	特例日	休憩	始業時刻	昼休憩開始	夜休憩開始	特例日	休憩
	勤務時間	終業時刻	昼休憩終了	夜休憩終了	週休日	.	終業時刻	昼休憩終了	夜休憩終了	週休日	
1	2021-03-11 (木)	09:00	12:00			0	9時0分	12時			0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>	17時45分	13時			
2	2021-03-12 (金)	09:00	12:00			0					0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>					
3	2021-03-13 (土)				○	0				○	0
	0時間0分					<input type="button" value="コピー"/>					
4	2021-03-14 (日)				○	0				○	0
	0時間0分					<input type="button" value="コピー"/>					
5	2021-03-15 (月)	09:00	12:00			0					0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>					
6	2021-03-16 (火)	09:00	12:00			0					0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>					
7	2021-03-17 (水)	09:00	12:00			0					0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>					
8	2021-03-18 (木)	09:00	12:00			0					0
	7時間45分	17:45	13:00			<input type="button" value="コピー"/>					
	2021-03-19 (金)	09:00	12:00			0					

カスタマイズ実績

- ・画面のカスタマイズとなるためJavaScriptとなります。

JavaScriptはモデル名とどの画面で
My + 処理が行われているかのファイルに
処理を記載します。

名前	更新日時	種類	サイズ
MyShowWorkingtimeRequest.js	2022/08/02 9:58	JavaScript ファイル	1 KB
MyUpdateWorkingtimeRequest.css	2022/08/02 10:06	カスケード スタイル シ...	1 KB
MyUpdateWorkingtimeRequest.js	2022/08/02 9:58	JavaScript ファイル	7 KB
showListWorkingtimeRequest.css	2022/08/02 10:06	カスケード スタイル シ...	1 KB
showListWorkingtimeRequest.js	2022/08/02 10:06	JavaScript ファイル	6 KB
showListWorkingtimeRequest_body.jsp	2022/08/02 10:06	JSP ファイル	6 KB
showListWorkingtimeRequest_javascriptj...	2022/08/02 10:06	JSP ファイル	1 KB
showListWorkingtimeRequest_navigation...	2022/08/02 10:06	JSP ファイル	2 KB

```
MyUpdateWorkingtimeRequest.js - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
//押下したボタンの明細行を更新の明細行にコピーする
require(["dijit/registry", "dojo/ready"], function(registry, ready) {
    ready(function() {
        for (let i = 1; i < 28; i++) {
            var btn = registry.byId("detail$002fp_button_nop_1_"+i);
            btn.on("click", function () {
                //新規コンテナ内容取得
                var workstarttimenew = document.getElementById('content_
                var workendtimenew = document.getElementById('content_wo
                var breakfromnoontimenew = document.getElementById('cont
                var breaktonoontimenew = document.getElementById('conten
                var breakfromnighttimenew = document.getElementById('con
                var breaktonighttimenew = document.getElementById('conte
                var weeklyholidaynew = document.getElementById('content_
                var specialtargetdatenew = document.getElementById('cont
                var weeklyholidaynew = document.getElementById('content_
                var braketimeshorteningforflexnew = document.getElemenB

                //更新コンテナ要素取得
                var workstarttimeupdatehour = registry.byId("workingtime
                var workstarttimeupdateminute = registry.byId("workingti
                var workendtimeupdatehour = registry.byId("workingtime_r
                var workendtimeupdateminute = registry.byId("workingtime
                var breakfromnoontimeupdatehour = registry.byId("working
                var breaktonoontimeupdatehour = registry.byId("workingti

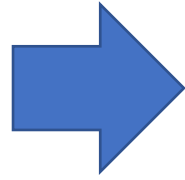
                var breakfromnighttimeupdatehour = registry.byId("workin
                var breakfromnighttimeupdateminute = registry.byId("work
                var breaktonighttimeupdatehour = registry.byId("working
                var breaktonighttimeupdateminute = registry.byId("workin
                var specialtargetdateupdate = registry.byId("workingtime
                var weeklyholidayupdate = registry.byId("workingtime_req
                var braketimeshorteningforflexupdate = registry.byId("wo

                //新規コンテナで取得した時間(09:45等)を先頭2文字と末尾2文字で分ける
                var workstarttimehour = workstarttimenew.slice(0,2);
                var workstarttimeminute = workstarttimenew.slice(-2);
                var workendtimehour = workendtimenew.slice(0,2);
                var workendtimeminute = workendtimenew.slice(-2);
                var breakfromnoontimehour = breakfromnoontimenew.slice(0
                var breaktonoontimehour = breaktonoontimenew.slice(0,2);
```

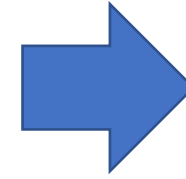

タクシー業 タクシーチケットデータ化業務のWagby導入・カスタマイズ事例



記載



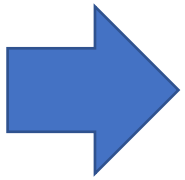
目視でPCへ手入力



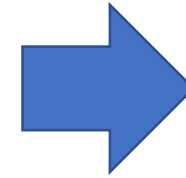
入力ミス・入力漏れ



記載



OCR



結果確認作業

★カスタマイズポイント (JSP・CSS)

- ・画像を見やすいレイアウトにしています。
- ・別の行をクリックすると対象の画像が表示されるようにカスタマイズしています。

No.9

全93件中93件表示 93件中、1件目から93件目を表示しています。

明細情報

料金合計	¥ 2,420
乗車区間	~

削除	No										
<input type="checkbox"/>	7	52015		3606333136	9,500	490	9,990	221013			23033:
<input type="checkbox"/>	8	52015		7139402021	4,120	400	4,520	221013			24033:
<input type="checkbox"/>	9	52015		2629711187	2,420	0	2,420	221013			24033:
<input type="checkbox"/>	10	52011		814280264151	5,620	340	5,960	221013			
<input type="checkbox"/>	11	52011		805063400180	7,840	2,150	9,990	221013			
<input type="checkbox"/>	12	52011		805067855084	9,990	0	9,990	221013			

顧客情報

★カスタマイズポイント (JSP・CSS)

- ・10行名をクリックすることで表示されている画像が変更されました。

全93件中93件表示 93件中、1件目から93件目を表示しています。

No.10

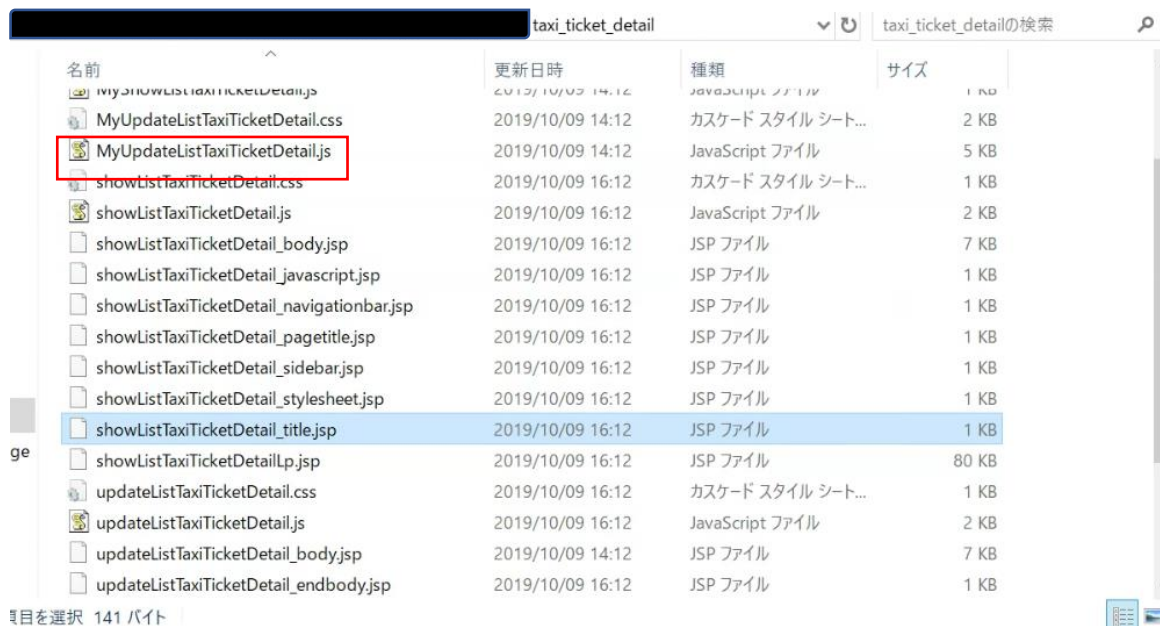
明細情報

単品番号	業務員名	料金計
		5960

削除	No									
<input type="checkbox"/>	7	52015		3606333136	9,500	490	9,990	221013		23033: ^
<input type="checkbox"/>	8	52015		7139402021	4,120	400	4,520	221013		24033:
<input type="checkbox"/>	9	52015		2629711187	2,420	0	2,420	221013		24033:
<input type="checkbox"/>	10	52011		814280264151	5,620	340	5,960	221013		
<input type="checkbox"/>	11	52011		805063400180	7,840	2,150	9,990	221013		
<input type="checkbox"/>	12	52011		805067855084	9,990	0	9,990	221013		

顧客情報

- 画面・見た目のカスタマイズとなる為JSP・CSS・JavaScriptの
カスタマイズとなります。



```
MyUpdateListTaxiTicketDetail.js - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
$(function(){

    var $list = $("#[id^=taxi_ticket_detail_detail_row_]");
    /*
     * 大きい画像を表示する。
     */
    function viewBigImage($image) {
        // 更新一覧
        var $imageDiv = $("#showListtaxi_ticket_detail_ulp");
        // 表示中の画像
        var $bigImage = $("#bigImage");
        // id
        var id = $image.attr("id");
        // index
        var index = id.match(/^taxi_ticket_detail_detail_row_([0-9]*)/);

        // 前回表示した画像が同じものであれば変更しない。
        if ($("#taxi_ticket_detail_body1").data("index") == index) {
            return;
        }
        // 現在行の番号を取得する。
        var rownum = $image.find("span[id$=rownum]").text();

        // 選択行の色を変える。
        $list.eq($("#taxi_ticket_detail_body1").data("index")).attr(
            $image.attr("data-custom-style-selectImg", "true");

        // 前の画像を消します。
        $bigImage.remove();

        // 新しい画像を表示します。
        $("<div>").append($("<div>").attr("id", "imageRownum")).appen

        // 行番号を入れます。
    }
}
```

カスタマイズ実績

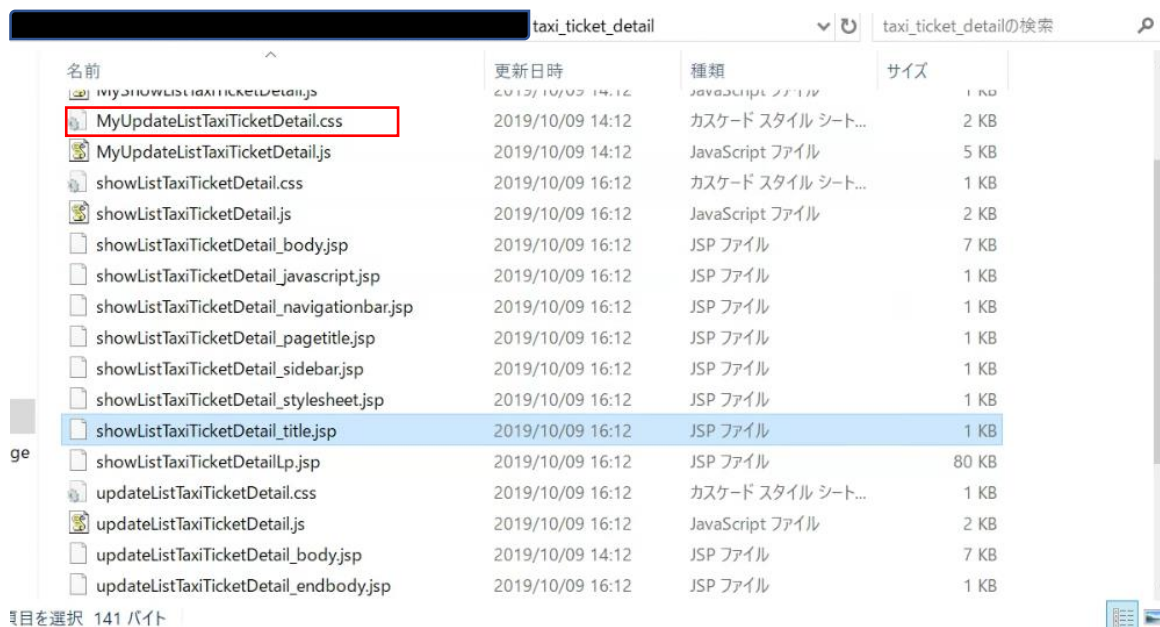
- 画面・見た目のカスタマイズとなる為JSP・CSS・JavaScriptの
カスタマイズとなります。

名前	更新日時	種類	サイズ
myShowListTaxiTicketDetails.js	2019/10/09 14:12	JavaScript ファイル	1 KB
MyUpdateListTaxiTicketDetail.css	2019/10/09 14:12	カスケード スタイル シート...	2 KB
MyUpdateListTaxiTicketDetail.js	2019/10/09 14:12	JavaScript ファイル	5 KB
showListTaxiTicketDetail.css	2019/10/09 16:12	カスケード スタイル シート...	1 KB
showListTaxiTicketDetail.js	2019/10/09 16:12	JavaScript ファイル	2 KB
showListTaxiTicketDetail_body.jsp	2019/10/09 16:12	JSP ファイル	7 KB
showListTaxiTicketDetail_javascript.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetail_navigationbar.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetail_pagetitle.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetail_sidebar.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetail_stylesheet.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetail_title.jsp	2019/10/09 16:12	JSP ファイル	1 KB
showListTaxiTicketDetailLp.jsp	2019/10/09 16:12	JSP ファイル	80 KB
updateListTaxiTicketDetail.css	2019/10/09 16:12	カスケード スタイル シート...	1 KB
updateListTaxiTicketDetail.js	2019/10/09 16:12	JavaScript ファイル	2 KB
updateListTaxiTicketDetail_body.jsp	2019/10/09 14:12	JSP ファイル	7 KB
updateListTaxiTicketDetail_endbody.jsp	2019/10/09 16:12	JSP ファイル	1 KB

```
updateListTaxiTicketDetail_body.jsp - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<%@ page pageEncoding="UTF-8" %>
<fmt:setBundle basename="jfcapp"/>
<table width="100%">
  <tr align="center">
    <td class="customMessage">
      <fmt:message key="my.__jfc_msg.updateListTaxiTicketDetail.htmlcontent">
        <fmt:param value="${sessionScope.session_fileNumber}" />
        <fmt:param value="${__jfc_result.size}" />
      </fmt:message>
    </td>
    <td class="display_resultcount">
      <c:choose>
        <c:when test="${__jfc_result.size > 0}">
          <fmt:message key="__jfc_common.search.status">
            <fmt:param value="${__jfc_result.size}" />
            <fmt:param value="${__jfc_result.current}" />
            <fmt:param value="${__jfc_result.next}" />
          </fmt:message>
        </c:when>
        <c:when test="${__jfc_result.size == 0}">
          <fmt:message key="__jfc_common.search.notfound" />
        </c:when>
        <c:otherwise>
          <fmt:message key="__jfc_common.search.action" />
        </c:otherwise>
      </c:choose>
    </td>
  </tr>
</table>
<form data-dojo-type="dijit/form/Form" id="showListtaxi_ticket_detail_ulp" name="showListtaxi_ticket_detail_ulp"
  action="updateListTaxiTicketDetail.do" method="post" onSubmit="buttonOff();">
  <input type="hidden" name="SENDACTION" />
  <input type="hidden" name="sessionId" value="<jfc:MD5Hash/>" />
  <c:if test="${__jfc_result.size > 0}">
```

カスタマイズ実績

- 画面・見た目のカスタマイズとなる為JSP・CSS・JavaScriptのカスタマイズとなります。



```
MyUpdateListTaxiTicketDetail.css - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#bigImage {
    width: auto;
    height: 280px;
}
#bigImage img {
    width: auto;
    height: 97%;
    max-width: 88%;
    float: none;
}
#bigImage img::after {
    content: '';
    clear: both;
    display: block;
}
tr[id^=taxi_ticket_detail_detail_row_] td:last-child {
    display: none;
}
.display_field_list_selected {
    background-color: rgb(234, 243, 253);
}
```

まとめ

- Wagbyはローコード開発ツールとして十分なWebアプリケーションを開発可能
- プログラミング知識があれば、カスタマイズでさらに複雑な処理を実装可能
- 生産性・柔軟性・拡張性を兼ね備えたプログラマにとって理想的なローコード開発ツール

★アライズイノベーションではカスタマイズトレーニングも用意しております。

カスタマイズ -カスタマイズ方法 サーバーサイドJavaScript-

カスタマイズ -カスタマイズ方法 サーバーサイドJavaScript-

ヘルパの [w07-25]

実行 カスタマイズ -カスタマイズ方法 Java-

ビルドを行うと「My画面識別子モデルIDController.java」が生成されます。
このコードを編集し、下記フォルダに保存すビルドをすることでクラスが有効になります。
赤字で記載されているのは新規で作成する必要のあるフォルダです。
プロジェクト識別子「wagby」モデルID「customer」としてあります。

customize → java → jp → jasminesoft → wagby → controller → customer

名前	更新日時	種類	サイズ
MyShowCustomerController.java	2019/08/14 18:50	JAVA ファイル	1 KB

Copyright © Arise Innovation, Inc. All Rights Reserved.

25

2. タイトル色を変更する(CSS)

CSSカスタマイズモデルの全モデル全てにCSSをカスタマイズ

今回はビルドをせず確認

My.cssに下記を記載します。

```
pageTitle {
border-left: 10px solid #cccccc;
filter: progid:DXImageTransform.Microsoft.Gradient(enabled=false);
background: -webkit-linear-gradient(to top, #cccccc 49%, #cccccc 51%);
background: -moz-linear-gradient(to top, #cccccc 49%, #cccccc 51%);
background: -ms-linear-gradient(to top, #cccccc 49%, #cccccc 51%);
background: -o-linear-gradient(to top, #cccccc 49%, #cccccc 51%);
background: linear-gradient(to top, #cccccc 49%, #cccccc 51%);
}
```

Copyright © Arise Innovation, Inc. All Rights Reserved.

3. ダイアログを表示する(Javascript)

今回はJavaScriptカスタマイズモデル (scriptsample) の登録画面でダイアログを出すための「MyInsertScriptSample.js」を「CustomerHistoryEntityService」に記述する必要があります。

今回はビルドをせず確認

MyInsertScriptSample.jsに下記を記載します。

```
require(["dijit/registry", "dojotoolkit"], function() {
// ボタン処理
var btnSend = registry.byId(btnSendOnClick);
// ボタン押下後処理
showConfirmDialog("01", btnSend);
});
```

ボタン押下後処理ダイアログを表示します

Copyright © Arise Innovation, Inc. All Rights Reserved.

4. 業務ロジック(スクリプト)他モデル登録

顧客情報モデルのヘルパ・登録スクリプトにて下記スクリプトを記載します。

```
//顧客情報
var companyName = customer.companyname;
var customername = customer.customername;

//顧客登録処理
var entityService = p.appctx.getBean("CustomerHistoryEntityService");
var customerHistoryClass = java.type("jp.jasminesoft.wagby.model.customer.history.CustomerHistory");
var customerHistory = new customerHistoryClass();

//処理
customerHistory.companyname=companyName;
customerHistory.customername=customername;
entityService.insert(customerHistory);
```

顧客情報登録ボタン押下時に登録したIDと会社名と顧客名を取得し、変数にそれぞれ代入します。

顧客登録履歴顧客登録履歴のServiceオブジェクト取得します。

Serviceオブジェクトは var entityService = p.appctx.getBean("<モデルID>EntityService"); です。モデルIDはキャメル記法で記載します。

新規登録するための顧客登録履歴の空のオブジェクトを作成します。

ここで「wagby」はプロジェクト識別子「customer_history」はモデルID「CustomerHistory」はキャメル記法でのモデルIDです。

記載方法は Java.type("jp.jasminesoft.プロジェクト識別子.model.モデルID.キャメルモデルID"); となります。

new演算子を使用し、データベースへ登録できる様にします。

//処理顧客登録履歴の会社名と顧客名に顧客情報の会社名と顧客名を代入します。最後にServiceのinsertを実行し、顧客登録履歴に登録します。

Copyright © Arise Innovation, Inc. All Rights Reserved.

27



Arise Innovation, Inc.

アライズイノベーション株式会社

東京都中央区勝どき3-13-1

Mail : sales@ariseinnovation.co.jp

TEL : 03-5534-9967

AIRead :  <https://airead.ai/>

Facebook:  @ariseinnovationinc