

# X-point

## 経路検索連携フォーム 作成ガイド

2026/02/01 版



## はじめに

### ◆本書の目的

本書は、X-point Cloud で利用する交通費精算フォームなどで経路検索機能と連携を行う際の、フォーム作成手順について説明しています。

### ◆対象とする読者

本書は「X-point」のフォーム管理者を対象としています。フォーム管理者とは「eFormMaker」によるフォーム作成を行ない、「X-point」のフォーム管理、マスタ管理権限を持つユーザを指します。

本機能は表明細機能を利用したフォームの作成、及び簡単な JavaScript 作成を行うことができるユーザを前提としています。

### ◆対応バージョン（2026/02/01 時点）

X-point	対応ブラウザ
X-point v3.13  「経路検索オプション」及び「セキュリティ基本サービス」契約をしているユーザー様	Microsoft Edge  Firefox  Google Chrome  Safari (Mac OS)  Safari (iOS 18, 17)  Chrome for Android

※ 「経路検索オプション」の契約が無いお客様は経路検索画面を利用はできません。

※ 本書はPCブラウザでの利用を前提に記述されています。

※ タブレット、スマートフォンでは検索画面の表示方法がPCブラウザと異なりますのでご注意願います。

※ スマートフォンで利用する際は検索画面が画面幅に収まらない場合があります。

※ 収まらない場合は、画面を横方向に向けて利用してください。

### ◆製品名について

本文中、「X-point サーバー」は「X-point」と表記しています。

また、各製品の名称は対応バージョンを省略してある箇所もありますのでご了承ください。

### ◆商標について

本書の一部、または全部を著作権所有者の許諾なしに、商用目的の為に複製、配布することはできません。

X-point、エクスポイントの名称およびロゴは株式会社エイトレッドの商標または登録商標です。

Microsoft、MS-DOS、Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。Macintosh、MacOS は Apple Computer, Inc. の米国およびその他の国における登録商標です。Adobe、Acrobat、Adobe Acrobat は Adobe Systems, Inc. の商標または登録商標です。ORACLE、Java、JavaScript は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

その他、記載された会社名およびロゴ、製品名などは該当する会社の商標または登録商標です。本書では、©、®、(TM) の表示を省略しています。ご了承ください。

### ◆製作著作

©2023 株式会社エイトレッド

## 目次／索引

<b>第1章. 既存フォームへの組み込み</b> .....	<b>4</b>
1.1. 経路検索の利用条件.....	4
1.2. 経路検索組み込みイメージ.....	5
1.3. 最小組み込みコード例.....	10
<b>第2章. そのほか</b> .....	<b>12</b>
2.1. 特定の駅を経由したい場合.....	12
2.2. 経路検索を利用した明細であることを書類上に示す方法.....	13
2.3. 片道、往復の指定を行う方法.....	15
2.4. 経路検索で最も安価な経路を選択した事を書類上に示す方法.....	15
2.5. 経路検索で設定した明細行のフィールドを編集禁止にする方法.....	18
2.6. 経路検索を利用するステップを制限する場合.....	19
2.7. 承認ステップでの編集を可にする場合.....	19

# 第1章. 既存フォームへの組み込み

経路検索の利用は一定の条件を満たすことで既存のフォームでの利用が可能です。

## 1.1. 経路検索の利用条件

### 【経路検索の利用条件】

- ① 表明細を使ったフォームで行に、発駅、着駅、金額（運賃）、路線、券種 の5項目が含まれる。

日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	備考
1					片・往			
2			~		片・往			
3			~		片・往			
4			~		片・往			
5			~		片・往			
6			~		片・往			
7			~		片・往			
8			~		片・往			
9			~		片・往			
10			~		片・往			

※ 必要な項目が無い場合は隠しフィールド等を利用しフィールドを追加すると利用できます。

- ② 経路検索結果を書類に設定した後、運賃合計を再計算する合計フィールドの ID  
合計フィールドは明細外の項目で任意指定になります。

合計		円 (内 消費税 0 円)	支払方法	現金・振込
----	--	---------------	------	-------

- ③ 定期区間検索結果の通勤所要時間を戻すフィールドの ID  
時間、分を分けて登録する場合は夫々を数値フィールドで用意する任意指定のフィールドになります。

通勤所要時間
約 時間 分

(時間フィールド ID、分フィールド ID)

- ④ 定期区間検索結果の定期区間コードを戻すフィールドの ID  
文字フィールド、Hidden フィールドで作成する任意指定のフィールドになります。  
乗換連携で検索を行った際に生成される文字列を保存する項目になるため、文字フィールドで作成した場合は「不可視」属性を“true”に指定します。

## 1.2. 経路検索組み込みイメージ

交通費精算フォームを例に経路検索を組み込む手順を説明します。

### ■ 組み込み前のフォーム

明細行の右側に経路検索用の画面呼び出しボタンを追加し行毎に経路検索を実行できるようにします。

日付	行き先	用件	乗車区間	交通検索	区分	金額	備考
1			～		片・往		
2			～		片・往		
3			～		片・往		
4			～		片・往		
5			～		片・往		
6			～		片・往		
7			～		片・往		
8			～		片・往		
9			～		片・往		
10			～		片・往		

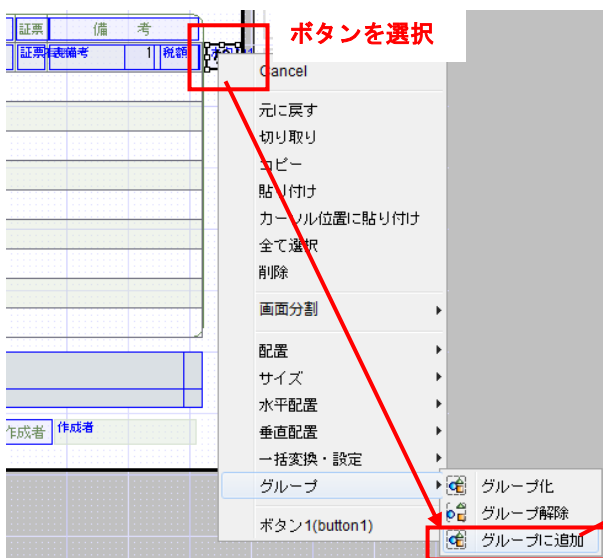
### ■ フォームを eFormMaker で修正します

明細行の横にボタンを配置し、検索用ボタンである事が判るよう「検」の文字をラベルに指定します。

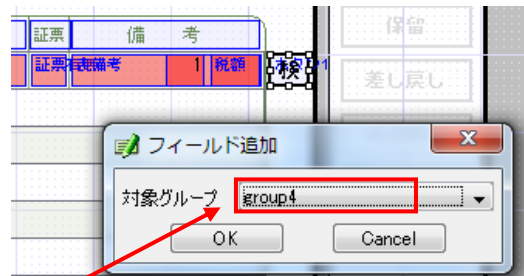
画像、ラベルフィールドを配置しても構いません。



### ■ ボタンを明細行のグループに追加します



ボタンを選択し【グループに追加】を実行



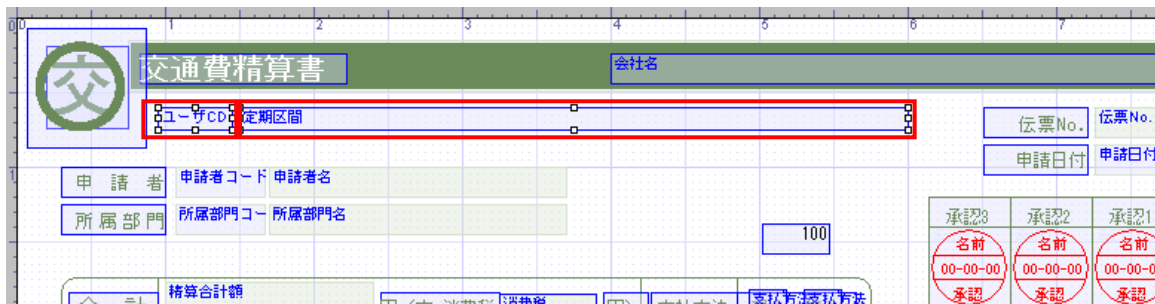
追加する明細行を選びグループに追加します。

■ 定期区間コードをマスタ検索で取得するための欄を追加

定期区間コード用のフィールドを文字フィールド、Hidden フィールドの何れかで追加します。

さらに、マスタ検索を利用する際のキーに利用する項目を文字フィールド、Hidden フィールドの何れかで追加します。

本項では検索用にユーザ CD を利用します。

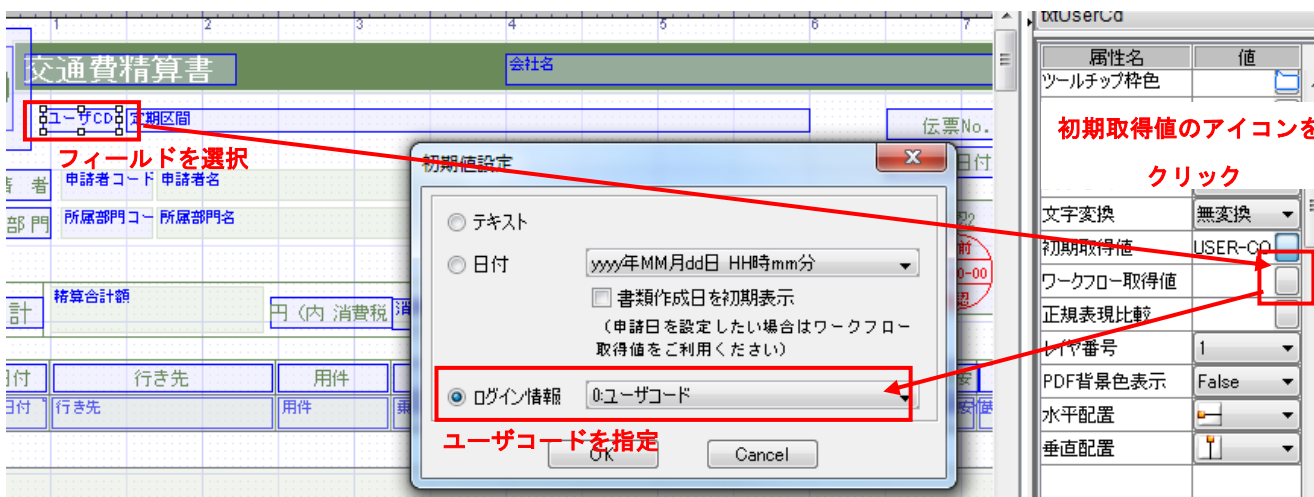


※ 文字フィールドを利用する場合は属性プロパティで「不可視」を“true”にします。

※ 定期区間コード指定による連携動作の確認を予定している場合、設定される内容を目視確認できるので定期区間コード用のフィールドはテキストフィールドで作成した方が良いでしょう。

■ ユーザ CD に初期値を指定

ユーザコードのワークフロー取得値に「ユーザコード」を設定します。



ユーザコードを利用し定期区間コードを設定する JavaScript を “onLoad 実行 JavaScript” に記述します。

【OnLoad 実行 JavaScript 例】

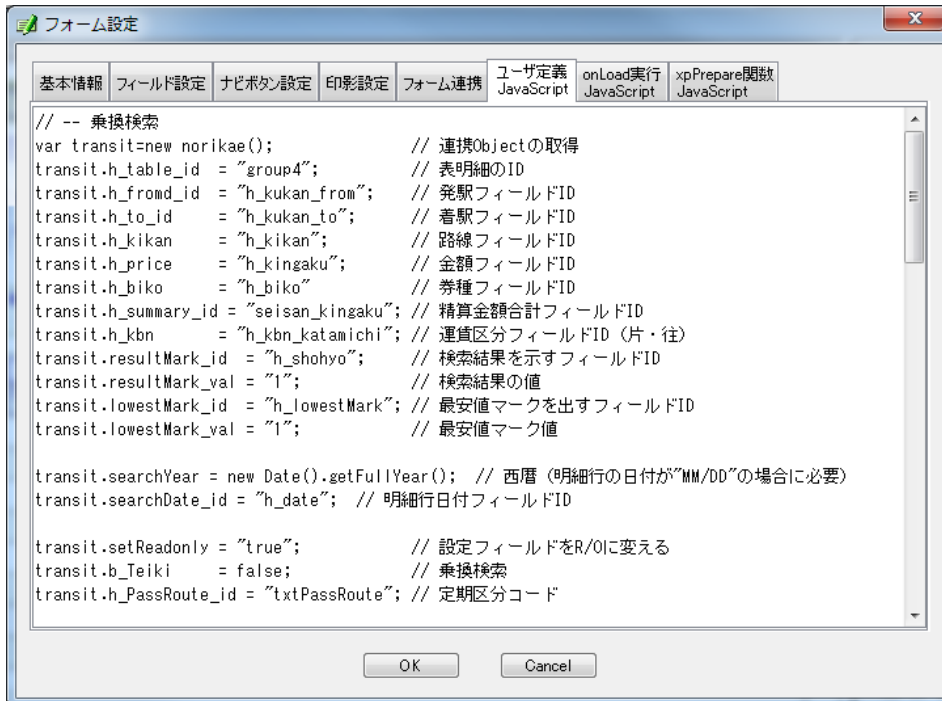
```
transit.init(); // 経路検索画面の初期化
```

```
var fld = XWEB.getXPObject("txtPassRoute", getCurrentPageNo());
if (fld.getValue().length==0) {
    getElementById_XP("txtUserCd").execMasterSeek();
}
```

追加する部分

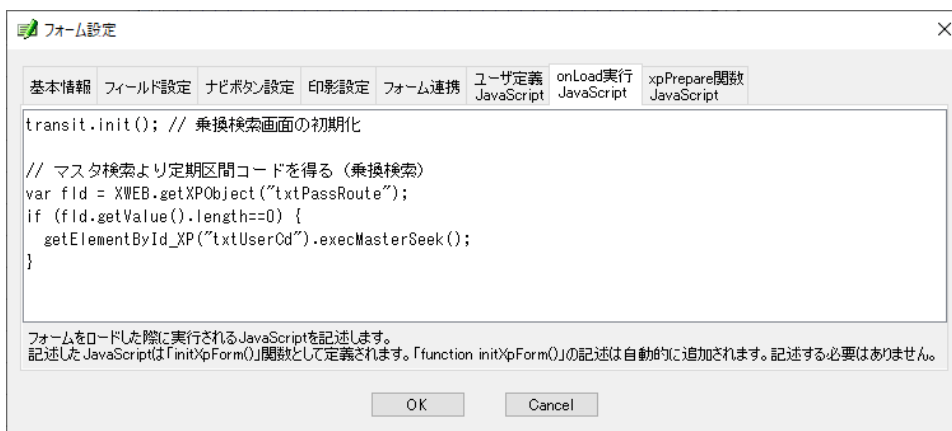
※ 赤字の部分は実際のフィールド ID に併せて修正します。

- 経路検索画面と連携するフォーム上のフィールド ID を“ユーザ定義 JavaScript”画面に設定します。連携オブジェクト変数の定義とフィールドの関係付けを記述します。



※ JavaScript の内容は『最小組み込みコード例』を参照してください。

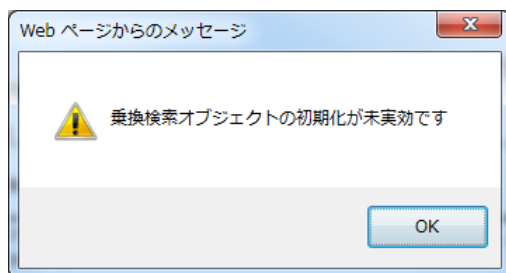
- 書類を開く際に経路検索の指定に従い各項目を初期化します。



※ 登録済み書類の明細項目に関する初期設定、定期区間コードの取得を行います。

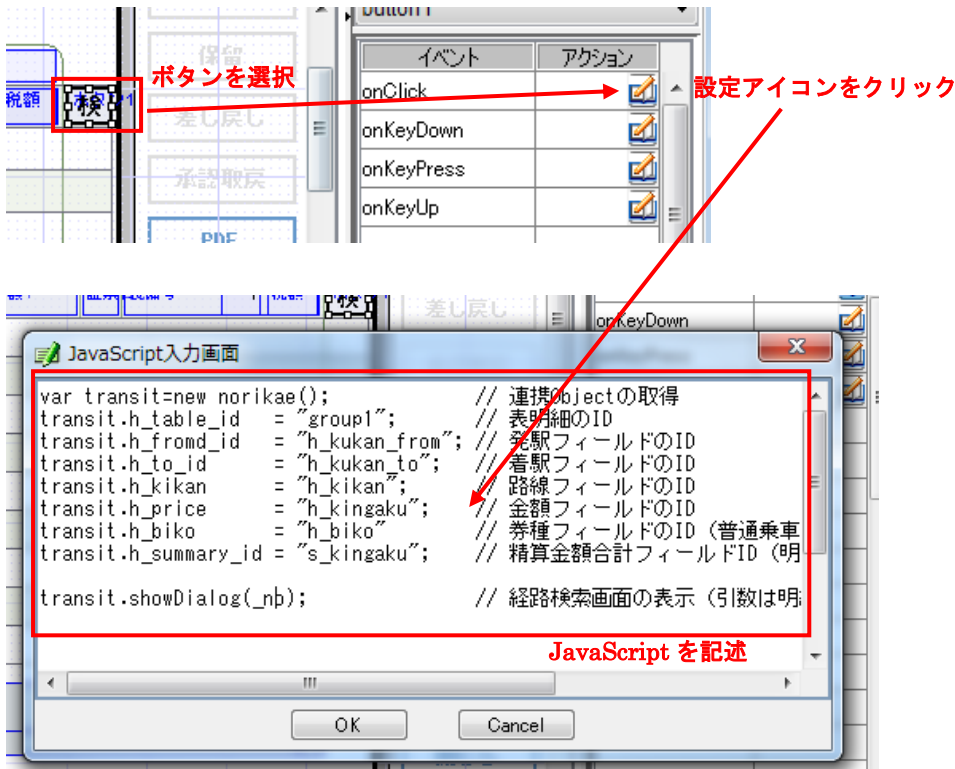
初期化の実行は省略できません。

記述を忘れると連携画面で警告表示が行われます。



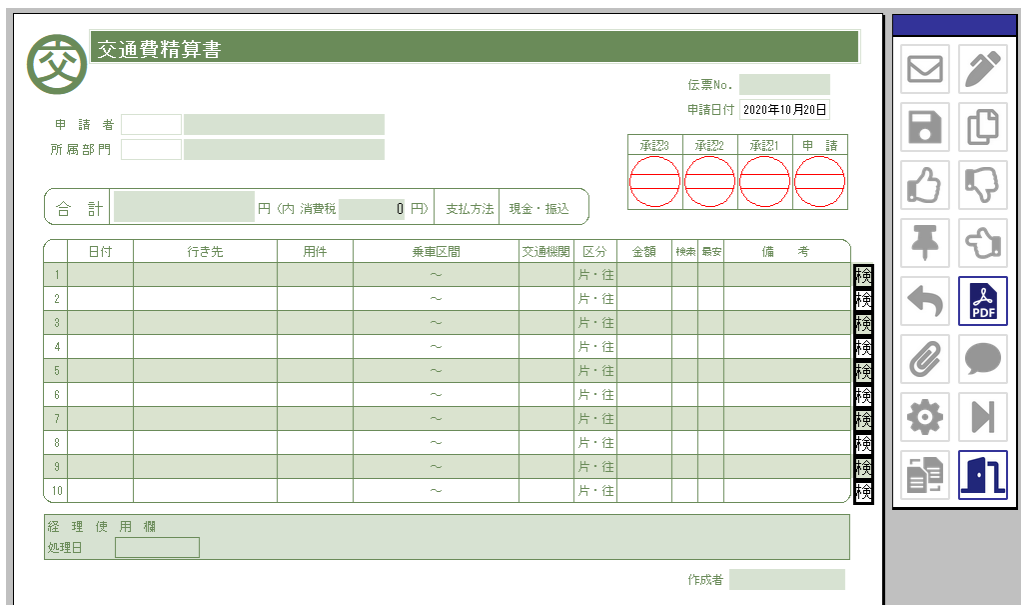
■ 経路検索画面を呼び出すための JavaScript を設定します

ボタンを選択しアクション・プロパティの onClick に連携用の JavaScript を記述します。



※ JavaScript の内容は『最小組み込みコード例』を参照してください。

■ 設定後のレイアウトを確認



※ eFormMaker のプレビュー操作の状態では経路検索を実行することはできません。  
経路検索は X-point Cloud にアップロードした後でのみ実行することができます。

■ X-point にアップロードし経路選択の動作を確認します。

フォームをアップロードし経路検索画面を表示して動作の確認を行います。



【確認が必要な最小限の項目】

- ① 発駅 . . . 入力中に候補リストが表示される
- ② 着駅 . . . 入力中に候補リストが表示される
- ③ ◆ . . . クリック毎に発駅と着駅が入れ替わる
- ④ 検索 . . . 発駅・着駅の入力後にクリックすると経路が表示される
- ⑤ 設定 . . . 経路検索後にクリックすると表示中の検索結果が書類にセットされる

## 1.3. 最小组み込みコード例

経路検索機能呼び出す際に記述する JavaScript の例を示します。例は連携に必要な最小限の内容です。検索条件の変更や片道、往復での金額計算が必要な場合は、「経路検索オプション 利用ガイド『6.4.連携 API 仕様』」の項を参考に JavaScript の修正・追記を行って下さい。

### 【経路検索の場合】

#### 1) ユーザ定義 JavaScript

```
var transit = new norikae();           // 連携 Object の取得
transit.h_table_id   = "group1";       // 表明細の ID
transit.h_from_id   = "h_kukan_from"; // 発駅フィールドの ID
transit.h_to_id     = "h_kukan_to";   // 着駅フィールドの ID
transit.h_kikan     = "h_kikan";      // 路線フィールドの ID
transit.h_price     = "h_kingaku";     // 金額フィールドの ID
transit.h_biko      = "h_biko";       // 券種フィールドの ID (普通乗車券、自由席、指定席など)
transit.h_summary_id = "s_kingaku";    // 精算金額合計フィールド ID (明細金額の値指定後に再計算を実行)
transit.h_PassRoute_id = "txtPassRoute"; // 定期区間コードフィールドの ID
```

※ 記載されている ID (右辺) は例です。実際のフォームに併せ修正を行って下さい。

#### 2) onLoad 実行 JavaScript

```
try {
    transit.init();
} catch(e) {
}
```

※ eFormMaker でプレビューする際、“init()” が未定義になるためエラーが発生します。

プレビュー時のエラーを防止するために try {} catch {} で囲みます。

定期区間コードを設定する場合は init() の後にマスタ検索の実行行を追加します

```
// マスタ検索より定期区間コードを得る (経路検索)
var fld = XWEB.getXPObject("txtPassRoute", currentPageNo());
if (fld.getValue().length==0) {
    getElementById_XP("txtUserCd").execMasterSeek();
}
```

※ 赤字の部分は実際のフィールド ID に併せて修正します。

#### 3) 経路検索呼び出し用ボタンの onClick に記述する JavaScript

```
if (!isEditMode()) {
    return; // 編集不可の場合は呼び出さない
}
transit.showDialog(_no); // 経路検索画面の表示 (引数は明細の行番号が入る)
```

## 【定期区間検索の場合】

### 1) ユーザ定義 JavaScript

```
var transit = new norikae();           // 連携 Object の取得
transit.h_table_id    = "group1";      // 表明細の ID
transit.h_fromd_id   = "txtKukan_from"; // 発駅フィールドの ID
transit.h_to_id      = "txtKukan_to";  // 着駅フィールドの ID
transit.h_kikan      = "txtKoutukikan"; // 路線フィールドの ID
transit.h_price      = "intTeikidai";  // 金額フィールドの ID
transit.b_Teiki      = "true";         // 定期区間検索モードを指定
transit.h_summary_id = "intGoukei";    // 金額合計フィールド ID (明細金額の値指定後に再計算を実行)
transit.h_PassRoute_id = "txtPassRoute"; // 定期区間コード返却フィールド ID
transit.timeRequired = "intJikan, intFun"; // 通勤時間フィールド ID (時間、分をカンマ区切りで指定)
transit.h_PassRoute_id = "txtPassRoute"; // 定期区間コードフィールドの ID
```

※ 記載されている ID (右辺) は例です。実際のフォームに併せ修正を行って下さい。

### 2) onLoad 実行 JavaScript

```
transit.init();
```

### 3) 書類に入力された経路全体の確認画面を表示するため onClick に記述する JavaScript

```
if (!isEditMode()){
    return; // 編集不可の場合は呼び出さない
}
transit.confirmDialog();
```

※ 連携にボタンフィールドを利用する場合は、isEditMode() による判定を行う必要はありません。

イメージフィールドを利用して連携を行う場合は isEditMode() が必須になります。

## 第2章. そのほか

### 2.1. 特定の駅を経由したい場合

経路検索オプションの経路検索では経由駅を指定することができます。発駅・着駅の下にある経由欄に駅名を指定します。

【指定例】

発駅 渋谷 ← 着駅 六本木

経由 青山一丁目 追加

閉じる

※ 複数の経由駅を指定する場合、経由駅入力欄の右側にある【追加】ボタンで欄を追加できます。  
最大15駅まで指定できます。(追加ボタンは一番上の経由欄のみで利用できます)

【検索結果】

経路1 2016年2月5日(金) 早 安 楽

所要時間 20分(乗車 7分、他 13分) 距離 4km

運賃 269円 乗り換え 1回

渋谷 ICカード乗車券 269円 ▼

5分  
2駅  
2.7km  
東京メトロ銀座線

青山一丁目 ← 経由駅が入ります

2分  
1.3km  
都営大江戸線

六本木

【書類への設定】

合計 269 円 (内 消費税 0 円) 支払方法 現金・振込

日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	備考
1			渋谷 ~ 青山一丁目	東京メトロ	往	269	v	
2			青山一丁目 ~ 六本木	都営大江	往		v	

## 2.2. 経路検索を利用した明細であることを書類上に示す方法

交通費精算フォームで経路検索を利用した場合、入力されている経路が手入力であるのか、経路検索のどちらで入力されているのかにより記入されている経路の妥当性を確認する作業の要否が変わります。

フォームに経路検索で指定された明細であるか事を示すには“resultMark\_id”プロパティを利用し明細行にチェックを行います。例として、交通費精算書フォームの「検索」列に「v」を表示する場合を説明します。

- 「交通費精算書フォーム」に「検索」列を作成します。

- 「検索」列の ID 名を「h\_norikae\_mark」に、タブ移動順序を「-1」にします。

属性名	値
タグ名	文字フィール...
ID	h_norikae_m...
タイプ	9
X座標	5
Y座標	250
幅	20
高さ	20
色	#000000
初期値	
最大入力文字数	
IMEモード	Off
無効	False
編集禁止	True
タブ移動順序	-1

※ 「検索」列に指定された項目は経路検索の初期化処理で編集禁止に設定されます。

- 経路検索画面を呼び出す JavaScript に「検索」列に関する指定を追加します。

```

var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.resultMark_id = "h_norikae_mark"; // 「検索」列の ID 名
transit.resultMark_val = "v"; // 検索結果をセットする際、列に設定する値
    
```

■ 「検索」列フィールドの追加、JavaScript の修正を行ったフォームをアップロードします。

■ フォームを表示し動作を確認します。

経路検索の画面を表示し「東京・大阪」間の検索結果を設定します。

明細が設定された行の「検索」列に「v」が表示されれば完成です。

交通費精算書

伝票No.   
 申請日付 2016年01月15日

申請者    
 所属部門

承認3	承認2	承認1	申請

合計

27,240

円 (内 消費税  円)

支払方法

現金・振込

日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	備考
1	(株)交通費精算		東京 ~ 新大阪	J R新幹線	片・往	27,240	v	自由席 (9740)
2			新大阪 ~ 大阪	J R京都線	片・往		v	
3			~		片・往			

【参考】

手入力の場合は「検索」列の表示が行われません。

交通費精算書

伝票No.   
 申請日付 2016年01月15日

申請者    
 所属部門

承認3	承認2	承認1	申請

合計

27,240

円 (内 消費税  円)

支払方法

現金・振込

日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	備考
1	(株)交通費精算		東京 ~ 新大阪	新幹線	片・往	27,240		
2			新大阪 ~ 大阪	京都線	片・往			
3			~		片・往			

## 2.3. 片道、往復の指定を行う方法

経路検索で表示される運賃を片道分、往復分のどちらで初期表示を行うか“kvn\_val”プロパティで指定することができます。

### 【片道の場合】

```
var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.kvn_val = "0"; // 片道
```

### 【往復の場合】

```
var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.kvn_val = "1"; // 往復
```

※ 経路検索画面の「料金設定」項目で、片道、往復の切り替えができます。

※ 定期区間検索の場合「kvn\_val」の指定は無視されます。

## 2.4. 経路検索で最も安価な経路を選択した事を書類上に表示する方法

経路検索は結果を複数表示します。選択する候補により、運賃が最も安い、移動時間が最も短い、乗換が少ない等が異なります。

交通費精算書の場合運賃が安価なルートであるか否かを知る必要が生じる場合があります。

ここでは、経路検索で選ばれたルートが安価なルートであるか否かを書類上にセットする方法を説明します。

■ 「交通費精算書フォーム」に「最安」列を作成します。

日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	最安	備考
1 表日付	行き先	用件	乗車区間from ~ 乗車区間to	交通機関	交通機関	金額	検索	最安	備考

■ ID 名を「h\_lowestMark」に、タブ移動順序を「-1」にします。

属性名	値
タグ名	最安値マーク
ID	h_lowestMark
タイプ	16
X座標	686
Y座標	250
幅	22
高さ	20
色	#000000 <input type="checkbox"/>
初期値	
チェックマーク	安
非チェックマーク	
チェック値	1
非チェック値	
無効	False
タブ移動順序	-1
移動禁止	False

※ 「最安」列に指定された項目は経路検索の初期化処理で編集禁止に設定されます。

※ 例ではチェックボックスに1をセットした際に「安」が表示されるようにしています。

■ 経路検索画面を呼び出す JavaScript に「検索」列に関する指定を追加します。

```
var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.lowestMark_id = "h_lowestMark"; // 最安値を表示する文字フィールドの ID 名
transit.lowestMark_val = "1"; // 最安値の際、フィールドにセットする文字列
```

■ フィールドの追加、JavaScript の修正を行ったフォームをアップロードします。

■ フォームを表示し動作を確認します。

経路検索を行い候補の右肩に「安」の文字がある候補を設定します。



追加したフィールドに「lowestMark\_val」プロパティに指定した文字列が表示されます。

交通費精算書									
申請者		u001 勝太郎		伝票No.					
所属部門		1310 営業1課		申請日付		2016年01月28日			
承認3		承認2		承認1		申請			
承認3		承認2		承認1		申請			
合計		1,288 円 (内消費税 0 円)		支払方法		現金・振込			
日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	最安	備考
1			東京 ~ 横浜	J R東海	片・往	928	v	安	自由席 (0)
2			横浜 ~ みなとみらい	みなとみ	片・往	360	v	安	

経路検索を「品川・大阪」間、「大阪・姫路」間で分けて行い「大阪・姫路」間は検索結果の最安値を使用しなかった場合、マークは「品川・大阪」間の検索結果のみにセットされます。

合計		16,840 円 (内消費税 0 円)		支払方法		現金・振込			
日付	行き先	用件	乗車区間	交通機関	区分	金額	検索	最安	備考
1			品川 ~ 新大阪	J R新幹線	往	13,620	v	安	自由席 (4870)
2			新大阪 ~ 大阪	J R京都	往		v	安	
3			大阪 ~ 新大阪	J R京都	往	1,490	v		
4			新大阪 ~ 姫路	J R新幹線	往	1,730	v		自由席 (1730)
5									

## 2.5. 経路検索で設定した明細行のフィールドを編集禁止にする方法

経路検索で設定した明細行を申請者が修正できないようにします。検索で設定した運賃の行であるか否かを区別するようにした書類の場合に必要となり、編集禁止にフィールドが指定されるためには次の3条件が成立する場合があります。

### 【条件】

- ① 連携オブジェクトの setReadOnly プロパティに“true”が指定されている。
- ② 結果設定明細をマークする resultMark\_id プロパティ、マークするフィールドに指定する値 resultMark\_val プロパティに ID と値が指定されている。
- ③ resultMark\_id が示すフィールドの設定値が resultMark\_val と一致する値である。

### ■ 編集禁止の制御を行う場合の JavaScript 記述例

#### 【編集可の場合】

```
var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.setReadOnly = "false"; // 編集可
transit.resultMark_id = "h_shohyo";
transit.resultMark_val = "1";
```

#### 【編集不可の場合】

```
var transit = new norikae();
transit.h_table_id = "group1";
transit.h_fromd_id = "h_kukan_from";
transit.h_to_id = "h_kukan_to";
transit.h_kikan = "h_kikan";
transit.h_price = "h_kingaku";
transit.h_biko = "h_biko";
transit.setReadOnly = "true"; // 編集不可
transit.resultMark_id = "h_shohyo";
transit.resultMark_val = "1";
```

フィールドの編集禁止は検索画面で【設定】を行ったタイミングで、設定内容を戻すフィールドに対してのみ行われます。上記記述例の場合、発駅、着駅、交通機関、金額、券種（備考）の5フィールドが編集禁止に指定されます。

※ 編集禁止に指定された行から表示した経路検索画面で【行クリア】を実行すると、指定された項目を空白に変更するとともに編集禁止が解除され編集可になります。

### ■ 編集禁止を利用する場合の注意

編集禁止の設定は、経路検索画面で【設定】を行ったタイミング、及び書類表示のタイミングに行われます。

The screenshot shows a web application interface for route search. At the top, there's a header with a logo and the title '交通費精算書' (Commute Expense Statement). Below the header are several input fields for user information, including '会社名' (Company Name), 'ユーザーCD', '定期区間', '伝票No.', '申請日付', '申請者', '申請者コード', '申請者名', '所属部門', '所属部門コード', '所属部門名', and '最安値マーク'. There are also buttons for '承認3', '承認2', '承認1', and '申請'. A table at the bottom displays search results with columns for '日付', '行き先', '用件', '乗車区間', '交通機関', '区分', '金額', '検索', and '備考'. The '乗車区間' column is further divided into 'from' and 'to'. Red boxes highlight the '乗車区間from', '乗車区間to', '交通機関', '金額', and '備考' fields in the first row of the table, indicating they are locked for editing.

※ 経理部門などで内容のチェック、修正を行う場合は必要に応じて編集を許可します。

※ 申請ステップ以外で経路検索が起動しないように、連携用の JavaScript の先頭にステップ判定を追加します。

## 2.6. 経路検索を利用するステップを制限する場合

経路検索、定期区間検索の利用を申請者に制限する場合、検索画面を呼び出すクリックボタンに次の記述を追加しておきます。

```
if (getCurrentStepNo() != 0) {  
    alert('申請ステップ以外は利用できません');  
    return;  
}
```

## 2.7. 承認ステップでの編集を可にする場合

経路検索で指定された項目を経理処理ステップなど特定のステップで、申請された内容から修正したい場合があります。そのような場合は、onLoad で init() 処理を実行する前で setReadOnly プロパティを変更します。

【onLoad 実行 JavaScript】

```
if ( getCurrentStep() == 4 ) {  
    transit.setReadOnly = "false";  
} else {  
    transit.setReadOnly = "true";  
}  
transit.init();
```

**■改訂履歴**

改版	改版内容
2021年4月1日版	初版
2022年7月20日版	対応ブラウザから「Internet Explorer」を削除
2022年10月12日版	対応ブラウザに「Safari iOS15」を追加
2023年3月3日版	「1.2. 経路検索組み込みイメージ」「1.3. 最小組み込みコード例」のマスタ検索サンプルコードを修正
2025年6月12日版	対応ブラウザに「Mac OS Safari」を追加
2025年6月30日版	対応環境より iOS 16 を削除