



目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 目的
 - 2.2. 前提条件
 - 2.3. 準備
- 3. 概要
 - 3.1. ユーザアプリケーションと IM-Workflow の関係
 - 3.1.1. システム案件ID
 - 3.1.2. ユーザーデータID
 - 3.1.3. 案件プロパティ
 - 3.2. リクエストパラメータ
 - 3.3. 案件処理系APIと画面動作仕様の違い
- 4. 画面の作成（標準処理画面）
 - 4.1. 申請画面の呼び出し
 - 4.1.1. スクリプト開発モデル
 - 4.1.2. JavaEE開発モデル
 - 4.1.3. JSP、TERASOLUNA Server Framework for Java (5.x)
 - 4.2. 一時保存画面の呼び出し
 - 4.2.1. スクリプト開発モデル
 - 4.2.2. JavaEE開発モデル
 - 4.2.3. JSP、TERASOLUNA Server Framework for Java (5.x)
 - 4.3. 申請（起票案件）／再申請／処理画面の呼び出し
 - 4.3.1. スクリプト開発モデル
 - 4.3.2. JavaEE開発モデル
 - 4.3.3. JSP、TERASOLUNA Server Framework for Java (5.x)
 - 4.4. 確認画面の呼び出し
 - 4.4.1. スクリプト開発モデル
 - 4.4.2. JavaEE開発モデル
 - 4.4.3. JSP、TERASOLUNA Server Framework for Java (5.x)
 - 4.5. 詳細画面の呼び出し
- 5. 画面の作成（処理モジュール）
 - 5.1. ワークフロー処理モジュールの呼び出し
 - 5.1.1. 申請モジュール
 - 5.1.2. 一時保存モジュール
 - 5.1.3. 処理モジュール
 - 5.1.4. 確認モジュール
 - 5.2. クライアントサイド JavaScript API
 - 1. imWorkflow
- 6. ユーザプログラムの作成
 - 6.1. 案件開始処理
 - 6.1.1. パラメータ
 - 6.1.2. 返却値
 - 6.2. 案件終了処理
 - 6.2.1. パラメータ
 - 6.2.2. 返却値
 - 6.3. アクション処理
 - 6.3.1. パラメータ
 - 6.3.2. 返却値
 - 6.4. 到達処理
 - 6.4.1. パラメータ
 - 6.4.2. 返却値
 - 6.5. 分岐開始処理
 - 6.5.1. パラメータ
 - 6.5.2. 返却値
 - 6.6. 分岐終了処理
 - 6.6.1. パラメータ
 - 6.6.2. 返却値
 - 6.7. 案件終了処理（トランザクションなし）
 - 6.7.1. パラメータ
 - 6.7.2. 返却値
- 7. その他プログラムの作成

- 7.1. 未完了案件削除処理リスナー
- 7.2. 完了案件削除処理リスナー
- 7.3. 過去案件削除処理リスナー
- 7.4. 案件退避処理リスナー
- 8. Appendix
 - 8.1. テンプレート
 - 8.2. サンプルプログラム
 - 8.2.1. 画面
 - 8.2.2. ユーザプログラム
 - 8.2.3. リスナー
- 9. カスタマイズ
 - 9.1. 呼び出し画面の初期表示値指定
 - 9.1.1. 指定可能なパラメータ
 - 9.1.2. 実装例
 - 9.2. 処理対象者プラグインの作成
 - 9.2.1. 対象ノード（拡張ポイント）
 - 9.2.2. サンプルの説明
 - 9.2.3. サンプルの実行準備
 - 9.2.4. サンプルの実行
 - 9.2.5. 処理対象者プラグインの構成
 - 9.3. 画面入力情報の保持
 - 9.4. 呼び出し画面からのコールバック関数の指定
 - 9.4.1. 実装例
 - 9.4.2. 標準画面を非同期で実行する場合の注意点
 - 9.4.3. 特記事項
 - 9.5. 処理完了後の画面遷移
 - 9.5.1. 遷移先を指定するためのパラメータ
 - 9.5.2. 遷移先画面が受け取ることのできるリクエストパラメータ
 - 9.5.3. 特記事項
 - 9.6. ユーザコンテンツと連続処理／連続確認の連携方法
 - 9.6.1. 連続処理／連続確認を継続実行する
 - 9.6.2. 連続処理／連続確認を中断する
 - 9.7. PC版ユーザコンテンツをスマートフォン用画面としても利用する
 - 9.7.1. 必要な作業
 - 9.8. ユーザコンテンツ画面への不正な直接アクセスを抑制する
 - 9.8.1. 対象者
 - 9.8.2. 対象バス種別
 - 9.8.3. 対応方法
 - 9.9. 動的処理対象者設定機能
 - 9.9.1. 機能概要
 - 9.9.2. 利用方法
 - 9.9.3. 利用例
 - 9.9.4. パラメータ詳細

変更年月日	変更内容
2012-10-01	初版
2012-12-21	第2版 下記を追加・変更しました <ul style="list-style-type: none"> 「リクエストパラメータ」の「imwSerialProcParams」に関する説明を追加・修正しました。 「実装例」にコールバック関数が受け取れる情報について説明を追加しました。 「特記事項」、「IM-Workflow バージョン8.0.2 における改善」を追加しました。 「処理完了後の画面遷移」章の見出しを変更しました。また「処理完了後の画面遷移」章以下の章立てを見直し、説明を追加しました。 <ul style="list-style-type: none"> 変更前：処理画面から受け取るリクエストパラメータ 変更後：処理完了後の画面遷移 「ユーザコンテンツと連続処理／連続確認の連携方法」を追加しました。
2013-04-01	第3版 下記を追加・変更しました <ul style="list-style-type: none"> 「前提条件」を追加しました。 「リクエストパラメータ」に、スマートフォン用画面の説明を追記しました。 「画面の作成（標準処理画面）」に、スマートフォン用画面の説明を追記しました。 「画面の作成（標準処理画面）」の制限事項に、章の説明を追記しました。 「画面」に、スマートフォン用画面の説明を追記しました。 「呼び出し画面の初期表示値指定」に、スマートフォン用画面の説明を追記しました。 「画面入力情報の保持」に、スマートフォン用画面の説明を追記しました。 「呼び出し画面からのコールバック関数の指定」に、スマートフォン用画面の説明を追記しました。 「処理完了後の画面遷移」に、スマートフォン用画面の説明を追記しました。 「ユーザコンテンツと連続処理／連続確認の連携方法」に、スマートフォン用画面の説明を追記しました。 「PC版ユーザコンテンツをスマートフォン用画面としても利用する」を追加しました。 上記のほか、誤字脱字などを修正しました。
2013-07-01	第4版 下記を追加・変更しました <ul style="list-style-type: none"> 「実装例」の実装サンプル記述を修正しました。 「標準画面を非同期で実行する場合の注意点」を追加しました。
2013-10-01	第5版 下記を追加・変更しました <ul style="list-style-type: none"> 「リクエストパラメータ」の記述を修正しました。 <ul style="list-style-type: none"> 「imwAuthUserCode」に関するただし書きを削除しました。
2014-01-01	第6版 下記を追加・変更しました <ul style="list-style-type: none"> サンプルjavaソースのプログラムパスを修正しました。 「リクエストパラメータ」の記述を修正しました。 <ul style="list-style-type: none"> 「imwGroupId」が非推奨である旨を記述しました。 「ユーザコンテンツ画面への不正な直接アクセスを抑制する」を追加しました。
2014-04-01	第7版 下記を追加・変更しました <ul style="list-style-type: none"> 「案件処理系APIと画面動作仕様の違い」を追加しました。
2014-08-01	第8版 下記を追加・変更しました <ul style="list-style-type: none"> 「リクエストパラメータ」に再申請時のimwAuthUserCodeに関する説明を追加しました。
2014-09-01	第9版 下記を追加・変更しました <ul style="list-style-type: none"> 「画面の作成（標準処理画面）」に開発モデル（パス種別）に対する説明を追加しました。 「詳細画面の呼び出し」を追加しました。 「処理対象者プラグインの構成」の説明を追加しました。
2014-12-01	第10版 下記を追加・変更しました <ul style="list-style-type: none"> 「リクエストパラメータ」の「imwSysDateTargetExpandFlag」に関する説明を追加・修正しました。

変更年月日	変更内容
2014-12-24	<p>第11版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「意図しないURLに対するバリデーション」を追加しました。 「動的処理対象者設定機能」を追加しました。
2015-04-01	<p>第12版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「リクエストパラメータ」に「imwShortCutFlag」を追加しました。 「意図しないURLに対するバリデーション」に関する説明を修正しました。
2015-08-01	<p>第13版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「案件終了処理（トランザクションなし）」を追加しました。 「テンプレート」に「案件終了処理（トランザクションなし）」を追加しました。 「動的処理対象者設定機能」にスマートフォン版の説明を追加しました。
2015-12-01	<p>第14版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「案件終了処理（トランザクションなし）」に、エラー発生時のロールバックに関する説明を追加しました。
2016-04-01	<p>第15版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「画面の作成（標準処理画面）」に、TERASOLUNA Server Framework for Java (5.x) に関する説明を追加しました。 「画面の作成（標準処理画面）」の制限事項を削除し、リリースノート に記載しました。 「ユーザプログラムの作成」に、パラメータと返却値に関する説明を追加しました。 連携先に TERASOLUNA Server Framework for Java (5.x) を追加しました。
2016-08-01	<p>第16版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「案件終了処理」の戻り値の説明を変更しました。 「到達処理」の戻り値の説明を変更しました。 「案件終了処理（トランザクションなし）」の戻り値の説明を変更しました。 「リクエストパラメータ」のimwUserCode、imwNodeIdの説明を追加しました。
2017-04-01	<p>第17版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「到達処理」のパラメータ「メール送信可否」の説明を追加しました。 「画面」のThemeBuilderの指定内容を修正しました。
2017-08-01	<p>第18版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 「案件終了処理」のパラメータ「メール送信可否」の説明を追加しました。 以下のユーザプログラムに対し、複数のユーザプログラムを設定されている際にエラーが発生した場合の動作の説明を追加しました。 <ul style="list-style-type: none"> 案件開始処理 案件終了処理 アクション処理 案件終了処理（トランザクションなし） 以下のユーザプログラムに対し、resultFlagの説明を修正しました。 <ul style="list-style-type: none"> 案件開始処理 案件終了処理 アクション処理 到達処理 分岐開始処理 案件終了処理（トランザクションなし） 「処理対象者プラグインの作成」の説明を変更しました。
2017-12-01	<p>第19版 下記を追加・変更しました</p> <ul style="list-style-type: none"> 以下のページに「動的処理対象者設定」におけるノードスキップの方法を追加しました。 <ul style="list-style-type: none"> 動的承認ノード・確認ノード 横配置ノード・縦配置ノード 「アクション処理」にアクション処理におけるユーザパラメータの説明を追加しました。

変更年月日	変更内容
2018-04-01	<p>第20版 下記を変更しました</p> <ul style="list-style-type: none"> 「処理対象者プラグインの構成」に処理対象者プラグインに関するコラムを追加しました。
2019-12-01	<p>第21版 下記を変更しました</p> <ul style="list-style-type: none"> コンテンツ定義の画面定義のパス種別「JSP or Servlet」を「URL」に名称を変更しました。
2020-08-01	<p>第22版 下記を追加しました</p> <ul style="list-style-type: none"> 「画面の作成 (処理モデル)」を追加しました。
2020-12-01	<p>第23版 下記を変更しました</p> <ul style="list-style-type: none"> 「画面」にコピー新規機能の説明を追記しました。 「画面の作成 (処理モデル)」の<im_workflow/js/api_base.js>の読み込みに関する説明を追加しました。
2021-04-01	<p>第24版 下記を変更しました</p> <ul style="list-style-type: none"> 「showApply」の「horizontalNodeConfigs/verticalNodeConfigs」について説明を変更しました。 「showProcess」の「horizontalNodeConfigs/verticalNodeConfigs」について説明を変更しました。
2021-08-01	<p>第25版 下記を変更しました</p> <ul style="list-style-type: none"> 「showApply」のuserDataIdの説明を修正しました。 「apply」のuserDataIdの説明を修正しました。
2023-10-01	<p>第26版 下記を変更しました</p> <ul style="list-style-type: none"> モーダル表示に関する新規パラメータの説明を追加しました。 <ul style="list-style-type: none"> 対象ページ <ul style="list-style-type: none"> 「showApply」 「showProcess」 対象パラメータ <ul style="list-style-type: none"> dynamicNodeConfigs.readonlyEnableToggle horizontalNodeConfigs.dispatchControl verticalNodeConfigs.dispatchControl 「アクション処理」のパラメータに項目を追加しました。 「処理対象者プラグインの構成」に処理対象者プラグインの作成に関するコラムを追加しました。
2026-04-01	<p>第27版 下記を変更しました</p> <ul style="list-style-type: none"> 「アクション処理」のパラメータの説明を追加しました。 モーダル表示、および案件処理を行うための JavaScript API に関するパラメータの説明を追加しました。 <ul style="list-style-type: none"> 対象ページ <ul style="list-style-type: none"> 「クライアントサイドJavaScript API」 対象パラメータ <ul style="list-style-type: none"> optionalParameter.userParameter optionalParameter.formaParam

目的

本書は、IM-Workflow で利用することが可能な画面およびモジュールを作成する方法について説明します。

本書は、IM-Workflow の機能を使用する方法を記述しています。

本書で使用するサンプルプログラムはあくまでも、IM-Workflow の機能およびAPI等の使用方法を理解することに主眼をおいています。そのため、必ずしも最適なコーディング方法とはいえない方法もあえて取っている箇所があります。あくまでも、サンプルとしての位置付けでとらえるようにしてください。

前提条件

本書に記述されているサンプルプログラムは、JavaEE開発モデルおよびスクリプト開発モデルで記述されています。

そのため、JavaEE開発モデルおよびスクリプト開発モデルに関する理解は必須です。

各開発モデルに関しては、付属する各種マニュアルおよびAPIリストを参照してください。

本書を理解するには、基本的な IM-Workflow に関する理解が必要です。付属する各種マニュアル、APIリスト、および制限事項を参照してください。

本書に記述されているサンプルプログラムのパスは、以下のディレクトリ配下のパスです。

< (展開したwar) /WEB-INF/ >

準備

IM-Workflow のサンプルプログラムを実行するための準備をします。

「[intra-mart Accel Platform セットアップガイド](#)」を参考に、IM-Workflow が動作する環境を構築します。

製品のインストール後は、システム管理者でログインし、メニュー[テナント環境セットアップ]より、テナント環境セットアップを行い、[サンプルデータセットアップ](#) も必ず行ってください。

本書に記述されているJavaEE開発モデルの[javaファイル]は、配置する場所を示します。

実際に配置されているファイルは、[classファイル]です。

JavaEE開発モデル[javaファイル]のサンプルプログラムについては、製品メディアに保存されています。

また、製品最新情報ダウンロードページ ([Products Information Site](#)) から入手することもできます。

概要

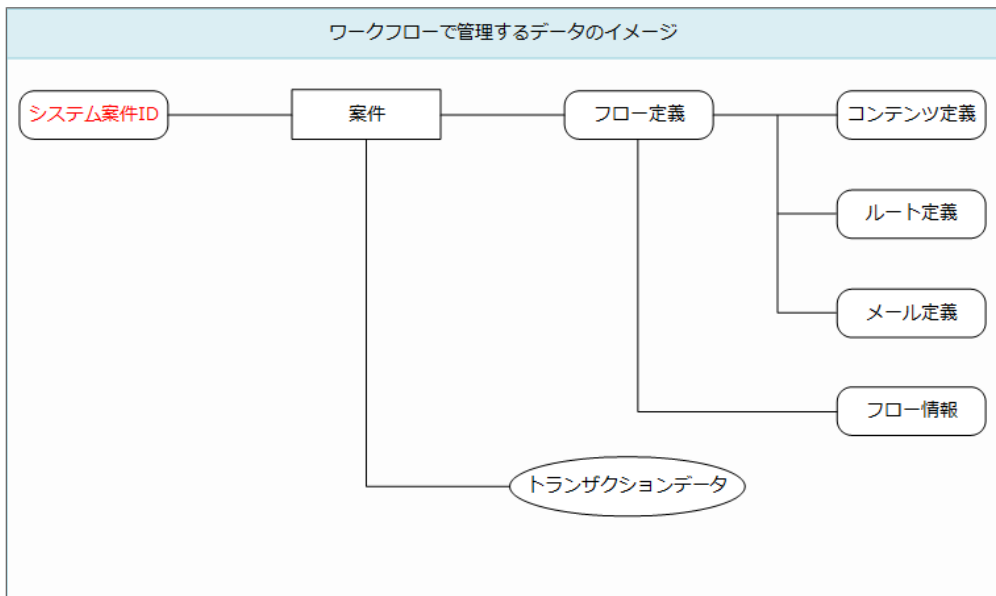
ユーザアプリケーションと IM-Workflow の関係

ユーザアプリケーションデータと IM-Workflow のデータは、それぞれ「ユーザデータID」と「システム案件ID」という2つのキーによって一意に特定されます。2つのキーは1対1の関係で関連付けられます。



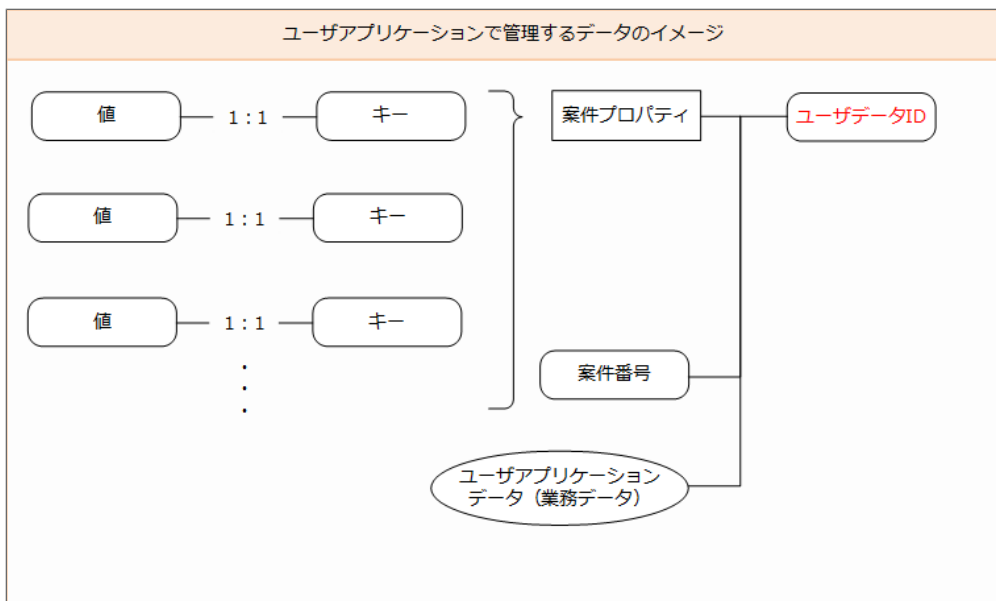
システム案件ID

システム案件IDとは、IM-Workflow において一意となるキーです。IM-Workflow のモジュールにおいて採番され、外部より指定することはできません。システム案件IDは、IM-Workflow のAPIやタグライブラリ等で案件を特定するために使用され、画面等に表示されることはありません。



ユーザデータID

ユーザデータIDとは、ユーザアプリケーション側で一意となるキーとして、ユーザアプリケーションで採番するキーです。申請または起票を行う場合に、IM-Workflow の提供するAPIおよびタグライブラリの引数として渡されます。ユーザデータIDは、システム案件IDと同様に、IM-Workflow のAPIやタグライブラリ等で案件を特定するために使用され、画面等に表示されることはありません。



案件プロパティ

案件プロパティとは、いわゆるKey-Value Storeです。「Key (キー)」と「Value (値)」のペアからなるデータモデルを案件単位に IM-Workflow で保存します。IM-Workflow が提供するAPIを通じて、任意のタイミングにおいて、登録・更新・削除および取得が可能です。

また、IM-Workflow が提供する各種一覧画面に表示したり、分岐条件におけるルール定義で参照する値として使用することができます。

リクエストパラメータ

各種一覧画面から呼び出される申請および処理等の画面で、必要な情報をリクエストパラメータとして受け取る事ができます。

No	パラメータ（物理名）	パラメータ（論理名）	詳細
1	imwGroupId	グループID	非推奨です。 過去との互換のために残されています。
2	imwUserCode	処理者CD（ログインユーザ本人のユーザコード）	
3	imwPageType	画面種別	表示された画面の種別 <ul style="list-style-type: none"> ・申請画面 ・一時保存画面 ・申請（起票案件）画面 ・再申請画面 ・処理画面 ・確認画面 ・処理詳細 ・参照詳細 ・確認詳細 ・過去案件詳細 ・申請画面（スマートフォン用） ・一時保存画面（スマートフォン用） ・申請（起票案件）画面（スマートフォン用） ・再申請画面（スマートフォン用） ・処理画面（スマートフォン用） ・確認画面（スマートフォン用）
4	imwUserDataId	ユーザデータID	
5	imwSystemMatterId	システム案件ID	
6	imwNodeId	処理対象ノードID	処理対象のノードが縦配置・横配置ノードの場合は、展開後のノードIDが渡されます。
7	imwArriveType	到達種別	
8	imwAuthUserCode	権限者CD	ログインユーザが案件を処理する際に選択可能な権限者コードです。具体的には、ログインユーザ本人や、ログインユーザを代理先として代理設定されている場合は代理元ユーザコードが該当します。 権限者が複数存在する場合、当パラメータは配列で渡されます。* ただし、権限者が複数存在する場合でも、申請/一時保存画面表示の際は一覧上で権限者が特定されているため、特定済みの権限者コードのみが渡されます。
9	imwApplyBaseDate	申請基準日	「yyyy/mm/dd」形式
10	imwContentsId	コンテンツID	
11	imwContentsVersionId	コンテンツバージョンID	
12	imwRouteId	ルートID	
13	imwRouteVersionId	ルートバージョンID	
14	imwFlowId	フローID	
15	imwFlowVersionId	フローバージョンID	
16	imwSerialProcParams	連続処理パラメータ	連続処理用のパラメータ IM-Workflow バージョン8.0.2 より、当パラメータは無効になりました。 必ず空文字（" "）が渡されるため、ユーザコンテンツ間での当パラメータの引き回しは不要です。 連続処理用の情報は「imwCallOriginalParams」に内包されます。
17	imwCallOriginalParams	呼出元パラメータ	呼出元ページのパラメータ
18	imwCallOriginalPagePath	呼出元ページパス	呼出元のページパス
19	imwSysDateTargetExpandFlag	システム日で対象者を展開するフラグ	“0”：無効、“1”：有効
20	imwShortCutFlag	ショートカットフラグ	ショートカットアクセスURLから詳細画面が表示されたことを示すフラグです。 ※IM-Workflow バージョン8.0.10より追加されました。 “0”：ショートカットアクセスURLからの遷移ではない “1”：ショートカットアクセスURLからの遷移

※imwAuthUserCode（権限者CD）について、各開発モデルでの取得例は以下の通りです。

ここで記載している内容は、次の観点において共通です。

- クライアントタイプ

スクリプト開発モデル

```
1 function init(request) {
2   var imwAuthUserCodeList = request.getParameterValues("imwAuthUserCode"); //権限者CDの配列
3 }
```

JavaEE開発モデル

```
1 HttpServletRequest request = getRequest();
2 String[] imwAuthUserCodeList = request.getParameterValues("imwAuthUserCode"); //権限者CDの配列
```

画面別取得可否一覧

No	パラメータ	申請 ※	一時 保存※	起票 ※	再申請 ※	処理 ※	確認 ※	処理 詳細	参照 詳細	確認 詳細	過去 案件 詳細
1	imwGroupId	○	○	○	○	○	○	○	○	○	○
2	imwUserCode	○	○	○	○	○	○	○	○	○	○
3	imwPageType	○	○	○	○	○	○	○	○	○	○
4	imwUserDataId	-	○	○	○	○	○	○	○	○	○
5	imwSystemMatterId	-	-	○	○	○	○	○	○	○	○
6	imwNodeId	○	○	○	○	○	○	-	-	-	-
7	imwArriveType	○	○	○	○	○	-	-	-	-	-
8	imwAuthUserCode	○	○	○	○	○	-	-	-	-	-
9	imwApplyBaseDate	○	○	○	○	○	○	○	○	○	○
10	imwFlowId	○	○	○	○	○	○	○	○	○	○
11	imwFlowVersionId	○	○	○	○	○	○	○	○	○	○
12	imwContentsId	○	○	○	○	○	○	○	○	○	○
13	imwContentsVersionId	○	○	○	○	○	○	○	○	○	○
14	imwRouteId	○	○	○	○	○	○	○	○	○	○
15	imwRouteVersionId	○	○	○	○	○	○	○	○	○	○
16	imwSerialProcParams	-	-	○-	○-	○-	○-	-	-	-	-
17	imwCallOriginalParams	○	○	○	○	○	○	-	-	-	-
18	imwCallOriginalPagePath	○	○	○	○	○	○	-	-	-	-
19	imwSysDateTargetExpandFalg	○	○	○	○	○	○	○	○	○	-
20	imwShortCutFlag	-	-	-	-	-	-	○	○	○	○

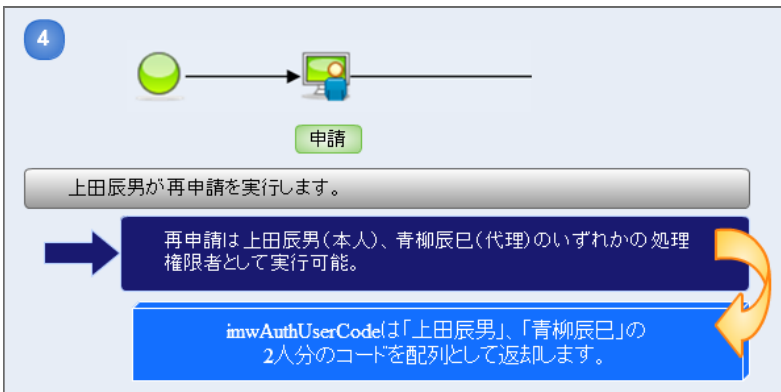
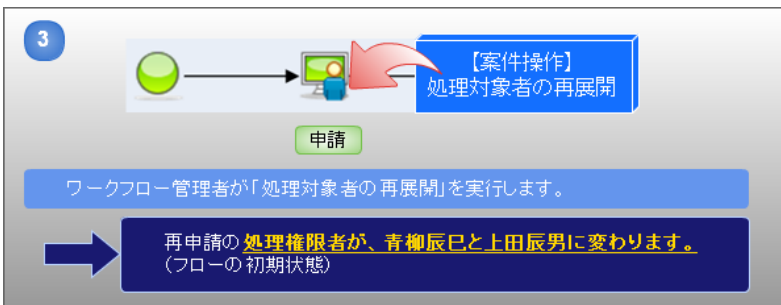
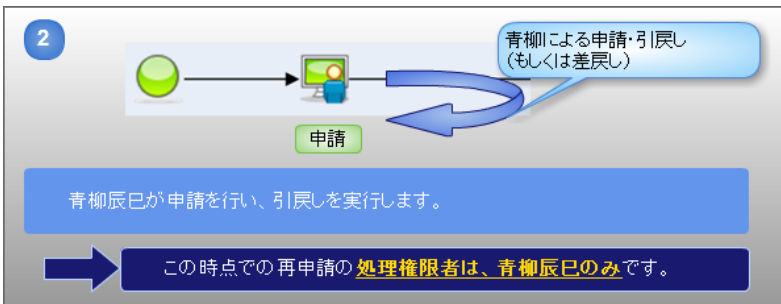
< 「○」：取得可能 / 「-」：取得不可能 >

※スマートフォン用の場合も同様です。

ただし、「imwAuthUserCode」のみ、スマートフォン用の起票、再申請、処理画面では取得することができません。

- 再申請時の「imwAuthUserCode」

リクエストパラメータの「imwAuthUserCode」は、基本的には特定の1ユーザコードを返却しますが、以下の図の例のような操作を行った場合には、複数のユーザコードを返却します。



案件処理系APIと画面動作仕様の違い

画面上からの操作とは異なり、案件処理系API (Webサービスを含む) を直接利用して案件の処理を行う場合は、業務的なチェックを行わずに処理が実行されます。ここでいう業務的なチェックとは、以下のようなチェックを指します。

- API引数として指定した処理権限者が、到達処理で展開された処理対象者に含まれるか
- API引数として指定した処理権限者と処理実行者が異なる場合、両者間で有効な代理設定が存在するか

画面上からの操作と同等の機能を、APIを利用して独自に実装する場合は、案件処理APIの実行前に各種API (処理権限判定APIなど) を併せて利用してください。

この章では、IM-Workflow が提供する案件の各処理画面と連携するための画面実装の基本部分について、「画面種別」、「開発モデル」、「クライアントタイプ」の観点で説明します。

上記観点の内訳は、次のとおりです。

- 画面種別
 - 申請画面
 - 一時保存画面
 - 申請（起票案件）画面
 - 再申請画面
 - 処理画面
 - 確認画面
 - 処理詳細
 - 参照詳細
 - 確認詳細
 - 過去案件詳細
- 開発モデル
 - スクリプト開発モデル
 - （「パス種別：スクリプト開発モデル」で利用するユーザコンテンツ画面）
 - JavaEE開発モデル
 - （「パス種別：JavaEE開発モデル」で利用するユーザコンテンツ画面）
 - JSP、TERASOLUNA Server Framework for Java (5.x)、IM-BloomMaker
 - （「パス種別：URL(旧名称:JSP or Servlet)」で利用するユーザコンテンツ画面）
- クライアントタイプ
 - PC
 - スマートフォン

また、画面作成における応用実装について、「[カスタマイズ](#)」で説明しています。
必要に応じて参照してください。



注意

2020 Summer(Zephyrine)以降、「[画面の作成（処理モジュール）](#)」に記載されているワークフロー処理モジュール、および、「[クライアントサイド JavaScript API](#)」を利用した画面実装を推奨します。
尚、この章に記載の「workflowOpenPage」タグは、これまで通り動作しますが、今後機能追加の予定はありません。
ワークフロー処理モジュール、および、「[クライアントサイド JavaScript API](#)」に対して機能追加されます。

申請画面の呼び出し

IM-Workflow で提供する申請を行うための画面（以下、「申請」画面）と連携する方法を説明します。

Contents

- [スクリプト開発モデル](#)
- [JavaEE開発モデル](#)
- [JSP、TERASOLUNA Server Framework for Java \(5.x\)](#)



「申請」画面を表示するためには、IM-Workflow が提供するタグライブラリおよびClient-side JavaScript APIを使用します。

スクリプト開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「申請」画面と連携する画面のヘッダ部 (<imart type="head"> ~ </imart>) に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="workflowOpenPageCsjs" />
4
5 </imart>
    
```

「申請」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

「imwUserDataId」は、「申請一覧」画面からのリクエストパラメータには含まれません。

ファンクション・コンテナで採番する必要があります。

```

1 <imart type="workflowOpenPage"
2   name="applyForm"
3   id="applyForm"
4   method="POST"
5   target="_top"
6   imwUserDataId=oRequest.imwUserDataId
7   imwAuthUserCode=oRequest.imwAuthUserCode
8   imwApplyBaseDate=oRequest.imwApplyBaseDate
9   imwNodeId=oRequest.imwNodeId
10  imwFlowId=oRequest.imwFlowId>
11 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「申請」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage('0');
4
5 </script>

```

スマートフォン用画面の場合

「申請」画面と連携する画面のヘッダ部（<imart type="head"> ~ </imart>）に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="spWorkflowOpenPageCsjs" />
4
5 </imart>

```

「申請」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

「imwUserDataId」は、「申請一覧」画面からのリクエストパラメータには含まれません。

ファンクション・コンテナで採番する必要があります。

```

1 <imart type="spWorkflowOpenPage"
2   name="applyForm"
3   id="workflowOpenPageForm"
4   method="POST"
5   target="_top"
6   imwUserDataId=$data.imwUserDataId
7   imwAuthUserCode=$data.imwAuthUserCode
8   imwApplyBaseDate=$data.imwApplyBaseDate
9   imwNodeId=$data.imwNodeId
10  imwFlowId=$data.imwFlowId>
11 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「申請」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('10');
4
5 </script>

```

JavaEE開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「申請」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:workflowOpenPageCsjs />
4
5 </imui:head>

```

「申請」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

「imwUserDataId」は、「申請一覧」画面からのリクエストパラメータには含まれません。

ServiceControllerなどで採番する必要があります。

```

1 <workflow:workflowOpenPage
2   name="applyForm"
3   id="applyForm"
4   method="POST"
5   target="_top"
6   imwUserDataId='<%= (String)request.getAttribute("imwUserDataId")%>'
7   imwAuthUserCode='<%= (String)request.getAttribute("imwAuthUserCode")%>'
8   imwApplyBaseDate='<%= (String)request.getAttribute("imwApplyBaseDate")%>'
9   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
10  imwFlowId='<%= (String)request.getAttribute("imwFlowId")%>'
11 </workflow:workflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「申請」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage('0');
4
5 </script>

```

スマートフォン用画面の場合

「申請」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:spWorkflowOpenPageCsjs />
4
5 </imui:head>

```

「申請」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

「imwUserDataId」は、「申請一覧」画面からのリクエストパラメータには含まれません。

ServiceControllerなどで採番する必要があります。

```

1 <workflow:spWorkflowOpenPage
2   name="applyForm"
3   id="applyForm"
4   method="POST"
5   target="_top"
6   imwUserDataId='<%= (String)request.getAttribute("imwUserDataId")%>'
7   imwAuthUserCode='<%= (String)request.getAttribute("imwAuthUserCode")%>'
8   imwApplyBaseDate='<%= (String)request.getAttribute("imwApplyBaseDate")%>'
9   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
10  imwFlowId='<%= (String)request.getAttribute("imwFlowId")%>'
11 </ workflow:spWorkflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「申請」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('10');
4
5 </script>

```

JSP、TERASOLUNA Server Framework for Java (5.x)

JavaEE開発モデルに準じます。

TERASOLUNA Server Framework for Java (5.x) を利用して実装する場合には、「IM-Workflow TERASOLUNA Server Framework プログラミングガイド」も併せて参照してください。

一時保存画面の呼び出し

IM-Workflow で提供する一時保存を行うための画面（以下、「一時保存」画面）と連携する方法を説明します。

Contents

- スクリプト開発モデル
- JavaEE開発モデル
- JSP、TERASOLUNA Server Framework for Java (5.x)



「一時保存」画面を表示するためには、IM-Workflow が提供するタグライブラリおよびClient-side JavaScript APIを使用します。

スクリプト開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「APIドキュメント」も併せて参照してください。

PC用画面の場合

「一時保存」画面と連携する画面のヘッダ部（<imart type="head"> ~ </imart>）に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="workflowOpenPageCsjs" />
4
5 </imart>
    
```

「一時保存」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="workflowOpenPage"
2   name="tempForm"
3   id="tempForm"
4   method="POST"
5   target="_top"
6   imwUserDataId=oRequest.imwUserDataId
7   imwAuthUserCode=oRequest.imwAuthUserCode
8   imwApplyBaseDate=oRequest.imwApplyBaseDate
9   imwNodeId=oRequest.imwNodeId
10  imwFlowId=oRequest.imwFlowId>
11 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「一時保存」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage('1');
4
5 </script>

```

スマートフォン用画面の場合

「一時保存」画面と連携する画面のヘッダ部（<imart type="head"> ~ </imart>）に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="spWorkflowOpenPageCsjs" />
4
5 </imart>

```

「一時保存」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="spWorkflowOpenPage"
2   name="tempForm"
3   id="tempForm"
4   method="POST"
5   target="_top"
6   imwUserDataId=$data.imwUserDataId
7   imwAuthUserCode=$data.imwAuthUserCode
8   imwApplyBaseDate=$data.imwApplyBaseDate
9   imwNodeId=$data.imwNodeId
10  imwFlowId=$data.imwFlowId>
11 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「一時保存」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('11');
4
5 </script>

```

JavaEE開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「一時保存」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:workflowOpenPageCsjs />
4
5 </imui:head>

```

「一時保存」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

```

1 <workflow:workflowOpenPage
2   name="tempForm"
3   id="tempForm"
4   method="POST"
5   target="_top"
6   imwUserDataId='<%= (String)request.getAttribute("imwUserDataId")%>'
7   imwAuthUserCode='<%= (String)request.getAttribute("imwAuthUserCode")%>'
8   imwApplyBaseDate='<%= (String)request.getAttribute("imwApplyBaseDate")%>'
9   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
10  imwFlowId='<%= (String)request.getAttribute("imwFlowId")%>'
11 </workflow:workflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「一時保存」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage('1');
4
5 </script>

```

スマートフォン用画面の場合

「一時保存」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:spWorkflowOpenPageCsjs />
4
5 </imui:head>

```

「一時保存」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「申請一覧」画面から取得したリクエストパラメータを指定します。

```

1 <workflow:spWorkflowOpenPage
2   name="tempForm"
3   id="tempForm"
4   method="POST"
5   target="_top"
6   imwUserDataId='<%= (String)request.getAttribute("imwUserDataId")%>'
7   imwAuthUserCode='<%= (String)request.getAttribute("imwAuthUserCode")%>'
8   imwApplyBaseDate='<%= (String)request.getAttribute("imwApplyBaseDate")%>'
9   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
10  imwFlowId='<%= (String)request.getAttribute("imwFlowId")%>'
11 </ workflow:spWorkflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「一時保存」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('11');
4
5 </script>

```

JSP、TERASOLUNA Server Framework for Java (5.x)

JavaEE開発モデルに準じます。

TERASOLUNA Server Framework for Java (5.x) を利用して実装する場合には、「[IM-Workflow TERASOLUNA Server Framework プログラミングガイド](#)」も併せて参照してください。

IM-Workflow で提供する申請（起票案件）／再申請／処理を行うための画面（以下、「処理」画面）と連携する方法を説明します。

Contents

- スクリプト開発モデル
- JavaEE開発モデル
- JSP、TERASOLUNA Server Framework for Java (5.x)



「処理」画面を表示するためには、IM-Workflow が提供するタグライブラリおよびClient-side JavaScript APIを使用します。

スクリプト開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「APIドキュメント」も併せて参照してください。

PC用画面の場合

「処理」画面と連携する画面のヘッダ部 (<imart type="head"> ~ </imart>) に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="workflowOpenPageCsjs" />
4
5 </imart>

```

「処理」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「未処理一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="workflowOpenPage"
2   name="approveForm"
3   id="approveForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId=$data.imwSystemMatterId
7   imwNodeId=$data.imwNodeId >
8 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「処理」画面が表示されます。

申請（起票案件）

```

1 <script type="text/javascript">
2
3   workflowOpenPage('2');
4
5 </script>

```

再申請

```

1 <script type="text/javascript">
2
3   workflowOpenPage('3');
4
5 </script>

```

処理

```

1 <script type="text/javascript">
2
3   workflowOpenPage('4');
4
5 </script>

```

スマートフォン用画面の場合

「処理」画面と連携する画面のヘッダ部（<imart type="head"> ~ </imart>）に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="spWorkflowOpenPageCsjs" />
4
5 </imart>

```

「処理」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「未処理一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="spWorkflowOpenPage"
2   name="approveForm"
3   id="approveForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId=$data.imwSystemMatterId
7   imwNodeId=$data.imwNodeId >
8 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「処理」画面が表示されます。

申請（起票案件）

```

1 <script type="text/javascript">
2   workflowOpenPage4Sp('12');
3
4 </script>
5

```

再申請

```

1 <script type="text/javascript">
2   workflowOpenPage4Sp('13');
3
4 </script>
5

```

処理

```

1 <script type="text/javascript">
2   workflowOpenPage4Sp('14');
3
4 </script>
5

```

JavaEE開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「処理」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2   <workflow:workflowOpenPageCsjs />
3
4 </imui:head>
5

```

「処理」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「未処理一覧」画面から取得したリクエストパラメータを指定します。

```

1 <workflow:workflowOpenPage
2   name="approveForm"
3   id="approveForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId='<%= (String)request.getAttribute("imwSystemMatterId")%>'
7   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
8 </workflow:workflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「処理」画面が表示されます。

申請（起票案件）

```

1 <script type="text/javascript">
2   workflowOpenPage('2');
3
4 </script>
5

```

再申請

```

1 <script type="text/javascript">
2   workflowOpenPage('3');
3
4 </script>
5

```

処理

```

1 <script type="text/javascript">
2
3   workflowOpenPage('4');
4
5 </script>

```

スマートフォン用画面の場合

「処理」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:spWorkflowOpenPageCsjs />
4
5 </imui:head>

```

「処理」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「未処理一覧」画面から取得したリクエストパラメータを指定します。

```

1 <workflow:spWorkflowOpenPage
2   name="approveForm"
3   id="approveForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId='<%= (String)request.getAttribute("imwSystemMatterId")%>'
7   imwNodeId='<%= (String)request.getAttribute("imwNodeId")%>'
8 </workflow:spWorkflowOpenPage >

```

下記のClient-side JavaScript APIを実行することにより、「処理」画面が表示されます。

申請（起票案件）

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('12');
4
5 </script>

```

再申請

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('13');
4
5 </script>

```

処理

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('14');
4
5 </script>

```

JSP、TERASOLUNA Server Framework for Java (5.x)

JavaEE開発モデルに準じます。

TERASOLUNA Server Framework for Java (5.x) を利用して実装する場合には、「IM-Workflow TERASOLUNA Server Framework プログラミングガイド」も併せて参照してください。

確認画面の呼び出し

IM-Workflow で提供する確認を行うための画面（以下、「確認」画面）と連携する方法を説明します。

Contents

- スクリプト開発モデル
- JavaEE開発モデル
- JSP、TERASOLUNA Server Framework for Java (5.x)



「確認」画面を表示するためには、IM-Workflow が提供するタグライブラリおよびClient-side JavaScript APIを使用します。

スクリプト開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「確認」画面と連携する画面のヘッダ部 (<imart type="head"> ~ </imart>) に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="workflowOpenPageCsjs" />
4
5 </imart>

```

「確認」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「確認一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="workflowOpenPage"
2   name="confirmForm"
3   id="confirmForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId=$data.imwSystemMatterId
7   imwNodeid=$data.imwNodeid>
8 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「確認」画面が表示されます。

```

1 <script type="text/javascript">
2   workflowOpenPage('5');
3
4 </script>
5

```

スマートフォン用画面の場合

「確認」画面と連携する画面のヘッダ部（<imart type="head"> ~ </imart>）に、下記のIMARTタグを記述します。

```

1 <imart type="head">
2
3 <imart type="spWorkflowOpenPageCsjs" />
4
5 </imart>

```

「確認」画面と連携する画面のボディ部に、下記のIMARTタグを記述します。

IMARTタグに指定する属性は、通常「確認一覧」画面から取得したリクエストパラメータを指定します。

```

1 <imart type="spWorkflowOpenPage"
2   name="confirmForm"
3   id="confirmForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId=$data.imwSystemMatterId
7   imwNodeId=$data.imwNodeId>
8 </imart>

```

下記のClient-side JavaScript APIを実行することにより、「確認」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage4Sp('15');
4
5 </script>

```

JavaEE開発モデル

IM-Workflow 用のタグライブラリの使用方法については、「[APIドキュメント](#)」も併せて参照してください。

PC用画面の場合

「確認」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```

1 <imui:head>
2
3 <workflow:workflowOpenPageCsjs />
4
5 </imui:head>

```

「確認」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「確認一覧」画面から取得したリクエストパラメータを指定します。

```

1 <workflow:workflowOpenPage
2   name="confirmForm"
3   id="confirmForm"
4   method="POST"
5   target="_top"
6   imwSystemMatterId='<%= (String) request.getAttribute("imwSystemMatterId") %>'
7   imwNodeId='<%= (String) request.getAttribute("imwNodeId") %>'>
8 </workflow:workflowOpenPage>

```

下記のClient-side JavaScript APIを実行することにより、「確認」画面が表示されます。

```

1 <script type="text/javascript">
2
3   workflowOpenPage('5');
4
5 </script>

```

スマートフォン用画面の場合

「確認」画面と連携する画面のヘッダ部（<imui:head> ~ </imui:head>）に、下記のタグライブラリを記述します。

```
1 <imui:head>
2
3 <workflow:spWorkflowOpenPageCsjs />
4
5 </imui:head>
```

「確認」画面と連携する画面のボディ部に、下記のタグライブラリを記述します。

タグライブラリに指定する属性は、通常「確認一覧」画面から取得したリクエストパラメータを指定します。

```
1 <workflow: spWorkflowOpenPage
2     name="confirmForm"
3     id="confirmForm"
4     method="POST"
5     target="_top"
6     imwSystemMatterId='<%= (String)request.getAttribute("imwSystemMatterId")%>'
7     imwNodeid='<%= (String)request.getAttribute("imwNodeid")%>'
8 </workflow:spWorkflowOpenPage>
```

下記のClient-side JavaScript APIを実行することにより、「確認」画面が表示されます。

```
1 <script type="text/javascript">
2
3     workflowOpenPage4Sp('15');
4
5 </script>
```

JSP、TERASOLUNA Server Framework for Java (5.x)

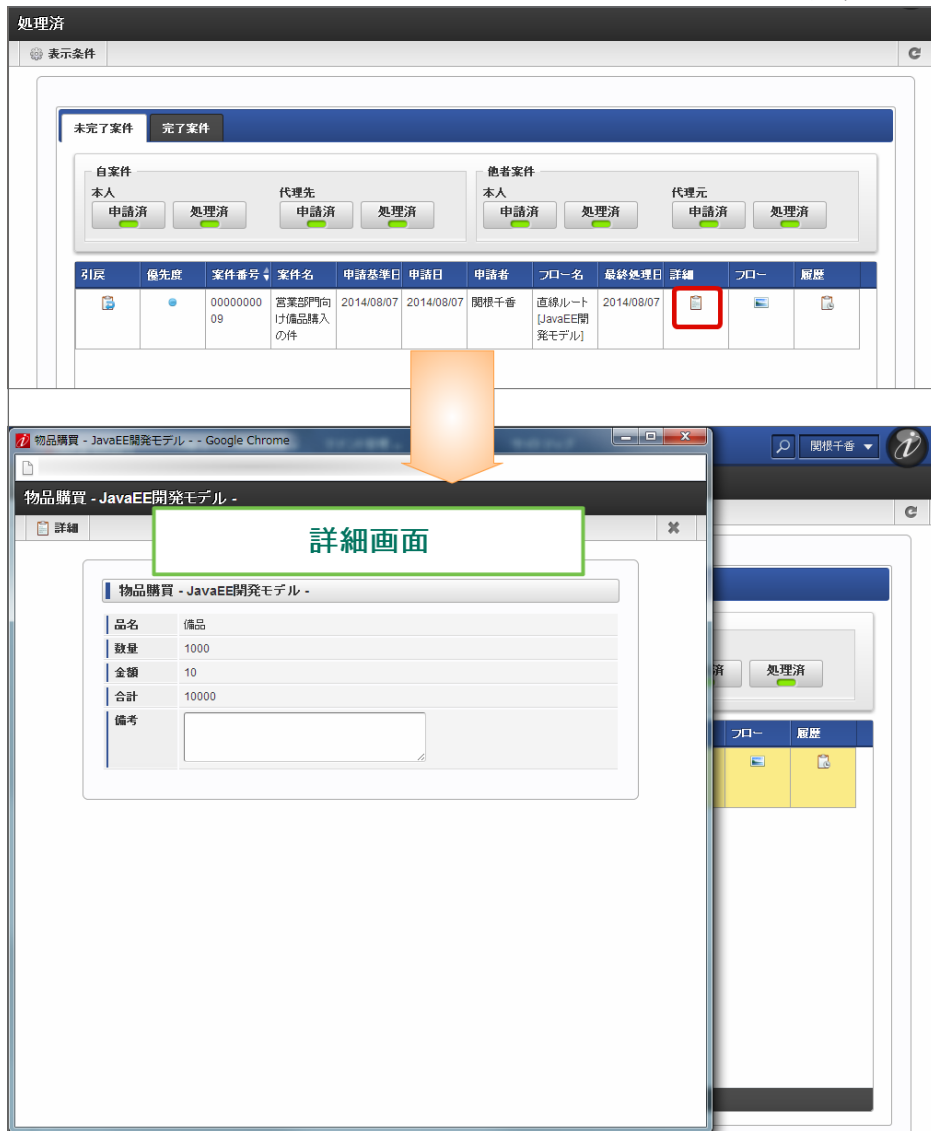
JavaEE開発モデルに準じます。

TERASOLUNA Server Framework for Java (5.x) を利用して実装する場合には、「[IM-Workflow TERASOLUNA Server Framework プログラミングガイド](#)」も併せて参照してください。

詳細画面の呼び出し

詳細画面は、他の処理画面と異なり、IM-Workflow が提供する標準処理画面との連携を行わないため、タグライブラリ「workflowOpenPageCsjs」および「workflowOpenPage」を実装する必要はありません。

呼び出し時の特徴として、詳細画面はポップアップで表示されます。



通常は、ポップアップ表示された詳細画面にも、呼び出し元画面と同様のヘッダ・フッタが表示されます。ヘッダが表示されていると、別画面へのメニュー遷移や、ログアウト処理などが可能です。これらの処理がポップアップ画面で行われた場合、呼び出し元画面での動作が不安定となる可能性があります。そのため、詳細画面では、特別な理由がない限り、テーマのヘッダ・フッタを非表示とする対応をとることを推奨します。

テーマのヘッダ・フッタを非表示するには、詳細画面のパスに対し、ページビルダー `HeadOnlyThemeBuilder` を適用します。

- 設定場所：

```
<WEB-INF/conf/theme-head-only-path-config/*.xml>
```

- 設定方法の詳細については、「[UIデザインガイドライン \(PC版\)](#)」を参照してください。

この章では、IM-Workflow が提供するワークフロー処理モジュールと連携するための画面実装の基本部分について説明します。

また、モジュールとの連携を含め、クライアントサイドから IM-Workflow の処理を行うための JavaScript API を提供しています。APIの詳細は、下記の「[クライアントサイドJavaScript API](#)」を参照してください。

ワークフロー処理モジュールの呼び出し

ワークフロー処理モジュールの呼び出し方法について説明します。

申請モジュール

「申請モジュール」を呼び出す方法について説明します。

「申請モジュール」は [クライアントサイドJavaScript API](#) を利用して呼び出します。

「申請」画面と連携する画面のヘッダ部に、下記のとおりimSecureTokenの定義と<im_workflow/js/api_base.js>の読み込みを記述します。

```
<imart type="head">
<meta http-equiv="X-Intramart-Secure-Token" content="<imart type="imSecureToken" mode="value" />"/>
<script src="im_workflow/js/api_base.js" defer></script>
</imart>
```

下記のような JavaScript を実行することで、「申請モジュール」が表示されます。詳細は [showApply](#) を参照してください。

```
1 <script type="text/javascript">
2 imWorkflow.modal.showApply({
3   processParameter: {
4     matterName: "sampleMatter",
5     priorityLevel: "9",
6     interfaceControl: {
7       priorityLevel: {
8         readonly: true
9       },
10      nego: {
11        display: false
12      }
13    }
14  },
15  optionalParameter: {
16    userParameter: {
17      item_name: $('#item_name').val(),
18      item_amount: $('#item_amount').val(),
19      item_price: $('#item_price').val(),
20      item_comment: $('#item_comment').val()
21    }
22  },
23  rebootModal: false
24 })
25 .then(function(result) {
26   if (result.isProcessDone) {
27     alert("done");
28   }
29   imWorkflow.transition.afterProcess();
30 });
31 </script>
```

また、サンプルデータとして「申請モーダル」の呼び出しを実装した、以下のスクリプト開発のファイルを提供しています。

```
<./jssp/src/sample/im_workflow/purchase/screen_modal/apply.html>
<./jssp/src/sample/im_workflow/purchase/screen_modal/apply.js>
```

コラム

このAPIはPromiseを返します。
また、ワークフロー処理モーダルが閉じたタイミングで処理を終了します。
ワークフロー処理モーダルが閉じるタイミングは以下のとおりです。

- 閉じるボタンクリック
- ESCキー押下
- 処理が正常終了

よって、モーダルが閉じた後の処理は、then メソッド内に記述してください。
また、申請後の処理は、返却値のisProcessDoneプロパティがtrueの場合の処理内で記述してください。

コラム

<im_workflow/js/api_base.js>を読み込む際はdefer属性を付与する必要があります。

一時保存モーダル

「一時保存モーダル」を呼び出す方法について説明します。

「一時保存モーダル」は [クライアントサイドJavaScript API](#) を利用して呼び出します。

「一時保存」画面と連携する画面のヘッダ部に、下記のとおりimSecureTokenの定義と<im_workflow/js/api_base.js>の読み込みを記述します。

```
<imart type="head">
<meta http-equiv="X-Intramart-Secure-Token" content="<imart type="imSecureToken" mode="value" />"/>
<script src="im_workflow/js/api_base.js" defer></script>
</imart>
```

下記のような JavaScript を実行することで、「一時保存モーダル」が表示されます。

詳細は [showTemporarySave](#) を参照してください。

```
1 <script type="text/javascript">
2 imWorkflow.modal.showTemporarySave({
3   processParameter: {
4     matterName: "sampleMatter"
5   },
6   optionalParameter: {
7     userParameter: {
8       item_name: $('#item_name').val(),
9       item_amount: $('#item_amount').val(),
10      item_price: $('#item_price').val(),
11      item_comment: $('#item_comment').val()
12    }
13  },
14  rebootModal: false
15 })
16 .then(function(result) {
17   if (result.isProcessDone) {
18     alert("done");
19   }
20   imWorkflow.transition.afterProcess();
21 });
22 </script>
```

また、サンプルデータとして「一時保存モーダル」の呼び出しを実装した、以下のスクリプト開発のファイルを提供しています。

```
<./jssp/src/sample/im_workflow/purchase/screen_modal/apply.html>
<./jssp/src/sample/im_workflow/purchase/screen_modal/apply.js>
```

i コラム

このAPIはPromiseを返します。
また、ワークフロー処理モダルが閉じたタイミングで処理を終了します。
ワークフロー処理モダルが閉じるタイミングは以下のとおりです。

- 閉じるボタンクリック
- ESCキー押下
- 処理が正常終了

よって、モダルが閉じた後の処理は、then メソッド内に記述してください。
また、一時保存後の処理は、返却値のisProcessDoneプロパティがtrueの場合の処理内で記述してください。

i コラム

<im_workflow/js/api_base.js>を読み込む際はdefer属性を付与する必要があります。

処理モダル

「処理モダル」を呼び出す方法について説明します。

「処理モダル」は [クライアントサイドJavaScript API](#) を利用して呼び出します。

「処理」画面と連携する画面のヘッダ部に、下記のとおりimSecureTokenの定義と<im_workflow/js/api_base.js>の読み込みを記述します。

```
<imart type="head">
<meta http-equiv="X-Intramart-Secure-Token" content="<imart type="imSecureToken" mode="value" />" />
<script src="im_workflow/js/api_base.js" defer></script>
</imart>
```

下記のようなJavaScript を実行することで、「処理モダル」が表示されます。
詳細は [showProcess](#) を参照してください。

```
1 <script type="text/javascript">
2   imWorkflow.modal.showProcess({
3     processParameter: {
4       interfaceControl: {
5         processComment: {
6           display: false
7         }
8       }
9     },
10    optionalParameter: {},
11    rebootModal: false
12  })
13  .then(function(result) {
14    if (result.isProcessDone) {
15      alert("done");
16    }
17    imWorkflow.transition.afterProcess();
18  });
19 </script>
```

また、サンプルデータとして「処理モダル」の呼び出しを実装した、以下のスクリプト開発のファイルを提供しています。

```
<./jssp/src/sample/im_workflow/purchase/screen_modal/approve.html>
<./jssp/src/sample/im_workflow/purchase/screen_modal/approve.js>
```

i コラム

このAPIはPromiseを返します。
また、ワークフロー処理モダルが閉じたタイミングで処理を終了します。
ワークフロー処理モダルが閉じるタイミングは以下のとおりです。

- 閉じるボタンクリック
- ESCキー押下
- 処理が正常終了

よって、モダルが閉じた後の処理は、then メソッド内に記述してください。
また、処理後の処理は、返却値のisProcessDoneプロパティがtrueの場合の処理内で記述してください。

i コラム

<im_workflow/js/api_base.js>を読み込む際はdefer属性を付与する必要があります。

確認モダル

「確認モダル」を呼び出す方法について説明します。

「確認モダル」は [クライアントサイドJavaScript API](#) を利用して呼び出します。

「確認」画面と連携する画面のヘッダ部に、下記のとおりimSecureTokenの定義と<im_workflow/js/api_base.js>の読み込みを記述します。

```
<imart type="head">
<meta http-equiv="X-Intramart-Secure-Token" content="<imart type="imSecureToken" mode="value" />" />
<script src="im_workflow/js/api_base.js" defer></script>
</imart>
```

下記のようなJavaScript を実行することで、「確認モダル」が表示されます。
詳細は [showConfirm](#) を参照してください。

```
1 <script type="text/javascript">
2 imWorkflow.modal.showConfirm({
3   processParameter: {
4     confirmComment: "confirm",
5     interfaceControl: {
6       confirmComment: {
7         readonly: true
8       }
9     }
10  },
11  rebootModal: false
12 })
13 .then(function(result) {
14   if (result.isProcessDone) {
15     alert("done");
16   }
17   imWorkflow.transition.afterProcess();
18 });
19 </script>
```

また、サンプルデータとして「確認モダル」の呼び出しを実装した、以下のスクリプト開発のファイルを提供しています。

```
<./jssp/src/sample/im_workflow/purchase/screen_modal/confirm.html>
<./jssp/src/sample/im_workflow/purchase/screen_modal/confirm.js>
```

i コラム

このAPIはPromiseを返します。
また、ワークフロー処理モジュールが閉じたタイミングで処理を終了します。
ワークフロー処理モジュールが閉じるタイミングは以下のとおりです。

- 閉じるボタンクリック
- ESCキー押下
- 処理が正常終了

よって、モジュールが閉じた後の処理は、then メソッド内に記述してください。
また、確認後の処理は、返却値のisProcessDoneプロパティがtrueの場合の処理内で記述してください。

i コラム

<im_workflow/js/api_base.js>を読み込む際はdefer属性を付与する必要があります。

クライアントサイド JavaScript API

imWorkflow

IM-Workflow のクライアントサイドJavascriptAPIです。

このAPIを使用するためには、<api_base.js>ファイルをdefer属性を付与して読み込む必要があります。
script タグを利用した js ファイルの読み込み方法については、以下のとおりです。

```
<script src="im_workflow/js/api_base.js" defer></script>
```

各機能については、以下のとおりです。

transition

画面遷移処理オブジェクトです。
IM-Workflow の画面遷移に関する処理を行います。

returnTo

imWorkflow.transition.returnTo()

ユーザコンテンツの呼び出し元画面に戻ります。
「申請一覧」画面や「案件一覧」画面、各種ワークフローポートレットから指定した「imwCallOriginalPagePath」に遷移します。

Contents

- サンプルコード

サンプルコード

```
imWorkflow.transition.returnTo();
```

afterProcess

imWorkflow.transition.afterProcess()

案件処理後の画面遷移を行います。

このライブラリのメソッドによる案件の処理が実行される前に、当メソッドを実行した場合は何もしません。
連続処理、または、連続確認中の場合は、次の案件のユーザコンテンツ画面に遷移します。
それ以外の場合は、ユーザコンテンツの呼び出し元画面に遷移します。
また、このメソッドは *process*、または、*modal* を利用した際の then メソッド内で利用してください。

Contents

- サンプルコード

サンプルコード

```
imWorkflow.modal.showApply()
.then(function(result) {
  imWorkflow.transition.afterProcess();
});
```

process

案件処理オブジェクトです。

IM-Workflow の案件に対する「処理」を実行します。

このオブジェクトの各メソッドを実行する場合は、予めセキュアトークンをブラウザで保持しておく必要があります。

セキュアトークンは、head タグの中で以下の様に meta タグを記述して imSecureToken の出力値を保持してください。

```
<meta http-equiv="X-Intramart-Secure-Token" content='<imart type="imSecureToken" mode="value" />' />
```

コラム

imSecureToken の詳細は、以下を参照してください。

- スクリプト開発の場合
 - 「スクリプト開発向けタグライブラリ/画面共通」 - 「imSecureTokenタグ」
- JavaEE開発の場合
 - 「JavaEE開発向けタグライブラリ/intra-mart標準」 - 「imSecureTokenタグ」

apply

imWorkflow.process.apply(Object parameter) : Promise<Object>

ワークフローを申請します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs
 - confirmNodeConfigs
 - horizontalNodeConfigs
 - verticalNodeConfigs
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.apply({
  processParameter: {
    flowId: "",
    userDataId: "",
    matterName: "",
    applyBaseDate: "",
    applyAuthUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    priorityLevel: "",
    temporaryDirectoryKey: "",
    branchSelects: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    nego: {}
  },
  optionalParameter: {}
})
.then(function(result) {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
申請用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

項目名	必須/任意	型 (最大バイト数)	説明
flowId	任意	文字列	フローID 「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択したフローIDが設定されるため不要 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、必須
userDataId	任意	文字列 (20)	ユーザデータID 未指定の場合、ワークフローパラメータ「 ユーザデータID設定方式 」の設定によりIDを決定する。 一時保存状態から申請する場合、一時保存情報のユーザデータIDを指定
matterName	任意	文字列 (200)	案件名 未指定の場合、以下の優先度で決定して申請 1.一時保存時に指定した案件名 (一時保存からの申請の場合) 2.フロー名
applyBaseDate	任意	文字列	申請基準日 未指定の場合、「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択した申請基準日が設定される 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、アカウントコンテキストのタイムゾーンにおける現在日付で申請 “yyyy/MM/dd”形式の文字列で指定
applyAuthUserCd	任意	文字列	申請権限者コード 未指定の場合、以下の優先度で決定して申請 1.「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択した申請者 2.実行ユーザ
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で申請 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
priorityLevel	任意	文字列	優先度 設定値は「 CodeList 」を参照 未指定の場合、「通常」で申請
temporaryDirectoryKey	任意	文字列	添付ファイル一時領域ディレクトリキー
branchSelects	任意	下記参照	分岐先選択情報 分岐先設定可能ノードの場合指定必須
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、削除可能ノードであればノードを削除、削除不可の場合はエラー
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 確認対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
nego	任意	下記参照	根回し情報

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。
forwardNodeIds	必須	配列（文字列）	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs

ノードを削除する場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  dynamicNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

confirmNodeConfigs

確認対象者なしとする場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

horizontalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。 詳細は「 動的処理対象者設定機能 」を参照してください。

verticalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。

項目名	必須/任意	型 (最大バイト数)	説明
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「 動的処理対象者設定機能 」を参照してください。



コラム

ノードの削除可否や割当可能ノード数、利用可能なプラグインは、[動的承認](#)・[横配置](#)・[縦配置ノード設定](#) の範囲で指定できます。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。

コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合はpromise.then()で返却値を受け取ってください。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
matterNumber	文字列	案件番号 処理が成功した場合に設定されます。
systemMatterId	文字列	システム案件ID 処理が成功した場合に設定されます。
userDataId	文字列	ユーザデータID 処理が成功した場合に設定されます。
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
0003	指定したフロー、申請基準日、権限者に対する申請権限がありません。	
0004	申請権限のチェックに失敗しました。管理者に連絡してください。	
0005	申請可能な組織権限がありません。	
0006	組織権限情報の取得に失敗しました。管理者に連絡してください。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
1005	申請情報が未指定です。	
1015	フローIDが未指定です。	
1018	指定できる分岐開始ノードが存在しません。	
1019	分岐先ノードが未指定です。	
1020	設定が必要な分岐先ノード数と指定された分岐先ノード数が異なります。	
1021	分岐先ノードの入力情報に不備があります。	
1022	同一の分岐先ノードに対して指定が重複しています。	
1023	選択可能なルートは単一であるのに対し分岐先ノードが複数指定されています。	
1024	指定できない分岐先ノードが指定されています。	
1025	コメントが許可された最大文字数を超過しています。	

エラーコード	エラーメッセージ	備考
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1027	指定できる確認ノードが存在しません。	
1028	設定可能なノード数以上の確認ノードが指定されています。	
1029	確認ノードの入力情報に不備があります。	
1030	同一の確認ノードに対して指定が重複しています。	
1031	指定できない確認ノードが指定されています。	
1032	指定できる動的承認ノードが存在しません。	
1033	動的承認ノードが未指定です。	
1034	設定が可能な動的承認ノード数と指定された動的承認ノード数が異なります。	
1035	動的承認ノードの入力情報に不備があります。	
1036	同一の動的承認ノードに対して指定が重複しています。	
1037	削除できない動的承認ノードに対して処理対象者の指定がされていません。	
1038	指定できない動的承認ノードが指定されています。	
1039	指定できる横配置ノードが存在しません。	
1040	横配置ノードが未指定です。	
1041	設定が可能な横配置ノード数と指定された横配置ノード数が異なります。	
1042	横配置ノードの入力情報に不備があります。	
1043	同一の横配置ノードに対して指定が重複しています。	
1044	指定できない横配置ノードが指定されています。	
1045	指定できる縦配置ノードが存在しません。	
1046	縦配置ノードが未指定です。	
1047	設定が可能な縦配置ノード数と指定された縦配置ノード数が異なります。	
1048	縦配置ノードの入力情報に不備があります。	
1049	同一の縦配置ノードに対して指定が重複しています。	
1050	指定できない縦配置ノードが指定されています。	
1051	案件名が許可された最大文字数を超えています。	
1052	案件名のチェック処理でエラーが発生しました。管理者に連絡してください。	
1053	処理対象者情報が未指定です。	
1054	処理対象者情報に不備があります。	
1055	存在しない優先度が指定されています。	
1056	処理種別定義チェック処理に失敗しました。管理者に連絡してください。	
1057	印影情報に不備があります。	
1058	このフローまたは案件は印影を利用できません。	
1059	権限者の印影設定情報が存在しません。	
1060	権限者の印影として利用できない印影が指定されています。	
1061	印影チェック処理でエラーが発生しました。管理者に連絡してください。	
1062	不正な申請基準日が指定されています。	
1063	申請基準日チェックに必要な情報が取得できませんでした。	
1064	申請基準日チェック処理でエラーが発生しました。	
1065	申請できない申請基準日です。	
1066	添付ファイルの追加は禁止されています。	
1067	添付ファイルの格納領域が見つかりませんでした。	
1068	添付ファイルの格納領域の取得に失敗しました。管理者に連絡してください。	

エラーコード	エラーメッセージ	備考
1069	確認ノードの指定に不備があります。	
1070	動的承認ノードの指定に不備があります。	
1071	横配置ノードの指定に不備があります。	
1072	縦配置ノードの指定に不備があります。	
1073	指定可能な展開ノード数ではありません。	
1074	展開ノードの処理設定が未指定です。	
1075	展開ノードのノード名が未指定です。	
1082	申請基準日のフォーマットが不正です。	
1083	設定により無効化されている処理対象者情報が指定されています。	
1084	設定により無効化されているため、指定された処理を行うことはできません。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2001	案件名の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
3000	申請設定情報が存在しません。管理者に連絡してください。	
3001	申請基準日時点で無効なフローです。	
3002	ルート情報が不正です。管理者に連絡してください。	
3003	ルート情報が不正です。管理者に連絡してください。	
3004	フロー情報の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
3011	ユーザデータIDの重複チェックに失敗しました。管理者に連絡してください。	
3012	ユーザデータIDは一時保存案件で使用済みです。	
3013	ユーザデータIDは案件で使用済みです。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
7004	印影機能は設定により利用できません。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ または、IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

applyFromUnapply

imWorkflow.process.applyFromUnapply(Object parameter) : Promise<Object>

ワークフローの起票案件を申請します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs
 - confirmNodeConfigs
 - horizontalNodeConfigs
 - verticalNodeConfigs
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.applyFromUnapply({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    matterName: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    priorityLevel: "",
    temporaryDirectoryKey: "",
    branchSelects: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
起票案件申請用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
matterName	任意	文字列 (200)	案件名 未指定の場合、起票時に指定した案件名で申請
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで申請
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で申請 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし

項目名	必須/任意	型 (最大バイト数)	説明
processComment	任意	文字列 (2000)	処理コメント
priorityLevel	任意	文字列	優先度 設定値は「CodeList」を参照 未指定の場合、「通常」で申請
temporaryDirectoryKey	任意	文字列	添付ファイル一時領域ディレクトリキー
branchSelects	任意	下記参照	分岐先選択情報 分岐先設定可能ノードの場合指定必須
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1. ルート定義で設定した処理対象者で申請 2. ルート定義で「指定なし」の場合、削除可ノードであればノードを削除、削除不可の場合はエラー
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 確認対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1. ルート定義で設定した処理対象者で申請
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1. ルート定義で設定した処理対象者で申請 2. ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1. ルート定義で設定した処理対象者で申請 2. ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。

項目名	必須/任意	型	説明
forwardNodeIds	必須	配列（文字列）	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs

ノードを削除する場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  dynamicNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

confirmNodeConfigs

確認対象者なしとする場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

horizontalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「動的処理対象者設定機能」を参照してください。

verticalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「処理権限者プラグイン一覧」を参照してください。

```
{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「動的処理対象者設定機能」を参照してください。



コラム

ノードの削除可否や割当可能ノード数、利用可能なプラグインは、動的承認・横配置・縦配置ノード設定の範囲で指定できます。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列（文字列）	宛先に指定するユーザのユーザコードを指定します。（複数指定可）
userCdsCc	任意	配列（文字列）	Ccに指定するユーザのユーザコードを指定します。（複数指定可）
userCdsBcc	任意	配列（文字列）	Bccに指定するユーザのユーザコードを指定します。（複数指定可）
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。



コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「[IM-Workflow プログラミングガイド](#)」-「[ユーザプログラムの作成 - アクション処理](#)」
アクション処理パラメータにおける「ユーザパラメータ」: [userParameter](#)
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「[IM-Workflow 管理者操作ガイド](#)」-「[ロジックフローの入出力設定 - アクション処理](#)」
ロジックフロー入力設定における「ユーザパラメータ」: [imwUserParameter](#)

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「[IM-Workflow 仕様書](#)」-「[IM-FormaDesigner / IM-BIS 連携フローの追加処理](#)」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1006	起票案件申請情報が未指定です。	
0003	指定したフロー、申請基準日、権限者に対する申請権限がありません。	
0004	申請権限のチェックに失敗しました。管理者に連絡してください。	
0005	申請可能な組織権限がありません。	
0006	組織権限情報の取得に失敗しました。管理者に連絡してください。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
1018	指定できる分岐開始ノードが存在しません。	
1019	分岐先ノードが未指定です。	
1020	設定が必要な分岐先ノード数と指定された分岐先ノード数が異なります。	
1021	分岐先ノードの入力情報に不備があります。	
1022	同一の分岐先ノードに対して指定が重複しています。	
1023	選択可能なルートは単一であるのに対し分岐先ノードが複数指定されています。	
1024	指定できない分岐先ノードが指定されています。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1027	指定できる確認ノードが存在しません。	
1028	設定可能なノード数以上の確認ノードが指定されています。	
1029	確認ノードの入力情報に不備があります。	
1030	同一の確認ノードに対して指定が重複しています。	
1031	指定できない確認ノードが指定されています。	
1032	指定できる動的承認ノードが存在しません。	
1033	動的承認ノードが未指定です。	
1034	設定が可能な動的承認ノード数と指定された動的承認ノード数が異なります。	
1035	動的承認ノードの入力情報に不備があります。	
1036	同一の動的承認ノードに対して指定が重複しています。	
1037	削除できない動的承認ノードに対して処理対象者の指定がされていません。	
1038	指定できない動的承認ノードが指定されています。	
1039	指定できる横配置ノードが存在しません。	
1040	横配置ノードが未指定です。	
1041	設定が可能な横配置ノード数と指定された横配置ノード数が異なります。	
1042	横配置ノードの入力情報に不備があります。	
1043	同一の横配置ノードに対して指定が重複しています。	
1044	指定できない横配置ノードが指定されています。	
1045	指定できる縦配置ノードが存在しません。	
1046	縦配置ノードが未指定です。	
1047	設定が可能な縦配置ノード数と指定された縦配置ノード数が異なります。	
1048	縦配置ノードの入力情報に不備があります。	
1049	同一の縦配置ノードに対して指定が重複しています。	
1050	指定できない縦配置ノードが指定されています。	
1051	案件名が許可された最大文字数を超えています。	
1052	案件名のチェック処理でエラーが発生しました。管理者に連絡してください。	
1053	処理対象者情報が未指定です。	

エラーコード	エラーメッセージ	備考
1054	処理対象者情報に不備があります。	
1055	存在しない優先度が指定されています。	
1056	処理種別定義チェック処理に失敗しました。管理者に連絡してください。	
1057	印影情報に不備があります。	
1058	このフローまたは案件は印影を利用できません。	
1059	権限者の印影設定情報が存在しません。	
1060	権限者の印影として利用できない印影が指定されています。	
1061	印影チェック処理でエラーが発生しました。管理者に連絡してください。	
1062	不正な申請基準日が指定されています。	
1063	申請基準日チェックに必要な情報が取得できませんでした。	
1064	申請基準日チェック処理でエラーが発生しました。	
1065	申請できない申請基準日です。	
1066	添付ファイルの追加は禁止されています。	
1067	添付ファイルの格納領域が見つかりませんでした。	
1068	添付ファイルの格納領域の取得に失敗しました。管理者に連絡してください。	
1069	確認ノードの指定に不備があります。	
1070	動的承認ノードの指定に不備があります。	
1071	横配置ノードの指定に不備があります。	
1072	縦配置ノードの指定に不備があります。	
1073	指定可能な展開ノード数ではありません。	
1074	展開ノードの処理設定が未指定です。	
1075	展開ノードのノード名が未指定です。	
1082	申請基準日のフォーマットが不正です。	
1083	設定により無効化されている処理対象者情報が指定されています。	
1084	設定により無効化されているため、指定された処理を行うことはできません。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2001	案件名の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
3000	申請設定情報が存在しません。管理者に連絡してください。	
3001	申請基準日時点で無効なフローです。	
3002	ルート情報が不正です。管理者に連絡してください。	
3003	ルート情報が不正です。管理者に連絡してください。	
3004	フロー情報の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
7004	印影機能は設定により利用できません。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ または、IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

エラーコード	エラーメッセージ	備考
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

approve

imWorkflow.process.approve(Object parameter) : Promise<Object>

ワークフローの案件を承認します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs
 - confirmNodeConfigs
 - horizontalNodeConfigs
 - verticalNodeConfigs
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.approve({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    temporaryDirectoryKey: "",
    fieldsForDelete: [],
    branchSelects: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
承認用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須

項目名	必須/任意	型 (最大バイト数)	説明
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で処理 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
temporaryDirectoryKey	任意	文字列	添付ファイル一時領域ディレクトリキー
filedsForDelete	任意	配列 (文字列)	削除対象ファイル (ファイルID) ファイルIDを指定
branchSelects	任意	下記参照	分岐先選択情報 分岐先設定可能ノードの場合指定必須
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して処理 1.ルート定義で設定した処理対象者で処理 2.ルート定義で「指定なし」の場合、削除可ノードであればノードを削除、削除不可の場合はエラー
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 確認対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して処理 1.ルート定義で設定した処理対象者で処理
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して処理 1.ルート定義で設定した処理対象者で処理 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して処理 1.ルート定義で設定した処理対象者で処理 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。
forwardNodeIds	必須	配列（文字列）	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs

ノードを削除する場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  dynamicNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

confirmNodeConfigs

確認対象者なしとする場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

horizontalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「 動的処理対象者設定機能 」を参照してください。

verticalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「 動的処理対象者設定機能 」を参照してください。



コラム

ノードの削除可否や割当可能ノード数、利用可能なプラグインは、[動的承認・横配置・縦配置ノード設定](#) の範囲で指定できます。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列（文字列）	宛先に指定するユーザのユーザコードを指定します。（複数指定可）
userCdsCc	任意	配列（文字列）	Ccに指定するユーザのユーザコードを指定します。（複数指定可）
userCdsBcc	任意	配列（文字列）	Bccに指定するユーザのユーザコードを指定します。（複数指定可）
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。



コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「[IM-Workflow プログラミングガイド](#)」-「[ユーザプログラムの作成 - アクション処理](#)」
アクション処理パラメータにおける「ユーザパラメータ」: [userParameter](#)
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「[IM-Workflow 管理者操作ガイド](#)」-「[ロジックフローの入出力設定 - アクション処理](#)」
ロジックフロー入力設定における「ユーザパラメータ」: [imwUserParameter](#)

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「[IM-Workflow 仕様書](#)」-「[IM-FormaDesigner / IM-BIS 連携フローの追加処理](#)」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1008	案件承認情報が未指定です。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
0014	処理権限のチェックに失敗しました。管理者に連絡してください。	
0015	処理可能な組織権限がありません。	
0016	組織権限情報の取得に失敗しました。管理者に連絡してください。	
1018	指定できる分岐開始ノードが存在しません。	
1019	分岐先ノードが未指定です。	
1020	設定が必要な分岐先ノード数と指定された分岐先ノード数が異なります。	
1021	分岐先ノードの入力情報に不備があります。	
1022	同一の分岐先ノードに対して指定が重複しています。	
1023	選択可能なルートは単一であるのに対し分岐先ノードが複数指定されています。	
1024	指定できない分岐先ノードが指定されています。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1027	指定できる確認ノードが存在しません。	
1028	設定可能なノード数以上の確認ノードが指定されています。	
1029	確認ノードの入力情報に不備があります。	
1030	同一の確認ノードに対して指定が重複しています。	
1031	指定できない確認ノードが指定されています。	
1032	指定できる動的承認ノードが存在しません。	
1033	動的承認ノードが未指定です。	
1034	設定が可能な動的承認ノード数と指定された動的承認ノード数が異なります。	
1035	動的承認ノードの入力情報に不備があります。	
1036	同一の動的承認ノードに対して指定が重複しています。	
1037	削除できない動的承認ノードに対して処理対象者の指定がされていません。	
1038	指定できない動的承認ノードが指定されています。	
1039	指定できる横配置ノードが存在しません。	
1040	横配置ノードが未指定です。	
1041	設定が可能な横配置ノード数と指定された横配置ノード数が異なります。	
1042	横配置ノードの入力情報に不備があります。	
1043	同一の横配置ノードに対して指定が重複しています。	
1044	指定できない横配置ノードが指定されています。	
1045	指定できる縦配置ノードが存在しません。	
1046	縦配置ノードが未指定です。	
1047	設定が可能な縦配置ノード数と指定された縦配置ノード数が異なります。	
1048	縦配置ノードの入力情報に不備があります。	
1049	同一の縦配置ノードに対して指定が重複しています。	
1050	指定できない縦配置ノードが指定されています。	
1053	処理対象者情報が未指定です。	
1054	処理対象者情報に不備があります。	
1055	存在しない優先度が指定されています。	
1056	処理種別定義チェック処理に失敗しました。管理者に連絡してください。	

エラーコード	エラーメッセージ	備考
1057	印影情報に不備があります。	
1058	このフローまたは案件は印影を利用できません。	
1059	権限者の印影設定情報が存在しません。	
1060	権限者の印影として利用できない印影が指定されています。	
1061	印影チェック処理でエラーが発生しました。管理者に連絡してください。	
1062	不正な申請基準日が指定されています。	
1063	申請基準日チェックに必要な情報が取得できませんでした。	
1064	申請基準日チェック処理でエラーが発生しました。	
1065	申請できない申請基準日です。	
1066	添付ファイルの追加は禁止されています。	
1067	添付ファイルの格納領域が見つかりませんでした。	
1068	添付ファイルの格納領域の取得に失敗しました。管理者に連絡してください。	
1069	確認ノードの指定に不備があります。	
1070	動的承認ノードの指定に不備があります。	
1071	横配置ノードの指定に不備があります。	
1072	縦配置ノードの指定に不備があります。	
1073	指定可能な展開ノード数ではありません。	
1074	展開ノードの処理設定が未指定です。	
1075	展開ノードのノード名が未指定です。	
1076	削除対象ファイル情報に不備があります。	
1077	このノードではファイルを削除できません。	
1078	システム案件IDが未指定です。	
1079	ノードIDが未指定です。	
1082	申請基準日のフォーマットが不正です。	
1083	設定により無効化されている処理対象者情報が指定されています。	
1084	設定により無効化されているため、指定された処理を行うことはできません。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2001	案件名の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2002	案件が存在しないため、案件名を補完できませんでした。	
2003	案件の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
3006	処理設定情報を取得できませんでした。管理者に連絡してください。	
3007	処理設定情報の取得に失敗しました。管理者に連絡してください。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
7004	印影機能は設定により利用できません。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ または、IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

エラーコード	エラーメッセージ	備考
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

approveEnd

imWorkflow.process.approveEnd(Object parameter) : Promise<Object>

ワークフローの案件を承認終了します。

Contents
<ul style="list-style-type: none"> ■ サンプルコード ■ パラメータ <ul style="list-style-type: none"> ■ processParameter <ul style="list-style-type: none"> ■ authUserDepartmentInfo ■ nego ■ optionalParameter ■ 返却値 <ul style="list-style-type: none"> ■ エラーコード

サンプルコード

```
imWorkflow.process.approveEnd({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    temporaryDirectoryKey: "",
    fieldsForDelete: [],
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
承認終了用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で処理 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント

項目名	必須/任意	型 (最大バイト数)	説明
temporaryDirectoryKey	任意	文字列	添付ファイル一時領域ディレクトリキー
fieldsForDelete	任意	配列 (文字列)	削除対象ファイル (ファイルID) ファイルIDを指定
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。

アクション処理に受け渡したいパラメータを指定してください。

i コラム

印影を指定する場合、`imwStampId` を指定してください。
 印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

i コラム

「`optionalParameter.userParameter`」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。
 「`optionalParameter.userParameter`」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
 アクション処理パラメータにおける「ユーザパラメータ」：`userParameter`
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
 ロジックフロー入力設定における「ユーザパラメータ」：`imwUserParameter`

formaParam

パラメータ「`formaParam`」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。
 詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

返却値

- `Promise<Object>`
 返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
 処理が成功した場合の返却値はありません。
 処理が失敗した場合は`promise.catch()`で返却値を受け取ってください。

項目名	型	説明
<code>workflowResult.data.errorCode</code>	文字列	エラーコード 処理が失敗した場合に設定されます。
<code>workflowResult.data.localizedErrorMessage</code>	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1007	案件終了情報が未指定です。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
0014	処理権限のチェックに失敗しました。管理者に連絡してください。	
0015	処理可能な組織権限がありません。	
0016	組織権限情報の取得に失敗しました。管理者に連絡してください。	
1025	コメントが許可された最大文字数を超過しています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1056	処理種別定義チェック処理に失敗しました。管理者に連絡してください。	
1057	印影情報に不備があります。	
1058	このフローまたは案件は印影を利用できません。	
1059	権限者の印影設定情報が存在しません。	
1060	権限者の印影として利用できない印影が指定されています。	
1061	印影チェック処理でエラーが発生しました。管理者に連絡してください。	
1066	添付ファイルの追加は禁止されています。	
1067	添付ファイルの格納領域が見つかりませんでした。	

エラーコード	エラーメッセージ	備考
1068	添付ファイルの格納領域の取得に失敗しました。管理者に連絡してください。	
1076	削除対象ファイル情報に不備があります。	
1077	このノードではファイルを削除できません。	
1078	システム案件IDが未指定です。	
1079	ノードIDが未指定です。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2002	案件が存在しないため、案件名を補完できませんでした。	
2003	案件の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
3006	処理設定情報を取得できませんでした。管理者に連絡してください。	
3007	処理設定情報の取得に失敗しました。管理者に連絡してください。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
7004	印影機能は設定により利用できません。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ または、IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

confirm

imWorkflow.process.confirm(Object parameter) : Promise<Object>

ワークフローの案件を確認します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.confirm({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserDepartmentInfo: {},
    confirmComment: "",
  }
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object

- parameter.processParameter : Object (下記参照)
確認用パラメータ情報オブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で確認 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
confirmComment	任意	文字列 (2000)	確認コメント

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

返却値

- Promise<Object>
返却オブジェクト (下記参照)
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1004	未完了案件の確認情報が未指定です。	
0001	確認可能な組織権限がありません。	
0002	組織権限情報の取得に失敗しました。管理者に連絡してください。	
1003	完了案件確認情報が未指定です。	
0007	確認可能な組織権限がありません。	
0008	組織権限情報の取得に失敗しました。管理者に連絡してください。	
0009	指定した案件、処理ノードに対する確認権限がありません。	

エラーコード	エラーメッセージ	備考
0010	確認権限のチェックに失敗しました。管理者に連絡してください。	
1016	システム案件IDが未指定です。	
1017	ノードIDが未指定です。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
0013	指定した案件、処理ノード、権限者に対する処理権がありません。	
6051	案件が見つかりませんでした。	
6052	案件情報が取得できません。	
6053	権限がありません。	

deny

imWorkflow.process.deny(Object parameter) : Promise<Object>

ワークフローの案件を否認します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.deny({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
否認用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で処理 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「IM-Workflow 管理者操作ガイド」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。

コラム

印影を指定する場合、`imwStampId` を指定してください。
印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

コラム

「OptionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「OptionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

返却値

- `Promise<Object>`
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合は`promise.catch()`で返却値を受け取ってください。

項目名	型	説明
<code>workflowResult.data.errorCode</code>	文字列	エラーコード 処理が失敗した場合に設定されます。
<code>workflowResult.data.localizedErrorMessage</code>	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1009	案件否認情報が未指定です。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ

エラーコード	エラーメッセージ	備考
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー

discontinue

imWorkflow.process.discontinue(Object parameter) : Promise<Object>

ワークフローの案件を取止めます。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.discontinue({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
取止め用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で処理 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
nego	任意	下記参照	根回し情報

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。

コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: userParameter
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: imwUserParameter

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1010	取止め情報が未指定です。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー

reapply

imWorkflow.process.reapply(Object parameter) : Promise<Object>

ワークフローの案件を再申請します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs
 - confirmNodeConfigs
 - horizontalNodeConfigs
 - verticalNodeConfigs
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.reapply({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    matterName: "",
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    priorityLevel: "",
    temporaryDirectoryKey: "",
    fieldsForDelete: [],
    branchSelects: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
再申請用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
matterName	任意	文字列 (200)	案件名 未指定の場合、申請時に指定した案件名で申請
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで申請

項目名	必須/任意	型 (最大バイト数)	説明
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で申請 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
priorityLevel	任意	文字列	優先度 設定値は「CodeList」を参照 未指定の場合、「通常」で申請
temporaryDirectoryKey	任意	文字列	添付ファイル一時領域ディレクトリキー
fileldsForDelete	任意	配列 (文字列)	削除対象ファイル (ファイルID) ファイルIDを指定
branchSelects	任意	下記参照	分岐先選択情報 分岐先設定可能ノードの場合指定必須
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、削除可ノードであればノードを削除、削除不可の場合はエラー
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 確認対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 処理対象者設定可能ノードに設定されている場合に指定 未指定の場合、以下の優先度で決定して申請 1.ルート定義で設定した処理対象者で申請 2.ルート定義で「指定なし」の場合、割当可能ノード数の最小0であればノードを削除、0以外の場合はエラー
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。
forwardNodeIds	必須	配列（文字列）	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs

ノードを削除する場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  dynamicNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

confirmNodeConfigs

確認対象者なしとする場合、processTargetConfigsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
processTargetConfigs	任意	オブジェクト	処理対象プラグイン情報を指定します。 詳細は「 動的処理対象者設定機能 」を参照してください。

horizontalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```

{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}

```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「 動的処理対象者設定機能 」を参照してください。

verticalNodeConfigs

ノードを削除する場合、matterNodeExpansionsに何も設定しないでください。利用可能なプラグインIDについては「[処理権限者プラグイン一覧](#)」を参照してください。

```

{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      matterNodeExpansions: [
        {
          nodeName: "string",
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}

```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
matterNodeExpansions	任意	オブジェクト	ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。詳細は「 動的処理対象者設定機能 」を参照してください。



コラム

ノードの削除可否や割当可能ノード数、利用可能なプラグインは、[動的承認](#)・[横配置](#)・[縦配置ノード設定](#) の範囲で指定できます。

nego

```

{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}

```

項目名	必須/任意	型	説明
userCdsTo	任意	配列（文字列）	宛先に指定するユーザのユーザコードを指定します。（複数指定可）
userCdsCc	任意	配列（文字列）	Ccに指定するユーザのユーザコードを指定します。（複数指定可）
userCdsBcc	任意	配列（文字列）	Bccに指定するユーザのユーザコードを指定します。（複数指定可）
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。



コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「[IM-Workflow プログラミングガイド](#)」-「[ユーザプログラムの作成 - アクション処理](#)」
アクション処理パラメータにおける「ユーザパラメータ」: [userParameter](#)
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「[IM-Workflow 管理者操作ガイド](#)」-「[ロジックフローの入出力設定 - アクション処理](#)」
ロジックフロー入力設定における「ユーザパラメータ」: [imwUserParameter](#)

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「[IM-Workflow 仕様書](#)」-「[IM-FormaDesigner / IM-BIS 連携フローの追加処理](#)」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1011	案件再申請情報が未指定です。	
0003	指定したフロー、申請基準日、権限者に対する申請権限がありません。	
0004	申請権限のチェックに失敗しました。管理者に連絡してください。	
0005	申請可能な組織権限がありません。	
0006	組織権限情報の取得に失敗しました。管理者に連絡してください。	
0011	権限者組織情報が不正です。	
0012	指定された所属組織情報は処理権限がありません。	
1018	指定できる分岐開始ノードが存在しません。	
1019	分岐先ノードが未指定です。	
1020	設定が必要な分岐先ノード数と指定された分岐先ノード数が異なります。	
1021	分岐先ノードの入力情報に不備があります。	
1022	同一の分岐先ノードに対して指定が重複しています。	
1023	選択可能なルートは単一であるのに対し分岐先ノードが複数指定されています。	
1024	指定できない分岐先ノードが指定されています。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1027	指定できる確認ノードが存在しません。	
1028	設定可能なノード数以上の確認ノードが指定されています。	
1029	確認ノードの入力情報に不備があります。	
1030	同一の確認ノードに対して指定が重複しています。	
1031	指定できない確認ノードが指定されています。	
1032	指定できる動的承認ノードが存在しません。	
1033	動的承認ノードが未指定です。	
1034	設定が可能な動的承認ノード数と指定された動的承認ノード数が異なります。	
1035	動的承認ノードの入力情報に不備があります。	
1036	同一の動的承認ノードに対して指定が重複しています。	
1037	削除できない動的承認ノードに対して処理対象者の指定がされていません。	
1038	指定できない動的承認ノードが指定されています。	
1039	指定できる横配置ノードが存在しません。	
1040	横配置ノードが未指定です。	
1041	設定が可能な横配置ノード数と指定された横配置ノード数が異なります。	
1042	横配置ノードの入力情報に不備があります。	
1043	同一の横配置ノードに対して指定が重複しています。	
1044	指定できない横配置ノードが指定されています。	
1045	指定できる縦配置ノードが存在しません。	
1046	縦配置ノードが未指定です。	
1047	設定が可能な縦配置ノード数と指定された縦配置ノード数が異なります。	
1048	縦配置ノードの入力情報に不備があります。	
1049	同一の縦配置ノードに対して指定が重複しています。	
1050	指定できない縦配置ノードが指定されています。	
1051	案件名が許可された最大文字数を超えています。	
1052	案件名のチェック処理でエラーが発生しました。管理者に連絡してください。	
1053	処理対象者情報が未指定です。	

エラーコード	エラーメッセージ	備考
1054	処理対象者情報に不備があります。	
1056	処理種別定義チェック処理に失敗しました。管理者に連絡してください。	
1057	印影情報に不備があります。	
1058	このフローまたは案件は印影を利用できません。	
1059	権限者の印影設定情報が存在しません。	
1060	権限者の印影として利用できない印影が指定されています。	
1061	印影チェック処理でエラーが発生しました。管理者に連絡してください。	
1062	不正な申請基準日が指定されています。	
1063	申請基準日チェックに必要な情報が取得できませんでした。	
1064	申請基準日チェック処理でエラーが発生しました。	
1065	申請できない申請基準日です。	
1066	添付ファイルの追加は禁止されています。	
1067	添付ファイルの格納領域が見つかりませんでした。	
1068	添付ファイルの格納領域の取得に失敗しました。管理者に連絡してください。	
1069	確認ノードの指定に不備があります。	
1070	動的承認ノードの指定に不備があります。	
1071	横配置ノードの指定に不備があります。	
1072	縦配置ノードの指定に不備があります。	
1073	指定可能な展開ノード数ではありません。	
1074	展開ノードの処理設定が未指定です。	
1075	展開ノードのノード名が未指定です。	
1082	申請基準日のフォーマットが不正です。	
1083	設定により無効化されている処理対象者情報が指定されています。	
1084	設定により無効化されているため、指定された処理を行うことはできません。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2001	案件名の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
3000	申請設定情報が存在しません。管理者に連絡してください。	
3001	申請基準日時点で無効なフローです。	
3002	ルート情報が不正です。管理者に連絡してください。	
3003	ルート情報が不正です。管理者に連絡してください。	
3004	フロー情報の取得に失敗しました。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
7004	印影機能は設定により利用できません。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ または、IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

エラーコード	エラーメッセージ	備考
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

reserve

imWorkflow.process.reserve(Object parameter) : Promise<Object>

ワークフローの案件を保留します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.reserve({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    processComment: "",
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
保留用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
processComment	任意	文字列 (2000)	処理コメント
nego	任意	下記参照	根回し情報

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列（文字列）	宛先に指定するユーザのユーザコードを指定します。（複数指定可）
userCdsCc	任意	配列（文字列）	Ccに指定するユーザのユーザコードを指定します。（複数指定可）
userCdsBcc	任意	配列（文字列）	Bccに指定するユーザのユーザコードを指定します。（複数指定可）
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。

コラム

印影を指定する場合、`imwStampId` を指定してください。
印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「[IM-Workflow プログラミングガイド](#)」 - 「[ユーザプログラムの作成 - アクション処理](#)」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「[IM-Workflow 管理者操作ガイド](#)」 - 「[ロジックフローの入出力設定 - アクション処理](#)」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

返却値

- `Promise<Object>`
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合は`promise.catch()`で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。

項目名	型	説明
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1013	保留情報が未指定です。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー-排他制御エラー

reserveCancel

imWorkflow.process.reserveCancel(Object parameter) : Promise<Object>

ワークフローの案件を保留解除します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.reserveCancel({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserCd: "",
    processComment: "",
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
保留解除用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
-----	-------	------------	----

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
processComment	任意	文字列 (2000)	処理コメント
nego	任意	下記参照	根回し情報

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。

 コラム

印影を指定する場合、`imwStampId` を指定してください。
印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

i コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」：userParameter
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」：imwUserParameter

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合はpromise.catch()で返却値を受け取ってください。

項目名	型	説明
workflowResult.data.errorCode	文字列	エラーコード 処理が失敗した場合に設定されます。
workflowResult.data.localizedErrorMessage	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1012	保留解除情報が未指定です。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
1025	コメントが許可された最大文字数を超過しています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー

saveTemporary

imWorkflow.process.saveTemporary(Object parameter) : Promise<Object>

ワークフローを一時保存します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.saveTemporary({
  processParameter: {
    userDataId: "",
    flowId: "",
    matterName: "",
    applyBaseDate: "",
    applyAuthUserCd: "",
    processComment: ""
  },
  optionalParameter: {}
})
.then(function(result) {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
一時保存用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
userDataId	任意	文字列(20)	ユーザデータID 「案件一覧」画面から遷移した場合は、一覧で選択した一時保存情報のユーザデータIDが設定されるため不要 未指定の場合、Identifier.get()で一時保存情報を新規作成 指定した場合、当該キーで特定される一時保存情報が存在しない場合、一時保存情報を新規作成 当該キーで特定される一時保存情報が存在する場合、一時保存情報を更新
flowId	任意	文字列	フローID 一時保存情報を新規作成時は必須 一時保存情報を更新時は不要
matterName	任意	文字列 (200)	案件名 未指定の場合、以下の優先度で決定して一時保存 1.一時保存時に指定した案件名 (一時保存情報更新の場合) 2.フロー名
applyBaseDate	任意	文字列	申請基準日 未指定の場合、「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択した申請基準日が設定される 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、アカウントコンテキストのタイムゾーンにおける現在日付で一時保存 “yyyy/MM/dd”形式の文字列で指定
applyAuthUserCd	任意	文字列	申請権限者コード 未指定の場合、実行ユーザで一時保存
processComment	任意	文字列 (2000)	処理コメント

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。

i コラム

印影を指定する場合、`imwStampId` を指定してください。
 印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

i コラム

「`optionalParameter.userParameter`」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。
 「`optionalParameter.userParameter`」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
 アクション処理パラメータにおける「ユーザパラメータ」：`userParameter`
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
 ロジックフロー入力設定における「ユーザパラメータ」：`imwUserParameter`

formaParam

パラメータ「`formaParam`」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。
 詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

返却値

- `Promise<Object>`
 返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
 処理が成功した場合は`promise.then()`で返却値を受け取ってください。
 処理が失敗した場合は`promise.catch()`で返却値を受け取ってください。

項目名	型	説明
<code>userDataId</code>	文字列	ユーザデータID 処理が成功した場合に設定されます。
<code>workflowResult.data.errorCode</code>	文字列	エラーコード 処理が失敗した場合に設定されます。
<code>workflowResult.data.localizedErrorMessage</code>	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1001	一時保存情報が未指定です。	
1002	一時保存情報が未指定です。	
6000	一時保存情報の取得に失敗しました。	
6001	一時保存情報が存在しません。	
6002	一時保存情報に関連する案件プロパティの取得に失敗しました。	
7000	一時保存機能は設定により利用できません。管理者に連絡してください。	
7001	一時保存機能の利用可否設定情報の取得に失敗しました。管理者に連絡してください。	
0017	指定したフロー、申請基準日、権限者に対する一時保存情報登録処理権限がありません。	
0018	申請権限のチェックに失敗しました。管理者に連絡してください。	
0019	指定した一時保存情報、権限者に対する一時保存情報更新権限がありません。	
0020	申請権限のチェックに失敗しました。管理者に連絡してください。	
1025	コメントが許可された最大文字数を超過しています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1051	案件名が許可された最大文字数を超過しています。	

エラーコード	エラーメッセージ	備考
1052	案件名のチェック処理でエラーが発生しました。管理者に連絡してください。	
1080	フローIDが未指定です。	
1081	ユーザデータIDが未指定です。	
2004	案件名の補完情報フロー名取得に失敗しました。	
3008	ユーザデータIDの採番に失敗しました。	
3009	更新対象の一時保存情報が見つかりませんでした。	
3010	更新対象の一時保存情報取得に失敗しました。管理者に連絡してください。	
3011	ユーザデータIDの重複チェックに失敗しました。管理者に連絡してください。	
3012	ユーザデータIDは一時保存案件で使用済みです。	
3013	ユーザデータIDは案件で使用済みです。	
6000	一時保存情報の取得に失敗しました。	
6001	一時保存情報が存在しません。	
6002	一時保存情報に関連する案件プロパティの取得に失敗しました。	
9001	(内部処理エラーメッセージ)	IM-FormaDesigner / IM-BIS の内部処理でシステムエラーが発生した場合
9010	(入力チェックエラーメッセージ)	IM-FormaDesigner の入力チェックエラーメッセージ IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ
9011	入力チェックでシステムエラーが発生しました。	IM-FormaDesigner の入力チェックでシステムエラー IM-FormaDesigner / IM-BIS で作成したワークフローの場合のみ

sendBack

imWorkflow.process.sendBack(Object parameter) : Promise<Object>

ワークフローの案件を差戻します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - nego
 - optionalParameter
- 返却値
 - エラーコード

サンプルコード

```
imWorkflow.process.sendBack({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    sendBackNodeIds: [],
    authUserCd: "",
    authUserDepartmentInfo: {},
    processComment: "",
    nego: {}
  },
  optionalParameter: {}
})
.then(function() {
  // success
})
.catch(function(error) {
  // error
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
差戻し用パラメータ情報オブジェクト

- parameter.optionalParameter : Object (下記参照)
 ユーザパラメータを内包するオブジェクト

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
sendBackNodeIds	任意	文字列/文字列 (1次元配列)	差戻し先ノードID 未指定の場合、申請ノードへ差戻し
authUserCd	任意	文字列	権限者コード 未指定の場合、実行ユーザで処理
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 未指定の場合、以下の優先順位で該当する組織情報で処理 1.カレント組織 (代理設定の場合は除く) 2.主所属 3.「会社コード」「組織セットコード」「組織コード」の昇順で1つ目の組織情報 4.所属なし
processComment	任意	文字列 (2000)	処理コメント
nego	任意	下記参照	根回し情報

authUserDepartmentInfo

所属なしユーザを指定する場合は、当オブジェクト配下のプロパティに空文字を指定してください。

```
{
  departmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

nego

```
{
  nego: {
    userCdsTo: [
      "string"
    ],
    userCdsCc: [
      "string"
    ],
    userCdsBcc: [
      "string"
    ],
    subject: "string",
    text: "string"
  }
}
```

項目名	必須/任意	型	説明
userCdsTo	任意	配列 (文字列)	宛先に指定するユーザのユーザコードを指定します。(複数指定可)
userCdsCc	任意	配列 (文字列)	Ccに指定するユーザのユーザコードを指定します。(複数指定可)
userCdsBcc	任意	配列 (文字列)	Bccに指定するユーザのユーザコードを指定します。(複数指定可)
subject	任意	文字列	送信するメールの件名を指定します。

項目名	必須/任意	型	説明
text	任意	文字列	メール定義で設定した内容の「根回し時のコメント」として追記したい内容を指定します。 メール定義については、「 IM-Workflow 管理者操作ガイド 」を参照してください。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、`imwStampId` を指定してください。
印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。



コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「[IM-Workflow プログラミングガイド](#)」-「[ユーザプログラムの作成 - アクション処理](#)」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
- IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「[IM-Workflow 管理者操作ガイド](#)」-「[ロジックフローの入出力設定 - アクション処理](#)」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
処理が成功した場合の返却値はありません。
処理が失敗した場合は`promise.catch()`で返却値を受け取ってください。

項目名	型	説明
<code>workflowResult.data.errorCode</code>	文字列	エラーコード 処理が失敗した場合に設定されます。
<code>workflowResult.data.localizedErrorMessage</code>	文字列	エラーメッセージ 処理が失敗した場合に設定されます。

エラーコード

エラーコード	エラーメッセージ	備考
1014	差戻し情報が未指定です。	
0013	指定した案件、処理ノード、権限者に対する処理権限がありません。	
1025	コメントが許可された最大文字数を超えています。	
1026	コメントのチェック処理でエラーが発生しました。管理者に連絡してください。	
1082	差戻し先ノードのチェックに失敗しました。管理者に連絡してください。	
1083	差戻し先ノードのチェック情報を取得できませんでした。	
1084	指定できない差戻し先が指定されています。	

エラーコード	エラーメッセージ	備考
1085	差戻し先の組み合わせが不正です。別ルートの子ノードが混在している可能性があります。	
2000	権限者組織情報の補完に必要な情報の取得に失敗しました。管理者に連絡してください。	
2005	ノード情報を取得できないため、差戻し先を補完できませんでした。管理者に連絡してください。	
2006	ノード情報の取得に失敗したため、差戻し先を補完できませんでした。管理者に連絡してください。	
3005	根回しアドレス情報を生成するためのユーザ情報の取得に失敗しました。	
7002	根回し機能は設定により利用できません。管理者に連絡してください。	
7003	根回しのチェック処理でエラーが発生しました。管理者に連絡してください。	
9000	処理に失敗しました。	案件処理プラグイン紐付け未定義のシステムエラー
9001	(アクション処理エラーメッセージ)	アクション処理で設定したエラーメッセージ
9002	処理に失敗しました。案件に対して他の操作が実行されているため処理できません。	ワークフロー排他制御エラー

modal

ワークフロー処理モーダル表示オブジェクトです。

ワークフロー処理モーダルを表示します。

このオブジェクトの各メソッドを実行する場合は、予めセキュアトークンをブラウザで保持しておく必要があります。

セキュアトークンは、head タグの中で以下の様に meta タグを記述して imSecureToken の出力値を保持してください。

```
<meta http-equiv="X-Intramart-Secure-Token" content='<imart type="imSecureToken" mode="value" />' />
```



コラム

imSecureToken の詳細は、以下を参照してください。

- スクリプト開発の場合
 - 「スクリプト開発向けタグライブラリ/画面共通」 - 「imSecureTokenタグ」
- JavaEE開発の場合
 - 「JavaEE開発向けタグライブラリ/intra-mart標準」 - 「imSecureTokenタグ」

showApply

imWorkflow.modal.showApply(Object parameter) : Promise<Object>

申請モーダルを表示します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs／confirmNodeConfigs
 - horizontalNodeConfigs／verticalNodeConfigs
 - interfaceControl
 - optionalParameter
- 返却値

サンプルコード

```

imWorkflow.modal.showApply({
  processParameter: {
    flowId: "",
    userDataId: "",
    matterName: "",
    applyAuthUserCd: "",
    applyBaseDate: "",
    authUserDepartmentInfo: {},
    priorityLevel: "",
    processComment: "",
    branchSelects: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    interfaceControl: {}
  },
  optionalParameter: {},
  rebootModal: false
})
.then(function(result) {
  // modal close
  if (result.isProcessDone) {
    // success
    alert("done");
  }
  imWorkflow.transition.afterProcess();
});

```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
申請モーダル表示用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト
- parameter.rebootModal : boolean
モーダルを閉じた際に入力した内容を保持するかを指定
true: モーダルを閉じる度に入力内容を破棄し、再度このメソッドを実行した際はモーダル表示処理を再実行します。
false: モーダルを閉じても入力内容はコンテンツ画面に保持され、再度このメソッドを実行した際は保持された情報を表示します。画面遷移をした場合、入力内容は破棄されます。(初期値)

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
flowId	任意	文字列	フローID 「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択したフローIDが設定されるため不要 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、必須
userDataId	任意	文字列 (20)	ユーザデータID 未指定の場合、ワークフローパラメータ「 ユーザデータID設定方式 」の設定によりIDを決定する。 一時保存状態から申請する場合、一時保存情報のユーザデータIDを指定
matterName	任意	文字列 (200)	案件名 案件名に初期値を設定したい場合に指定
applyAuthUserCd	任意	文字列	申請権限者コード 申請権限者に初期値を設定したい場合に指定
applyBaseDate	任意	日付/文字列	申請基準日 未指定の場合、「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択した申請基準日が設定される 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、アカウントコンテキストのタイムゾーンにおける現在日付で申請 “yyyy/MM/dd”形式の文字列指定も可能
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 担当組織に初期値を設定したい場合に指定
priorityLevel	任意	文字列	優先度 優先度に初期値を設定したい場合に指定 設定値は「 CodeList 」を参照
processComment	任意	文字列 (2000)	処理コメント 処理コメントに初期値を設定したい場合に指定
branchSelects	任意	下記参照	分岐先選択情報 ルート選択に初期値を設定したい場合に指定

項目名	必須/任意	型 (最大バイト数)	説明
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
interfaceControl	任意	下記参照	インタフェース制御 モーダル画面のインタフェースを変更したい場合に指定

authUserDepartmentInfo

```
{
  authUserDepartmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。
forwardNodeIds	必須	配列 (文字列)	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs / confirmNodeConfigs

```
{
  dynamicNodeConfigs: [
    {
      enable: true,
      nodeId: "string",
      nodeInformation: "string",
      readonlyEnableToggle: false,
      searchCondition: {
        criteria: "any"
      },
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      searchCondition: {
        criteria: "any"
      },
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
enable	任意	boolean	<p>「フロー設定」においてノードを設定するか否かの初期値を制御します。 この設定は、動的承認ノードに対してのみ有効です。 フロー定義における動的承認ノードの削除設定が「削除可能」な場合のみ有効な項目です。 true の場合、処理対象者を設定する状態で表示します。 false の場合、ノードを削除する状態で表示します。 「interfaceControl.display」と組み合わせて、両方にfalseを設定した場合、該当の動的承認ノードを削除（ノードスキップ）できます。 未指定の場合、trueとして動作します。</p>
nodeId	必須	文字列	設定対象のノードIDを指定します。
nodeInformation	任意	文字列	ノードに対する任意のメッセージを指定します。
readonlyEnableToggle	任意	boolean	<p>「フロー設定」において「ノード有効可否トグル」の状態変更可否を制御します。 この設定は、動的承認ノードに対してのみ有効です。 フロー定義における動的承認ノードの削除設定が「削除可能」な場合のみ有効な項目です。 true の場合、「ノード有効可否トグル」を状態変更不可の状態を表示します。 false の場合、「ノード有効可否トグル」を状態変更可能の状態を表示します。 「interfaceControl.readonly」=true と組み合わせて指定した場合、当該項目の指定は無視されます。 未指定の場合、falseとして動作します。</p>
searchCondition	任意	オブジェクト	<p>処理対象者の検索時条件を指定します。 「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。 当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索（キーワード タブ）」のみです。</p>
criteria	任意	オブジェクト	<p>ユーザ検索（キーワード タブ）に対する暗黙条件を指定可能です。 具体的には、「IM-共通マスタ 検索画面起動引数一覧」において以下に該当する引数を指定可能です。</p> <ul style="list-style-type: none"> 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 対象となる引数 分類 = 「暗黙条件」 <p>「searchCondition」を指定した場合、省略することはできません。 「searchCondition」・「criteria」の両方が省略された場合、検索条件（暗黙条件）指定なしで動作します。</p>
processTargetConfigs	任意	オブジェクト	<p>処理対象プラグイン情報を指定します。 利用可能なプラグインIDについては「処理権限者プラグイン一覧」を参照してください。 未指定の場合、現在設定済みの処理対象者が適用されます。</p>



コラム

「searchCondition」、または、「processTargetConfigs」を設定した場合、保存済み設定機能は利用できません。



コラム

ノードの削除可否や利用可能なプラグインは、動的承認・横配置・縦配置ノード設定の範囲で指定できます。

```

{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      dispatchControl: {
        max: "number",
        min: "number"
      },
      searchCondition: {
        criteria: "string"
      },
      matterNodeExpansions: [
        {
          nodeName: "string",
          searchCondition: {
            criteria: "any"
          },
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}

```

```

{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      dispatchControl: {
        max: "number",
        min: "number"
      },
      searchCondition: {
        criteria: "string"
      },
      matterNodeExpansions: [
        {
          nodeName: "string",
          searchCondition: {
            criteria: "any"
          },
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}

```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
nodeInformation	任意	文字列	ノードに対する任意のメッセージを指定します。
dispatchControl	任意	オブジェクト	割当可能ノード数を指定します。 当プロパティによる制御は、フロー定義で設定済みの割当可能ノード数の範囲内においてのみ有効です。 「max」プロパティを設定することで、割当可能ノード数 (最大) を指定可能です。 「min」プロパティを設定することで、割当可能ノード数 (最小) を指定可能です。
searchCondition	任意	オブジェクト	処理対象者の検索時条件を指定します。 「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。 当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索 (キーワード タブ)」のみです。

項目名	必須/任意	型 (最大バイト数)	説明
criteria	任意	オブジェクト	<p>ユーザ検索 (キーワード タブ) に対する暗黙条件を指定可能です。 具体的には、「IM-共通マスタ 検索画面起動引数一覧」において以下に該当する引数を指定可能です。</p> <ul style="list-style-type: none"> 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 対象となる引数 分類 = 「暗黙条件」 <p>「searchCondition」を指定した場合、省略することはできません。 「searchCondition」・「criteria」の両方が省略された場合、検索条件 (暗黙条件) 指定なしで動作します。</p>
matterNodeExpansions	任意	オブジェクト	<p>ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。</p> <p>「interfaceControl.display」が false (表示しない) の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最小」「最大」値の範囲で、当パラメータで定義したノード数で展開されます。 <p>「interfaceControl.display」が true (表示する) の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> 展開可能なノード数の最小個数は、次の値のどちらか「大きいほう」を採用します。 <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最小」値 dispatchControl.min で指定された値 (有効な値を指定している場合のみ考慮します) 上記ルールで決定した最小個数より当パラメータで指定したノード数の方が少ない場合、処理を行うためには画面上での追加設定が必要です 展開可能なノード数の最大個数は、次の値のどちらか「小さいほう」を採用します。 <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最大」値 dispatchControl.max で指定された値 (有効な値を指定している場合のみ考慮します) 上記ルールで決定した最大個数より当パラメータで指定したノード数の方が多い場合、最大個数を超えた分は設定されません。 <p>対象のノードを削除 (ノードスキップ) する場合、以下の設定を行ってください。</p> <ul style="list-style-type: none"> 「matterNodeExpansions」の設定を省略または空で指定、かつ、displayにfalseを指定。
nodeName	任意	文字列 (50)	ノード名を指定します。
processTargetConfigs	任意	オブジェクト	<p>処理対象プラグイン情報を指定します。 利用可能なプラグインIDについては「処理権限者プラグイン一覧」を参照してください。 未指定の場合、現在設定済みの処理対象者が適用されます。</p>



コラム

「searchCondition」、または、「matterNodeExpansions」を設定した場合、保存済み設定機能は利用できません。



コラム

割当可能ノード数や利用可能なプラグインは、[動的承認](#)・[横配置](#)・[縦配置ノード設定](#) の範囲で指定できます。

interfaceControl

申請モダルの各項目に対するインタフェースの制御 (表示、参照、必須) を指定します。
 制御の内容と設定値ごとの動作は以下の通りです。

- 表示 (display)
 - true : 項目を表示します。(初期値)
 - false : 項目を非表示にします。
- 参照 (readonly)
 - true : 項目を入力不可 (参照のみ) にします。
 - false : 項目を入力可とします。(初期値)
- 必須 (required)

- true : 項目を必須入力にします。
- false : 項目を任意入力にします。(初期値)

項目ごとに制御できる内容は異なります。

```
{
  interfaceControl: {
    matterName: {
      display: true,
      readonly: false
    },
    applyBaseDate: {
      display: true
    },
    applyAuthUserCd: {
      display: true
    },
    authUserDepartmentInfo: {
      display: true,
      readonly: false
    },
    priorityLevel: {
      display: true,
      readonly: false
    },
    processComment: {
      display: true,
      readonly: false,
      required: false
    },
    attachmentFile: {
      display: true
    },
    branchSelects: [
      {
        branchStartNodeid: "string",
        display: true,
        readonly: false
      }
    ],
    dynamicNodeConfigs: [
      {
        nodeid: "string",
        display: true,
        readonly: false
      }
    ],
    confirmNodeConfigs: [
      {
        nodeid: "string",
        display: true,
        readonly: false
      }
    ],
    horizontalNodeConfigs: [
      {
        nodeid: "string",
        display: true,
        readonly: false
      }
    ],
    verticalNodeConfigs: [
      {
        nodeid: "string",
        display: true,
        readonly: false
      }
    ],
    stamp: {
      display: true
    },
    nego: {
      display: true,
    }
  }
}
```

項目名	必須/任意	型	説明
matterName	任意	オブジェクト	案件名に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。

項目名	必須/任意	型	説明
applyBaseDate	任意	オブジェクト	申請基準日に対するインタフェース制御を指定します。 「display」を設定できます。
applyAuthUserCd	任意	オブジェクト	申請権限者に対するインタフェース制御を指定します。 「display」を設定できます。
authUserDepartmentInfo	任意	オブジェクト	担当組織に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。
priorityLevel	任意	オブジェクト	優先度に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。
processComment	任意	オブジェクト	処理コメントに対するインタフェース制御を指定します。 「display」と「readonly」と「required」を設定できます。
attachmentFile	任意	オブジェクト	添付ファイルに対するインタフェース制御を指定します。 「display」を設定できます。
branchSelects	任意	配列 (オブジェクト)	分岐先選択のインタフェース制御を指定します。 分岐先選択では、分岐先となる分岐先開始ノード (branchStartNodeId) ごとに 「display」、「readonly」を設定できます。
dynamicNodeConfigs	任意	配列 (オブジェクト)	動的承認ノードのインタフェース制御を指定します。 動的承認ノード (nodeId) ごとに「display」、「readonly」を設定できます。
confirmNodeConfigs	任意	配列 (オブジェクト)	確認ノードのインタフェース制御を指定します。 確認ノード (nodeId) ごとに「display」、「readonly」を設定できます。
horizontalNodeConfigs	任意	配列 (オブジェクト)	横配置ノードのインタフェース制御を指定します。 横配置ノード (nodeId) ごとに「display」、「readonly」を設定できます。
verticalNodeConfigs	任意	配列 (オブジェクト)	縦配置ノードのインタフェース制御を指定します。 縦配置ノード (nodeId) ごとに「display」、「readonly」を設定できます。
stamp	任意	オブジェクト	印影に対するインタフェース制御を指定します。 「display」を設定できます。
nego	任意	オブジェクト	根回しメールに対するインタフェース制御を指定します。 「display」を設定できます。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。

コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
正常終了時はpromise.then()で返却値を受け取ってください。

項目名	型	説明
isProcessDone	boolean	処理実行フラグ 処理を実行し、モーダル画面が閉じた場合に true を返却します。 処理を実行せずに、×ボタン押下でモーダル画面を閉じた場合は false を返却します。
data.matterNumber	文字列	案件番号 処理を実行して正常終了したときに返却します。 処理を実行せずに、×ボタンをクリックしてモーダルを閉じた場合は、返却されません。
data.systemMatterId	文字列	システム案件ID 処理を実行して正常終了したときに返却します。 処理を実行せずに、×ボタンをクリックしてモーダルを閉じた場合は、返却されません。
data.userDataId	文字列	ユーザデータID 処理を実行して正常終了したときに返却します。 処理を実行せずに、×ボタンをクリックしてモーダルを閉じた場合は、返却されません。

showConfirm

imWorkflow.modal.showConfirm(Object parameter) : Promise<Object>

確認モーダルを表示します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - authUserDepartmentInfo
 - interfaceControl
- 返却値

サンプルコード

```
imWorkflow.modal.showConfirm({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    authUserDepartmentInfo: {},
    confirmComment: "",
    interfaceControl: {}
  },
  optionalParameter: {},
  rebootModal: false
})
.then(function(result) {
  // modal close
  if (result.isProcessDone) {
    // success
    alert("done");
  }
  imWorkflow.transition.afterProcess();
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object（下記参照）
確認モーダル表示用パラメータ情報オブジェクト
- parameter.rebootModal : boolean
モーダルを閉じた際に入力した内容を保持するかを指定
true: モーダルを閉じる度に入力内容を破棄し、再度このメソッドを実行した際はモーダル表示処理を再実行します。
false: モーダルを閉じても入力内容はコンテンツ画面に保持され、再度このメソッドを実行した際は保持された情報を表示します。画面遷移をした場合、入力内容は破棄されます。（初期値）

processParameter

項目名	必須/任意	型（最大バイト数）	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 担当組織に初期値を設定したい場合に指定
confirmComment	任意	文字列（2000）	確認コメント 確認コメントに初期値を設定したい場合に指定
interfaceControl	任意	下記参照	インタフェース制御 モーダル画面のインタフェースを変更したい場合に指定

authUserDepartmentInfo

```
{
  authUserDepartmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

interfaceControl

確認モーダルの各項目に対するインタフェースの制御（表示、参照、必須）を指定します。
制御の内容と設定値ごとの動作は以下の通りです。

- 表示（display）
 - true : 項目を表示します。（初期値）
 - false : 項目を非表示にします。

- 参照 (readonly)
 - true : 項目を入力不可 (参照のみ) にします。
 - false : 項目を入力可とします。(初期値)
- 必須 (required)
 - true : 項目を必須入力にします。
 - false : 項目を任意入力にします。(初期値)

項目ごとに制御できる内容は異なります。

```
{
  interfaceControl: {
    matterNumber: {
      display: true
    },
    matterName: {
      display: true
    },
    applyBaseDate: {
      display: true
    },
    applyDate: {
      display: true
    },
    applyAuthUserCd: {
      display: true
    },
    authUserDepartmentInfo: {
      display: true,
      readonly: false
    },
    priorityLevel: {
      display: true
    },
    confirmComment: {
      display: true,
      readonly: false,
      required: false
    }
  }
}
```

項目名	必須/任意	型	説明
matterNumber	任意	オブジェクト	案件番号に対するインタフェース制御を指定します。 「display」を設定できます。
matterName	任意	オブジェクト	案件名に対するインタフェース制御を指定します。 「display」を設定できます。
applyBaseDate	任意	オブジェクト	申請基準日に対するインタフェース制御を指定します。 「display」を設定できます。
applyDate	任意	オブジェクト	申請日に対するインタフェース制御を指定します。 「display」を設定できます。
applyAuthUserCd	任意	オブジェクト	申請権限者に対するインタフェース制御を指定します。 「display」を設定できます。
authUserDepartmentInfo	任意	オブジェクト	担当組織に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。
priorityLevel	任意	オブジェクト	優先度に対するインタフェース制御を指定します。 「display」を設定できます。
confirmComment	任意	オブジェクト	処理コメントに対するインタフェース制御を指定します。 「display」と「readonly」と「required」を設定できます。

返却値

- Promise<Object>
 - 返却オブジェクト (下記参照)
- 当APIはPromiseを返します。
 - 正常終了時はpromise.then()で返却値を受け取ってください。

項目名	型	説明
isProcessDone	boolean	処理実行フラグ 処理を実行し、モーダル画面が閉じた場合に true を返却します。 処理を実行せずに、×ボタン押下でモーダル画面を閉じた場合は false を返却します。

showProcess

imWorkflow.modal.showProcess(Object parameter) : Promise<Object>

処理モーダルを表示します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - 処理種別
 - authUserDepartmentInfo
 - branchSelects
 - dynamicNodeConfigs／confirmNodeConfigs
 - horizontalNodeConfigs／verticalNodeConfigs
 - interfaceControl
 - optionalParameter
- 返却値

サンプルコード

```
imWorkflow.modal.showProcess({
  processParameter: {
    systemMatterId: "",
    nodeId: "",
    processType: [],
    matterName: "",
    authUserDepartmentInfo: {},
    priorityLevel: "",
    processComment: "",
    branchSelects: [],
    sendBackNodeIds: [],
    dynamicNodeConfigs: [],
    confirmNodeConfigs: [],
    horizontalNodeConfigs: [],
    verticalNodeConfigs: [],
    interfaceControl: {}
  },
  optionalParameter: {},
  rebootModal: false
})
.then(function(result) {
  // modal close
  if (result.isProcessDone) {
    // success
    alert("done");
  }
  imWorkflow.transition.afterProcess();
});
```

パラメータ

- parameter : Object
- parameter.processParameter : Object（下記参照）
処理モーダル表示用パラメータ情報オブジェクト
- parameter.optionalParameter : Object（下記参照）
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト
- parameter.rebootModal : boolean
モーダルを閉じた際に入力した内容を保持するかを指定
true: モーダルを閉じる度に入力内容を破棄し、再度このメソッドを実行した際はモーダル表示処理を再実行します。
false: モーダルを閉じても入力内容はコンテンツ画面に保持され、再度このメソッドを実行した際は保持された情報を表示します。画面遷移をした場合、入力内容は破棄されます。（初期値）

processParameter

項目名	必須/任意	型（最大バイト数）	説明
systemMatterId	任意	文字列	システム案件ID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのシステム案件IDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須
nodeId	任意	文字列	ノードID 「案件一覧」画面から遷移した場合は、一覧で選択したタスクのノードIDが設定されるため不要 「案件一覧」画面以外から遷移した場合は、必須

項目名	必須/任意	型 (最大バイト数)	説明
processType	任意	下記参照	処理種別 処理種別に対する制御をしたい場合に設定
matterName	任意	文字列(200)	案件名 案件名に初期値を設定したい場合に指定 再申請、起票案件申請の場合に有効
authUserDepartmentInfo	任意	下記参照	権限者所属組織情報 担当組織に初期値を設定したい場合に指定
priorityLevel	任意	文字列	優先度 優先度に初期値を設定したい場合に指定 再申請、起票案件申請の場合に有効 設定値は「CodeList」を参照
processComment	任意	文字列 (2000)	処理コメント 処理コメントに初期値を設定したい場合に指定
branchSelects	任意	下記参照	分岐先選択情報 ルート選択に初期値を設定したい場合に指定
sendBackNodeIds	任意	配列 (文字列)	差戻し先ノードID 差戻し先の初期値を設定したい場合に指定
dynamicNodeConfigs	任意	下記参照	動的承認ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
confirmNodeConfigs	任意	下記参照	確認ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
horizontalNodeConfigs	任意	下記参照	横配置ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
verticalNodeConfigs	任意	下記参照	縦配置ノード設定情報 「動的処理対象者設定機能」を利用する場合に指定
interfaceControl	任意	下記参照	インタフェース制御 モーダル画面のインタフェースを変更したい場合に指定

処理種別

```
{
  "processType": {
    "disableTypes": [
      "string"
    ],
    "selected": "string"
  }
}
```

項目名	必須/任意	型	説明
disableTypes	任意	配列 (文字列)	無効にする処理種別を指定します。
selected	任意	文字列	モーダル画面を表示したときに初期選択する処理種別を指定します。



コラム

「disableTypes」、「selected」に指定できる処理種別の値は以下の通りです。

- 起票: “apply_from_unapply”
- 再申請: “reapply”
- 取止め: “discontinue”
- 承認: “approve”
- 承認終了: “approve_end”
- 否認: “deny”
- 保留: “reserve”
- 保留解除: “reserve_cancel”
- 差戻し: “sendback”

authUserDepartmentInfo

```
{
  authUserDepartmentInfo: {
    companyCd: "string",
    departmentSetCd: "string",
    departmentCd: "string"
  }
}
```

項目名	必須/任意	型	説明
companyCd	必須	文字列	会社コードを指定します。
departmentSetCd	必須	文字列	組織セットコードを指定します。
departmentCd	必須	文字列	組織コードを指定します。

branchSelects

```
{
  branchSelects: [
    {
      branchStartNodeId: "string",
      forwardNodeIds: [
        "string"
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
branchStartNodeId	必須	文字列	初期値を指定するルート選択の分岐開始ノードのノードIDを指定します。
forwardNodeIds	必須	配列（文字列）	初期値として選択するノードのノードIDを配列で指定します。 単一選択の場合は1つ、複数選択の場合は選択するすべてのノードのノードIDを設定してください。

dynamicNodeConfigs / confirmNodeConfigs

```
{
  dynamicNodeConfigs: [
    {
      enable: true,
      nodeId: "string",
      nodeInformation: "string",
      readonlyEnableToggle: false,
      searchCondition: {
        criteria: "any"
      },
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

```
{
  confirmNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      searchCondition: {
        criteria: "any"
      },
      processTargetConfigs: [
        {
          pluginId: "string",
          parameter: "string"
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型	説明
-----	-------	---	----

項目名	必須/任意	型	説明
enable	任意	boolean	<p>「フロー設定」においてノードを設定するか否かの初期値を制御します。</p> <p>この設定は、動的承認ノードに対してのみ有効です。</p> <p>フロー定義における動的承認ノードの削除設定が「削除可能」な場合のみ有効な項目です。</p> <p>true の場合、処理対象者を設定する状態が表示します。</p> <p>false の場合、ノードを削除する状態が表示します。</p> <p>「interfaceControl.display」と組み合わせて、両方にfalseを設定した場合、該当の動的承認ノードを削除（ノードスキップ）できます。</p> <p>未指定の場合、trueとして動作します。</p>
nodeId	必須	文字列	設定対象のノードIDを指定します。
nodeInformation	任意	文字列	ノードに対する任意のメッセージを指定します。
readOnlyEnableToggle	任意	boolean	<p>「フロー設定」において「ノード有効可否トグル」の状態変更可否を制御します。</p> <p>この設定は、動的承認ノードに対してのみ有効です。</p> <p>フロー定義における動的承認ノードの削除設定が「削除可能」な場合のみ有効な項目です。</p> <p>true の場合、「ノード有効可否トグル」を状態変更不可の状態を表示します。</p> <p>false の場合、「ノード有効可否トグル」を状態変更可能の状態を表示します。</p> <p>「interfaceControl.readonly」=true と組み合わせて指定した場合、当該項目の指定は無視されます。</p> <p>未指定の場合、falseとして動作します。</p>
searchCondition	任意	オブジェクト	<p>処理対象者の検索時条件を指定します。</p> <p>「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。</p> <p>当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索（キーワード タブ）」のみです。</p>
criteria	任意	オブジェクト	<p>ユーザ検索（キーワード タブ）に対する暗黙条件を指定可能です。</p> <p>具体的には、「IM-共通マスタ 検索画面起動引数一覧」において以下に該当する引数を指定可能です。</p> <ul style="list-style-type: none"> ■ 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 ■ 対象となる引数 分類 = 「暗黙条件」 <p>「searchCondition」を指定した場合、省略することはできません。</p> <p>「searchCondition」・「criteria」の両方が省略された場合、検索条件（暗黙条件）指定なしで動作します。</p>
processTargetConfigs	任意	オブジェクト	<p>処理対象プラグイン情報を指定します。</p> <p>利用可能なプラグインIDについては「処理権限者プラグイン一覧」を参照してください。</p> <p>未指定の場合、現在設定済みの処理対象者が適用されます。</p>



コラム

「searchCondition」、または、「processTargetConfigs」を設定した場合、保存済み設定機能は利用できません。



コラム

ノードの削除可否や利用可能なプラグインは、[動的承認](#)・[横配置](#)・[縦配置](#)ノード設定の範囲で指定できます。

[horizontalNodeConfigs](#)／[verticalNodeConfigs](#)

```
{
  horizontalNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      dispatchControl: {
        max: "number",
        min: "number"
      },
      searchCondition: {
        criteria: "string"
      },
      matterNodeExpansions: [
        {
          nodeName: "string",
          searchCondition: {
            criteria: "any"
          },
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

```
{
  verticalNodeConfigs: [
    {
      nodeId: "string",
      nodeInformation: "string",
      dispatchControl: {
        max: "number",
        min: "number"
      },
      searchCondition: {
        criteria: "string"
      },
      matterNodeExpansions: [
        {
          nodeName: "string",
          searchCondition: {
            criteria: "any"
          },
          processTargetConfigs: [
            {
              pluginId: "string",
              parameter: "string"
            }
          ]
        }
      ]
    }
  ]
}
```

項目名	必須/任意	型 (最大バイト数)	説明
nodeId	必須	文字列	設定対象のノードIDを指定します。
nodeInformation	任意	文字列	ノードに対する任意のメッセージを指定します。
dispatchControl	任意	オブジェクト	割当可能ノード数を指定します。 当プロパティによる制御は、フロー定義で設定済みの割当可能ノード数の範囲内においてのみ有効です。 「max」プロパティを設定することで、割当可能ノード数 (最大) を指定可能です。 「min」プロパティを設定することで、割当可能ノード数 (最小) を指定可能です。
searchCondition	任意	オブジェクト	処理対象者の検索時条件を指定します。 「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。 当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索 (キーワード タブ)」のみです。

項目名	必須/任意	型 (最大バイト数)	説明
criteria	任意	オブジェクト	<p>ユーザ検索 (キーワード タブ) に対する暗黙条件を指定可能です。 具体的には、「IM-共通マスタ 検索画面起動引数一覧」において以下に該当する引数を指定可能です。</p> <ul style="list-style-type: none"> 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 対象となる引数 分類 = 「暗黙条件」 <p>「searchCondition」を指定した場合、省略することはできません。 「searchCondition」・「criteria」の両方が省略された場合、検索条件 (暗黙条件) 指定なしで動作します。</p>
matterNodeExpansions	任意	オブジェクト	<p>ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。</p> <p>「interfaceControl.display」が false (表示しない) の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最小」「最大」値の範囲で、当パラメータで定義したノード数で展開されます。 <p>「interfaceControl.display」が true (表示する) の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> 展開可能なノード数の最小個数は、次の値のどちらか「大きいほう」を採用します。 <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最小」値 dispatchControl.min で指定された値 (有効な値を指定している場合のみ考慮します) 上記ルールで決定した最小個数より当パラメータで指定したノード数の方が少ない場合、処理を行うためには画面上での追加設定が必要です 展開可能なノード数の最大個数は、次の値のどちらか「小さいほう」を採用します。 <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最大」値 dispatchControl.max で指定された値 (有効な値を指定している場合のみ考慮します) 上記ルールで決定した最大個数より当パラメータで指定したノード数の方が多い場合、最大個数を超えた分は設定されません。 <p>対象のノードを削除 (ノードスキップ) する場合、以下の設定を行ってください。</p> <ul style="list-style-type: none"> 「matterNodeExpansions」の設定を省略または空で指定、かつ、displayにfalseを指定。
nodeName	任意	文字列 (50)	ノード名を指定します。
processTargetConfigs	任意	オブジェクト	<p>処理対象プラグイン情報を指定します。 利用可能なプラグインIDについては「処理権限者プラグイン一覧」を参照してください。 未指定の場合、現在設定済みの処理対象者が適用されます。</p>



コラム

「searchCondition」、または、「matterNodeExpansions」を設定した場合、保存済み設定機能は利用できません。



コラム

割当可能ノード数や利用可能なプラグインは、[動的承認](#)・[横配置](#)・[縦配置ノード設定](#) の範囲で指定できます。

interfaceControl

処理モジュールの各項目に対するインタフェースの制御 (表示、参照、必須) を指定します。
 制御の内容と設定値ごとの動作は以下の通りです。

- 表示 (display)
 - true : 項目を表示します。(初期値)
 - false : 項目を非表示にします。
- 参照 (readonly)
 - true : 項目を入力不可 (参照のみ) にします。
 - false : 項目を入力可とします。(初期値)

- 必須 (required)
 - true : 項目を必須入力にします。
 - false : 項目を任意入力にします。(初期値)

項目ごとに制御できる内容は異なります。

```
{
  interfaceControl: {
    processType: {
      display: true
    },
    matterNumber: {
      display: true
    },
    matterName: {
      display: true,
      readonly: false
    },
    applyBaseDate: {
      display: true
    },
    applyDate: {
      display: true
    },
    applyAuthUserCd: {
      display: true
    },
    authUserCd: {
      display: true
    },
    authUserDepartmentInfo: {
      display: true,
      readonly: false
    },
    priorityLevel: {
      display: true,
      readonly: false
    },
    processComment: {
      display: true,
      readonly: false,
      required: false
    },
    attachmentFile: {
      display: true
    },
    branchSelects: [
      {
        branchStartNodeId: "string",
        display: true,
        readonly: false
      }
    ],
    sendBack: {
      readonly: true
    },
    dynamicNodeConfigs: [
      {
        nodeId: "string",
        display: true,
        readonly: false
      }
    ],
    confirmNodeConfigs: [
      {
        nodeId: "string",
        display: true,
        readonly: false
      }
    ],
    horizontalNodeConfigs: [
      {
        nodeId: "string",
        display: true,
        readonly: false
      }
    ],
    verticalNodeConfigs: [
      {
        nodeId: "string",
        display: true,
        readonly: false
      }
    ]
  }
}
```

```

],
stamp: {
  display: true
},
nego: {
  display: true
}
}
}
}

```

項目名	必須/任意	型	説明
processType	任意	オブジェクト	処理種別に対するインタフェース制御を指定します。 「display」を設定できます。
matterNumber	任意	オブジェクト	案件番号に対するインタフェース制御を指定します。 「display」を設定できます。
matterName	任意	オブジェクト	案件名に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。 「readonly」は再申請、起票案件申請の場合に有効です。
applyBaseDate	任意	オブジェクト	申請基準日に対するインタフェース制御を指定します。 「display」を設定できます。
applyDate	任意	オブジェクト	申請日に対するインタフェース制御を指定します。 「display」を設定できます。
applyAuthUserCd	任意	オブジェクト	申請権限者に対するインタフェース制御を指定します。 「display」を設定できます。
authUserCd	任意	オブジェクト	処理者に対するインタフェース制御を指定します。 「display」を設定できます。
authUserDepartmentInfo	任意	オブジェクト	担当組織に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。
priorityLevel	任意	オブジェクト	優先度に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。 「readonly」は再申請、起票案件申請の場合に有効です。
processComment	任意	オブジェクト	処理コメントに対するインタフェース制御を指定します。 「display」と「readonly」と「required」を設定できます。
attachmentFile	任意	オブジェクト	添付ファイルに対するインタフェース制御を指定します。 「display」を設定できます。
branchSelects	任意	配列 (オブジェクト)	分岐先選択のインタフェース制御を指定します。 分岐先選択では、分岐先となる分岐先開始ノード (branchStartNodeId) ごとに 「display」、「readonly」を設定できます。
sendBack	任意	オブジェクト	差戻しに対するインタフェース制御を指定します。 「readonly」を設定できます。
dynamicNodeConfigs	任意	配列 (オブジェクト)	動的承認ノードのインタフェース制御を指定します。 動的承認ノード (nodeId) ごとに「display」、「readonly」を設定できます。
confirmNodeConfigs	任意	配列 (オブジェクト)	確認ノードのインタフェース制御を指定します。 確認ノード (nodeId) ごとに「display」、「readonly」を設定できます。
horizontalNodeConfigs	任意	配列 (オブジェクト)	横配置ノードのインタフェース制御を指定します。 横配置ノード (nodeId) ごとに「display」、「readonly」を設定できます。
verticalNodeConfigs	任意	配列 (オブジェクト)	縦配置ノードのインタフェース制御を指定します。 縦配置ノード (nodeId) ごとに「display」、「readonly」を設定できます。
stamp	任意	オブジェクト	印影に対するインタフェース制御を指定します。 「display」を設定できます。
nego	任意	オブジェクト	根回しメールに対するインタフェース制御を指定します。 「display」を設定できます。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、imwStampId を指定してください。
印影を利用して、imwStampId を指定しない場合、常用印が利用されます。



コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: userParameter
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: imwUserParameter

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
正常終了時はpromise.then()で返却値を受け取ってください。

項目名	型	説明
isProcessDone	boolean	処理実行フラグ 処理を実行し、モーダル画面が閉じた場合に true を返却します。 処理を実行せずに、×ボタン押下でモーダル画面を閉じた場合は false を返却します。

showTemporarySave

imWorkflow.modal.showTemporarySave(Object parameter) : Promise<Object>

一時保存モーダルを表示します。

Contents

- サンプルコード
- パラメータ
 - processParameter
 - interfaceControl
 - optionalParameter
- 返却値

サンプルコード

```

imWorkflow.modal.showTemporarySave({
  processParameter: {
    userDataId: "",
    flowId: "",
    matterName: "",
    applyBaseDate: "",
    applyAuthUserCd: "",
    processComment: "",
    interfaceControl: {}
  },
  optionalParameter: {},
  rebootModal: false
})
.then(function(result) {
  // modal close
  if (result.isProcessDone) {
    // success
    alert("done");
  }
  imWorkflow.transition.afterProcess();
});

```

パラメータ

- parameter : Object
- parameter.processParameter : Object (下記参照)
一時保存用パラメータ情報オブジェクト
- parameter.optionalParameter : Object (下記参照)
ユーザパラメータ、および、Forma用ユーザパラメータを内包するオブジェクト
- parameter.rebootModal : boolean
モーダルを閉じた際に入力した内容を保持するかを指定
true: モーダルを閉じる度に入力内容を破棄し、再度このメソッドを実行した際はモーダル表示処理を再実行します。
false: モーダルを閉じて入力内容はコンテンツ画面に保持され、再度このメソッドを実行した際は保持された情報を表示します。画面遷移をした場合、入力内容は破棄されます。(初期値)

processParameter

項目名	必須/任意	型 (最大バイト数)	説明
userDataId	任意	文字列 (20)	ユーザデータID 「案件一覧」画面から遷移した場合は、一覧で選択した一時保存情報のユーザデータIDが設定されるため不要 未指定の場合、Identifier.get()で一時保存情報を新規作成 指定した場合、当該キーで特定される一時保存情報が存在しない場合、一時保存情報を新規作成 当該キーで特定される一時保存情報が存在する場合、一時保存情報を更新
flowId	任意	文字列	フローID 一時保存情報を新規作成時は必須 一時保存情報を更新時は不要
matterName	任意	文字列 (200)	案件名 案件名に初期値を設定したい場合に指定
applyBaseDate	任意	日付/文字列	申請基準日 未指定の場合、「申請一覧」「案件一覧 (一時保存)」画面から遷移した場合は、一覧で選択した申請基準日が設定される 「申請一覧」「案件一覧 (一時保存)」画面以外から遷移した場合は、アカウントコンテキストのタイムゾーンにおける現在日付で一時保存 “yyyy/MM/dd”形式の文字列指定も可能
applyAuthUserCd	任意	文字列	申請権限者コード 申請権限者に初期値を設定したい場合に指定
processComment	任意	文字列 (2000)	処理コメント 処理コメントに初期値を設定したい場合に指定
interfaceControl	任意	下記参照	インタフェース制御 モーダル画面のインタフェースを変更したい場合に指定

interfaceControl

一時保存モーダルの各項目に対するインタフェースの制御 (表示、参照、必須) を指定します。
制御の内容と設定値ごとの動作は以下の通りです。

- 表示 (display)
 - true : 項目を表示します。(初期値)
 - false : 項目を非表示にします。
- 参照 (readonly)

- true : 項目を入力不可 (参照のみ) にします。
- false : 項目を入力可とします。(初期値)
- 必須 (required)
 - true : 項目を必須入力にします。
 - false : 項目を任意入力にします。(初期値)

項目ごとに制御できる内容は異なります。

```
{
  interfaceControl: {
    matterName: {
      display: true,
      readonly: false
    },
    applyBaseDate: {
      display: true
    },
    applyAuthUserCd: {
      display: true
    },
    processComment: {
      display: true,
      readonly: false,
      required: false
    }
  }
}
```

項目名	必須/任意	型	説明
matterName	任意	オブジェクト	案件名に対するインタフェース制御を指定します。 「display」と「readonly」を設定できます。
applyBaseDate	任意	オブジェクト	申請基準日に対するインタフェース制御を指定します。 「display」を設定できます。
applyAuthUserCd	任意	オブジェクト	申請権限者に対するインタフェース制御を指定します。 「display」を設定できます。
processComment	任意	オブジェクト	処理コメントに対するインタフェース制御を指定します。 「display」と「readonly」と「required」を設定できます。

optionalParameter

```
{
  optionalParameter: {
    userParameter: {
      imwStampId: "string",
      xxxx: "string"
    },
    formaParam: {
      items: {
        textbox1: "string"
      }
    }
  }
}
```

userParameter

パラメータ「userParameter」の主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。アクション処理に受け渡したいパラメータを指定してください。



コラム

印影を指定する場合、`imwStampId` を指定してください。
印影を利用して、`imwStampId` を指定しない場合、常用印が利用されます。

i コラム

「optionalParameter.userParameter」のアクション処理へのマッピングについて

IM-Workflow のアクション処理には、スクラッチ開発によるプログラムや、IM-LogicDesigner のロジックフローを利用できます。「optionalParameter.userParameter」は、これらのアクション処理の下記パラメータにマッピングされます。

- スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合
 - 「IM-Workflow プログラミングガイド」-「ユーザプログラムの作成 - アクション処理」
アクション処理パラメータにおける「ユーザパラメータ」: `userParameter`
 - IM-LogicDesigner のロジックフローをアクション処理として利用している場合
 - 「IM-Workflow 管理者操作ガイド」-「ロジックフローの入出力設定 - アクション処理」
ロジックフロー入力設定における「ユーザパラメータ」: `imwUserParameter`

formaParam

パラメータ「formaParam」は、IM-FormaDesigner / IM-BIS で作成したワークフローに対して、アプリケーションデータを登録したい場合に指定が必要です。詳細は以下のリンクを参照してください。

- 「IM-Workflow 仕様書」-「IM-FormaDesigner / IM-BIS 連携フローの追加処理」

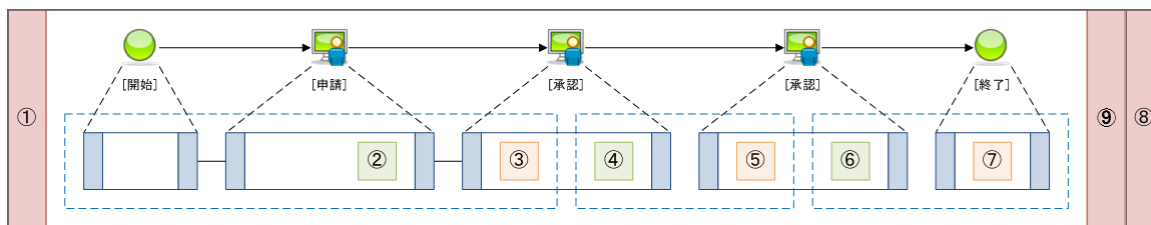
返却値

- Promise<Object>
返却オブジェクト（下記参照）
- 当APIはPromiseを返します。
正常終了時はpromise.then()で返却値を受け取ってください。

項目名	型	説明
isProcessDone	boolean	処理実行フラグ 処理を実行し、モーダル画面が閉じた場合に true を返却します。 処理を実行せずに、×ボタン押下でモーダル画面を閉じた場合は false を返却します。
data.userDataId	文字列	ユーザデータID 処理を実行して正常終了したときに返却します。 処理を実行せずに、×ボタンをクリックしてモーダルを閉じた場合は、返却されません。

i コラム

ワークフロー処理モーダルとは、ワークフロー処理を実行するためのモーダル画面のことです。2020 Summer(Zephyrine) より利用できます。これまでの標準処理画面からデザインを刷新し、開発モデルによらず利用できるようにCSJS APIとして動作します。「workflowOpenPage」タグは、これまで通り動作しますが、今後機能追加の予定はありません。ワークフロー処理モーダルを利用した画面実装を推奨します。



No	処理名	項番
1	案件開始処理	①
2	案件終了処理	⑧
3	アクション処理	② ④ ⑥
4	到達処理	③ ⑤ ⑦
5	案件終了処理（トランザクションなし）	⑨

案件開始処理

Contents

- パラメータ
- 返却値

案件開始処理とは、案件が開始する時に、一度実行される処理です。下記の場合に実行されます。

- 申請者が“申請”を行った場合
- “起票”の案件を作成した場合（APIのみ）

案件開始処理は、IM-Workflow モジュールのトランザクション内で実行されるため、このプログラム中ではDBトランザクション制御を行うことはできません。

パラメータ

案件開始処理では、以下の情報をパラメータとして受け取る事ができます。

No	パラメータ（物理名）	パラメータ（論理名）	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日（“yyyy/MM/dd”）
11	processDate	処理日	文字列	処理日（“yyyy/MM/dd”）
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス

返却値

案件開始処理では、以下の情報を返却します。（任意/必須・・・●：必須 △：任意）

- スクリプト開発モデル

No	返却値（物理名）	返却値（論理名）	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true : 成功 false : 失敗 指定しない場合、成功（true）として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。

※ 結果メッセージを設定した場合、画面にエラーメッセージを表示します。



コラム

結果フラグとして“false”（失敗）を設定した場合の動作は以下の通りです。

- 該当の案件開始処理以外に設定されている後続の案件開始処理は実行されません。

- JavaEE開発モデル

返却値はありません。

※ 「`jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException`」をスローした場合、引数に設定したエラーメッセージを画面に表示します。詳細は「[APIドキュメント](#)」を参照してください。



コラム

`WorkflowExternalException` をスローした場合の動作は以下の通りです。

- 該当の案件開始処理以外に設定されている後続の案件開始処理は実行されません。

案件終了処理

Contents

- [パラメータ](#)
- [返却値](#)

案件終了処理とは、案件が終了する時に、一度実行される処理です。下記の場合に実行されます。

- 最後の承認者が“承認”を行った場合
- 承認者が“承認終了”を行った場合
- 承認者が“否認”を行った場合
- 申請者が“取止め”を行った場合
- 案件操作で終了ノードに到達した場合

案件終了処理は、直前のアクション処理や到達処理とは独立した処理（トランザクション）です。そのため、案件終了処理でエラーが発生した場合、直前の処理を戻すこと（ロールバック）はできません。

案件終了処理は、IM-Workflow モジュールのトランザクション内で実行されるため、このプログラム中ではDBトランザクション制御を行うことはできません。

パラメータ

案件終了処理では、以下の情報をパラメータとして受け取る事ができます。

No	パラメータ（物理名）	パラメータ（論理名）	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日 (“yyyy/MM/dd”)
11	processDate	処理日	文字列	処理日 (“yyyy/MM/dd”)
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID

No	パラメータ (物理名)	パラメータ (論理名)	属性	詳細
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス
15	actFlag	代理フラグ	文字列	"0": 本人にて処理 "1": 代理者にて処理
16	lastProcessNodeId	最終処理ノードID	文字列	最終処理のノードID
17	lastAuthUserCd	最終処理権限者コード	文字列	最終処理の処理権限者コード
18	lastExecUserCd	最終処理実行者コード	文字列	最終処理の処理実行者コード
19	lastResultStatus	最終処理結果ステータス	文字列	最終処理の処理結果ステータス [1]
20	mailIds	メールテンプレートID	文字列[]	メール種別「処理結果通知」で設定されているメールID
21	imBoxIds	ImBoxId	文字列[]	IMBox種別「処理結果通知」で設定されているIMBoxID
22	mailReplaceMap	メール置換文字情報	Map	メールの置換文字情報 [2]
23	imBoxReplaceMap	IMBox置換文字情報	Map	IMBoxの置換文字情報 [2]

[1] ……コード値は、「APIドキュメントのIM-Workflow CodeList」を参照してください。

[2] (1, 2) ……置換文字列と置換内容の詳細は、「IM-Workflow 仕様書 別紙」を参照してください。

返却値

案件終了処理では、以下の情報を返却します。(任意/必須 ……●: 必須 △: 任意)

■ スクリプト開発モデル

No	返却値 (物理名)	返却値 (論理名)	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true: 成功 false: 失敗 指定しない場合、成功 (true) として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。
3	data	メール送信可否	真偽値	●	このパラメータは「処理結果通知」の送信制御を行います。 このパラメータによる制御はメールに限らず、IMBox などにも適用されます。 true: 送信する false: 送信しない

※ 結果メッセージを設定した場合、例外ログにエラーメッセージを出力します。



コラム

結果フラグとして"false" (失敗) を設定した場合の動作は以下の通りです。

- 該当の案件終了処理以外に設定されている後続の案件終了処理は実行されません。
- 処理結果通知を設定している場合、「メール送信可否」の設定値に関係なく、処理結果通知による通知は行われません。

■ JavaEE開発モデル

No	返却値 (論理名)	属性	必須/任意	詳細
1	メール送信可否	真偽値	●	このパラメータは「処理結果通知」の送信制御を行います。 このパラメータによる制御はメールに限らず、IMBox などにも適用されます。 true: 送信する false: 送信しない

※ 「jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException」をスローした場合、引数に設定したエラーメッセージを例外ログに出力します。詳細は「APIドキュメント」を参照してください。



コラム

WorkflowExternalException をスローした場合の動作は以下の通りです。

- 該当の案件終了処理以外に設定されている後続の案件終了処理は実行されません。
- 処理結果通知を設定している場合、「メール送信可否」の設定値に関係なく、処理結果通知による通知は行われません。

アクション処理

Contents

- パラメータ
- 返却値

アクション処理とは、下記のような行を行った場合に実行される処理です。

No	アクション	メソッド
1	申請	apply
2	再申請	reapply
3	申請（一時保存）	applyFromTempSave
4	申請（未処理）	applyFromUnapply
5	取止め	discontinue
6	引戻し	pullBack
7	差戻し後引戻し	sendBackToPullBack
8	承認	approve
9	承認終了	approveEnd
10	否認	deny
11	差戻し	sendBack
12	保留	reserve
13	保留解除	reserveCancel
14	案件操作	matterHandle
15	一時保存（新規登録）	tempSaveCreate
16	一時保存（更新）	tempSaveUpdate
17	一時保存（削除）	tempSaveDelete

アクション処理は、IM-Workflow モジュールのトランザクション内で実行されるため、このプログラム中ではDBトランザクション制御を行うことはできません。

パラメータ

アクション処理では、以下の情報をパラメータとして受け取る事ができます。

- アクション処理パラメータ
- ユーザパラメータ

アクション処理パラメータ

No	パラメータ（物理名）	パラメータ（論理名）	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日（"yyyy/MM/dd"）
11	processDate	処理日	文字列	処理日（"yyyy/MM/dd"）
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	matterName	案件名	文字列	該当案件の案件名
15	matterNumber	案件番号	文字列	該当案件の案件番号

No	パラメータ (物理名)	パラメータ (論理名)	属性	詳細
16	priorityLevel	優先度	文字列	該当処理の優先度 ※1
17	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス
18	actFlag	代理フラグ	文字列	"0": 本人にて処理 "1": 代理者にて処理
19	nodeId	ノードID	文字列	該当処理のノードID
20	nextNodeIds	移動先 (次ノード) ノードID	文字列[]	以下処理種別の場合に、移動先ノードIDが設定されます。 差戻し/引戻し/案件操作
21	authUserCd	処理権限者コード	文字列	処理権限者コード
22	execUserCd	処理実行者コード	文字列	処理実行者コード
23	resultStatus	処理結果ステータス	文字列	該当処理の処理結果ステータス [1]
24	authCompanyCode	権限会社コード	文字列	以下処理種別の場合、権限会社コードが設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) 取止め/承認/承認終了/否認/差戻し
25	authOrgzSetCode	権限組織セットコード	文字列	以下処理種別の場合、権限組織セットコードが設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) 取止め/承認/承認終了/否認/差戻し
26	authOrgzCode	権限組織コード	文字列	以下処理種別の場合、権限組織コードが設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) 取止め/承認/承認終了/否認/差戻し
27	processComment	処理コメント	文字列	該当処理の処理コメント
28	lumpProcessFlag	一括処理フラグ	文字列	"0": 通常承認 "1": 一括承認
29	autoProcessFlag	自動処理フラグ	文字列	"0": 手動処理 "1": 自動処理 (到達処理での自動承認や、バッチでの自動処理)
30	DCNodeConfigModels	動的・確認ノード設定情報オブジェクトの配列	オブジェクト[]	以下処理種別の場合に、処理時の標準処理画面や処理モーダルで入力した動的・確認ノード設定情報が設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) /承認 オブジェクトの詳細は以下のドキュメントを参照してください <ul style="list-style-type: none"> ■ 「IM-Workflow APIドキュメント (JavaEE開発モデル)」 - 「DynamicAndCnfmNodeConfigModel」 ■ 「IM-Workflow APIドキュメント (スクリプト開発モデル)」 - 「DynamicAndCnfmNodeConfigInfo オブジェクト」
31	HVNodeConfigModels	横配置・縦配置ノード設定情報オブジェクトの配列	オブジェクト[]	以下処理種別の場合に、処理時の標準処理画面や処理モーダルで入力した横配置・縦配置ノード設定情報が設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) /承認 オブジェクトの詳細は以下のドキュメントを参照してください <ul style="list-style-type: none"> ■ 「IM-Workflow APIドキュメント (JavaEE開発モデル)」 - 「HorizontalAndVerticalNodeConfigModel」 ■ 「IM-Workflow APIドキュメント (スクリプト開発モデル)」 - 「HorizontalAndVerticalNodeConfigInfo オブジェクト」
32	branchSelectModels	分岐先選択情報オブジェクトの配列	オブジェクト[]	以下処理種別の場合に、処理時の標準処理画面や処理モーダルで入力した分岐先選択情報が設定されます。 申請/再申請/申請 (一時保存) /申請 (未処理) /承認 オブジェクトの詳細は以下のドキュメントを参照してください <ul style="list-style-type: none"> ■ 「IM-Workflow APIドキュメント (JavaEE開発モデル)」 - 「BranchSelectModel」 ■ 「IM-Workflow APIドキュメント (スクリプト開発モデル)」 - 「BranchSelectInfo オブジェクト」

[1] ...コード値は、「APIドキュメントのIM-Workflow CodeList」を参照してください。



コラム

パラメータ「DCNodeConfigModels」「HVNodeConfigModels」「branchSelectModels」は 2023 Autumn(Hollyhock) で追加されました。

ユーザパラメータ

パラメータ (物理名)	パラメータ (論理名)	属性	詳細
-------------	-------------	----	----

パラメータ（物理名）	パラメータ（論理名）	属性	詳細
userParameter	ユーザパラメータ	Map オブジェクト	アクション処理に受け渡される任意データ

ユーザパラメータの主な用途としては、画面上の入力情報をサーバ側の処理であるアクション処理に渡すことが想定されています。

ユーザパラメータの主なマッピング元は次のとおりです。

- IM-BloomMaker アクション
 - 「IM-Workflow 仕様書」 - 「IM-BloomMakerアクションアイテム」
 - 案件処理のためのモーダル表示、および案件処理のアクションにおける入力値: optionalParameter.userParameter
- API
 - IM-Workflow クライアントサイドAPI
 - 「IM-Workflow プログラミングガイド」 - 「クライアントサイド JavaScript API」
 - モーダル表示、および案件処理を行うための JavaScript API におけるパラメータ「ユーザパラメータ」: optionalParameter.userParameter
 - IM-Workflow API
 - 「IM-Workflow APIドキュメント（JavaEE開発モデル）」
 - 「IM-Workflow APIドキュメント（スクリプト開発モデル）」
 - 案件処理用メソッド（例: ApplyManager#apply）におけるパラメータ「ユーザパラメータ」: userParam
 - IM-Workflow REST API
 - 「IM-Workflow 仕様書」 - 「REST API」
 - 案件処理用メソッドにおけるリクエストボディ: optionalParameter.userParameter
 - IM-BIS API
 - 「IM-BIS for Accel Platform APIドキュメント」
 - 案件処理用メソッド（例: BisApplyManager#apply）におけるパラメータ「ユーザパラメータ」: userParam



コラム

IM-Workflow が提供する標準機能を利用するうえで、ユーザパラメータに「空」または「Null」が連携される場合があります。

- ユーザパラメータに「Null」が連携される処理
 - 引戻し
 - 差戻し後引戻し
 - 承認（「自動処理」による実行時）
 - 否認（「自動処理」による実行時）
 - 差戻し（「自動処理」による実行時）
 - 案件操作
 - 一時保存（削除）
- ユーザパラメータに「空」が連携される処理
 - スクリプト開発モデルでは「空のオブジェクト」、JavaEE開発モデルでは「空のMap」が連携されます。
 - 承認（「一括処理」による実行時）

上記のケースに該当するかをアクション処理内で判定するには、次の材料を利用してください。

- アクション処理実行の契機となった案件処理の種類
 - アクション処理プログラムにおけるメソッドの種類（スクリプト開発モデル、またはJavaEE開発モデルで開発されたアクション処理の場合）
 - ロジックフロー入力設定における「処理結果ステータス」: resultStatus（IM-LogicDesigner のロジックフローをアクション処理として利用している場合）
- アクション処理パラメータ
 - lumpProcessFlag（一括処理フラグ）
 - autoProcessFlag（自動処理フラグ）

返却値

アクション処理では、以下の情報を返却します。（任意/必須・・・●：必須 △：任意）

- スクリプト開発モデル

No	返却値（物理名）	返却値（論理名）	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true : 成功 false : 失敗 指定しない場合、成功 (true) として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。
3	data	案件番号	文字列 (最大20バイト)	△	申請/再申請/申請（一時保存）/申請（未処理）の場合のみ、null以外の場合に案件番号を上書きします。

※ 結果メッセージを設定した場合、画面にエラーメッセージを表示します。

i コラム

結果フラグとして“false”（失敗）を設定した場合の動作は以下の通りです。

- 該当のアクション処理以外に設定されている後続のアクション処理は実行されません。

- JavaEE開発モデル

No	返却値（論理名）	属性	必須/任意	詳細
1	案件番号	文字列 (最大20バイト)	△	申請/再申請/申請（一時保存）/申請（未処理）の場合のみ、null 以外の場合に案件番号を上書きします。

※ 「jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException」をスローした場合、引数に設定したエラーメッセージを画面に表示します。詳細は「APIドキュメント」を参照してください。

i コラム

WorkflowExternalException をスローした場合の動作は以下の通りです。

- 該当のアクション処理以外に設定されている後続のアクション処理は実行されません。

到達処理

Contents

- パラメータ
- 返却値

到達処理とは、ノードに到達した場合に実行される処理です。

この処理は、アクション処理や IM-Workflow の内部処理とは独立した処理（thread）として実行されます。

そのため、到達処理でエラーが発生した場合、直前の処理を戻すこと（ロールバック）はできません。

（直前のアクション処理とは、トランザクションも別です。）

このプログラム中で、データベースの登録／更新／削除処理を行う場合は、独自にDBトランザクション制御を行ってください。

下記のような場合に実行されます。

- 前ノードの処理者が、“申請”または“承認”を行って到達した場合
- 他のノードから、“差戻し”され到達した場合
- “引戻し”を行って到達した場合
- 案件操作で到達した場合

パラメータ

到達処理では、以下の情報をパラメータとして受け取る事ができます。

No	パラメータ（物理名）	パラメータ（論理名）	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日（“yyyy/MM/dd”）
11	processDate	処理日	文字列	処理日（“yyyy/MM/dd”）
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	matterName	案件名	文字列	該当案件の案件名

No	パラメータ (物理名)	パラメータ (論理名)	属性	詳細
15	matterNumber	案件番号	文字列	該当案件の案件番号
16	priorityLevel	優先度	文字列	該当処理の優先度 ※1
17	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス
18	actFlag	代理フラグ	文字列	"0": 本人にて処理 "1": 代理者にて処理
19	nodeId	ノードID	文字列	該当処理のノードID
20	preNodeId	前ノードID	文字列	前処理のノードID
21	preNodeAuthUserCd	前ノード処理権限者コード	文字列	前処理の処理権限者コード
22	preNodeExecUserCd	前ノード処理実行者コード	文字列	前処理の処理実行者コード
23	preNodeResultStatus	前ノード処理結果ステータス	文字列	前処理の処理結果ステータス [1]
24	preNodeAuthCompanyCode	前ノード権限会社コード	文字列	前処理の権限会社コード
25	preNodeAuthOrgzSetCode	前ノード権限組織セットコード	文字列	前処理の権限組織セットコード
26	preNodeAuthOrgzCode	前ノード権限組織コード	文字列	前処理の権限組織コード
27	preNodeProcessComment	前ノード処理コメント	文字列	前処理の処理コメント
28	mailIds	メールテンプレートID	文字列[]	到達したノードのメール種別「処理依頼」で設定されているメールID
29	imBoxIds	IMBoxId	文字列[]	到達したノードのIMBox種別「処理依頼」で設定されているIMBoxID
30	mailReplaceMap	メール置換文字情報	Map	メールの置換文字情報 [2]
31	imBoxReplaceMap	IMBox置換文字情報	Map	IMBoxの置換文字情報 [2]

[1] ……コード値は、「APIドキュメントのIM-Workflow CodeList」を参照してください。

[2] (1, 2) ……置換文字列と置換内容の詳細は、「IM-Workflow 仕様書 別紙」を参照してください。

返却値

到達処理では、以下の情報を返却します。(任意/必須 ……●: 必須 △: 任意)

■ スクリプト開発モデル

No	返却値 (物理名)	返却値 (論理名)	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true: 成功 false: 失敗 指定しない場合、成功 (true) として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。
3	data	メール送信可否	真偽値	●	このパラメータは「処理依頼」の送信制御を行います。 このパラメータによる制御はメールに限らず、IMBox、IM-Notice などにも適用されます。 true: 送信する false: 送信しない

※ 結果メッセージを設定した場合、例外ログにエラーメッセージを出力します。



コラム

結果フラグとして"false" (失敗) を設定した場合の動作は以下の通りです。

- 該当の到達処理以外に設定されている後続の到達処理は実行されません。
- 処理依頼を設定している場合、「メール送信可否」の設定値に関係なく、処理依頼による通知が行われます。

■ JavaEE開発モデル

No	返却値 (論理名)	属性	必須/任意	詳細
1	メール送信可否	真偽値	●	このパラメータは「処理依頼」の送信制御を行います。 このパラメータによる制御はメールに限らず、IMBox、IM-Notice などにも適用されます。 true: 送信する false: 送信しない

※ 「jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException」をスローした場合、引数に設定したエラーメッセージを例外ログに出力します。詳細は「APIドキュメント」を参照してください。

i コラム

WorkflowExternalException をスローした場合の動作は以下の通りです。

- 該当の到達処理以外に設定されている後続の到達処理は実行されません。
- 処理依頼を設定している場合、「メール送信可否」の設定値に関係なく、処理依頼による通知が行われます。

分岐開始処理

Contents

- パラメータ
- 返却値

分岐開始処理とは、分岐開始ノードで「ユーザプログラムで分岐する」を選択した場合に、実行される処理です。分岐先ノード毎に順番に実行されます。

分岐開始処理は、IM-Workflow モジュールのトランザクション内で実行されるため、このプログラム中ではDBトランザクション制御を行うことはできません。

分岐開始処理において、ルート遷移可否として 遷移する (true) を返却することにより、実行中の分岐開始処理が設定された分岐先ノードに進みます。全ての分岐開始処理のルート遷移可否が 遷移しない (false) の場合は、案件は分岐開始ノードで停止します。このような場合は、案件操作処理で案件を進めてください。

パラメータ

分岐開始処理では、以下の情報をパラメータとして受け取る事ができます。

No	パラメータ (物理名)	パラメータ (論理名)	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日 ("yyyy/MM/dd")
11	processDate	処理日	文字列	処理日 ("yyyy/MM/dd")
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス
15	nodeId	ノードID	文字列	該当処理のノードID

返却値

分岐開始処理では、以下の情報を返却します。(任意/必須・・・●:必須 △:任意)

- スクリプト開発モデル

No	返却値 (物理名)	返却値 (論理名)	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true : 成功 false : 失敗 指定しない場合、成功 (true) として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。
3	data	ルート遷移可否	真偽値	●	[分岐] true : 遷移する false : 遷移しない [結合] true : 結合する false : 結合しない

※ 結果メッセージを設定した場合、画面にエラーメッセージを表示します。

- JavaEE開発モデル

No	返却値（論理名）	属性	必須/任意	詳細
1	ルート遷移可否	真偽値	●	[分岐] true : 遷移する false : 遷移しない [結合] true : 結合する false : 結合しない

※ 「jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException」をスローした場合、引数に設定したエラーメッセージを画面に表示します。詳細は「[APIドキュメント](#)」を参照してください。

分岐終了処理

Contents

- [パラメータ](#)
- [返却値](#)

分岐終了処理とは、分岐終了ノードで「ユーザプログラムで分岐終了する」を選択した場合に、実行される処理です。分岐終了ノードに案件が到達する度に実行されます。

分岐終了処理は、IM-Workflow モジュールのトランザクション内で実行されるため、このプログラム中ではDBトランザクション制御を行うことはできません。

分岐終了処理において、ルート遷移可否として 結合する (true) を返却することにより、未到達のノードを待たずに次のノードに進みます。全てのノードが到達しても結果が全て 結合しない (false) の場合は、案件は分岐終了ノードで停止します。このような場合は、案件操作処理で案件を進めてください。

パラメータ

分岐終了処理のパラメータは、「[分岐開始処理](#)」の「[パラメータ](#)」と同様です。

返却値

分岐終了処理の返却値は、「[分岐開始処理](#)」の「[返却値](#)」と同様です。

案件終了処理（トランザクションなし）

Contents

- [パラメータ](#)
- [返却値](#)

案件終了処理（トランザクションなし）とは、案件終了処理のトランザクションとは独立した処理です。案件が終了する時に、一度実行されます。

案件終了処理（トランザクションなし）は、直前のアクション処理や到達処理とは独立した処理です。そのため、案件終了処理（トランザクションなし）でエラーが発生した場合、直前の処理を戻す（ロールバック）ことはできません。

また、案件終了処理（トランザクションなし）は、案件終了処理の前に実行されます。そのため、案件終了処理でエラーが発生した場合、案件終了処理（トランザクションなし）は実施済みとなり、処理を戻す（ロールバック）ことはできません。

このプログラム中で、データベースの登録／更新／削除処理を行う場合は、独自にDBトランザクション制御を行ってください。

下記の場合に実行されます。

- 最後の承認者が“承認”を行った場合
- 承認者が“承認終了”を行った場合
- 承認者が“否認”を行った場合
- 申請者が“取止め”を行った場合
- 案件操作で終了ノードに到達した場合

パラメータ

案件終了処理（トランザクションなし）では、以下の情報をパラメータとして受け取る事ができます。

No	パラメータ（物理名）	パラメータ（論理名）	属性	詳細
1	loginGroupId	ログイングループID	文字列	処理実行時のログイングループID ログイングループIDは、テナントIDと同値です。
2	localeId	ロケールID	文字列	処理実行時のロケールID
3	targetLocales	ターゲットロケールID	文字列[]	システムで利用しているロケールID

No	パラメータ (物理名)	パラメータ (論理名)	属性	詳細
4	contentsId	コンテンツID	文字列	該当案件のコンテンツID
5	contentsVersionId	コンテンツバージョンID	文字列	該当案件のコンテンツバージョンID
6	routeId	ルートID	文字列	該当案件のルートID
7	routeVersionId	ルートバージョンID	文字列	該当案件のルートバージョンID
8	flowId	フローID	文字列	該当案件のフローID
9	flowVersionId	フローバージョンID	文字列	該当案件のフローバージョンID
10	applyBaseDate	申請基準日	文字列	申請基準日 (“yyyy/MM/dd”)
11	processDate	処理日	文字列	処理日 (“yyyy/MM/dd”)
12	systemMatterId	システム案件ID	文字列	該当案件のシステム案件ID
13	userDataId	ユーザデータID	文字列	該当案件のユーザデータID
14	parameter	実行プログラムパス	文字列	該当処理の実行プログラムパス
15	actFlag	代理フラグ	文字列	“0”：本人にて処理 “1”：代理者にて処理
16	lastProcessNodeId	最終処理ノードID	文字列	最終処理のノードID
17	lastAuthUserCd	最終処理権限者コード	文字列	最終処理の処理権限者コード
18	lastExecUserCd	最終処理実行者コード	文字列	最終処理の処理実行者コード
19	lastResultStatus	最終処理結果ステータス	文字列	最終処理の処理結果ステータス [1]

[1] ……コード値は、「APIドキュメントのIM-Workflow CodeList」を参照してください。

返却値

案件終了処理（トランザクションなし）では、以下の情報を返却します。（任意/必須 ……●：必須 △：任意）

■ スクリプト開発モデル

No	返却値 (物理名)	返却値 (論理名)	属性	必須/任意	詳細
1	resultFlag	結果フラグ	真偽値	△	true：成功 false：失敗 指定しない場合、成功 (true) として扱います。
2	message	結果メッセージ	文字列	△	結果フラグが失敗の場合のみ、設定します。

※ 結果メッセージを設定した場合、例外ログにエラーメッセージを出力します。

コラム

結果フラグとして“false”（失敗）を設定した場合の動作は以下の通りです。

- 該当の案件終了処理（トランザクションなし）以外に設定されている後続の案件終了処理（トランザクションなし）は実行されません。

■ JavaEE開発モデル

返却値はありません。

※ 「jp.co.intra_mart.foundation.workflow.exception.WorkflowExternalException」をスローした場合、引数に設定したエラーメッセージを例外ログに出力します。詳細は「APIドキュメント」を参照してください。

コラム

WorkflowExternalException をスローした場合の動作は以下の通りです。

- 該当の案件終了処理（トランザクションなし）以外に設定されている後続の案件終了処理（トランザクションなし）は実行されません。

未完了案件削除処理リスナー

未完了案件削除処理リスナーとは、未完了案件を削除した際に実行されるプログラムです。
通常、「案件操作」画面より“案件削除”を行った場合、または未完了案件を削除するAPIを実行した際に呼び出されます。

未完了案件削除処理リスナーは、通常「コンテンツ定義」に設定します。

また、テナント単位で処理を行う場合は、下記のファイルに設定します。

```
%PUBLIC_STORAGE_PATH%/im_workflow/conf/param/param_group_%テナントID%.xml
```

```
1 <!--
2   未完了案件削除リスナーの種類
3   [java] or [script] or [logic-flow] or [] (指定なし)
4   [] (指定なし)を設定した場合はリスナーを起動しない
5 -->
6 <param>
7   <param-name>delete-active-matter-type</param-name>
8   <param-value></param-value>
9 </param>
10 <!--
11   未完了案件削除リスナーのパス
12   1. 案件削除リスナーの種類がjava : パッケージ名
13   2. 案件削除リスナーの種類がscript : WEB-INF/jjspからのパス
14   3. 案件削除リスナーの種類がlogic-flow : ロジックフロー定義情報
15 -->
16 <param>
17   <param-name>delete-active-matter-listener-path</param-name>
18   <param-value></param-value>
19 </param>
```

※[ワークフローパラメータ]画面からも設定することが可能です。

設定方法の詳細については、「[IM-Workflow 管理者操作ガイド](#)」または「[IM-Workflow 仕様書](#)」を参照してください。

完了案件削除処理リスナー

完了案件削除処理リスナーとは、完了案件を削除した際に実行されるプログラムです。
通常、「参照」画面の完了案件タブより案件の“削除”を行った場合、または完了案件を削除するAPIを実行した際に呼び出されます。

完了案件削除処理リスナーは、通常「コンテンツ定義」に設定します。

また、テナント単位で処理を行う場合は、下記のファイルに設定します。

```
%PUBLIC_STORAGE_PATH%/im_workflow/conf/param/param_group_%テナントID%.xml
```

```
1 <!--
2   完了案件削除リスナーの種類
3   [java] or [script] or [logic-flow] or [] (指定なし)
4   [] (指定なし)を設定した場合はリスナーを起動しない
5 -->
6 <param>
7   <param-name>delete-complete-matter-listener-type</param-name>
8   <param-value></param-value>
9 </param>
10 <!--
11   完了案件削除リスナーのパス
12   1. 案件削除リスナーの種類がjava : パッケージ名
13   2. 案件削除リスナーの種類がscript : WEB-INF/jjspからのパス
14   3. 案件削除リスナーの種類がlogic-flow : ロジックフロー定義情報
15 -->
16 <param>
17   <param-name>delete-complete-matter-listener-path</param-name>
18   <param-value></param-value>
19 </param>
```

※[ワークフローパラメータ]画面からも設定することが可能です。

設定方法の詳細については、「[IM-Workflow 管理者操作ガイド](#)」または「[IM-Workflow 仕様書](#)」を参照してください。

過去案件削除処理リスナー

過去案件削除処理リスナーとは、過去案件を削除した際に実行されるプログラムです。

過去案件削除処理リスナーは、通常「コンテンツ定義」に設定します。

また、テナント単位で処理を行う場合は、下記のファイルに設定します。

```
%PUBLIC_STORAGE_PATH%/im_workflow/conf/param/param_group_%テナントID%.xml
```

```

1 <!--
2   過去案件削除リスナーの種類
3     [java] or [script] or [logic-flow] or [] (指定なし)
4     [] (指定なし) を設定した場合はリスナーを起動しない
5 -->
6 <param>
7   <param-name>delete-archive-matter-listener-type</param-name>
8   <param-value></param-value>
9 </param>
10 <!--
11   過去案件削除リスナーのパス
12   1. 案件削除リスナーの種類がjava : パッケージ名
13   2. 案件削除リスナーの種類がscript : WEB-INF/jsspからのパス
14   3. 案件削除リスナーの種類がlogic-flow : ロジックフロー定義情報
15 -->
16 <param>
17   <param-name>delete-archive-matter-listener-path</param-name>
18   <param-value></param-value>
19 </param>

```

※[ワークフローパラメータ]画面からも設定することが可能です。

設定方法の詳細については、「[IM-Workflow 管理者操作ガイド](#)」または「[IM-Workflow 仕様書](#)」を参照してください。

案件退避処理リスナー

案件退避処理リスナーとは、案件を退避した際に実行されるプログラムです。

通常、ジョブ「IM-Workflow/アーカイブ」を実行した際に呼び出されます。

案件退避処理リスナーは、通常「コンテンツ定義」に設定します。

また、テナント単位で処理を行う場合は、下記のファイルに設定します。

```
%PUBLIC_STORAGE_PATH%/im_workflow/conf/param/param_group_%テナントID%.xml
```

```

1 <!--
2   案件退避リスナーの種類
3     [java] or [script] or [logic-flow] or [] (指定なし)
4     [] (指定なし) を設定した場合はリスナーを起動しない
5 -->
6 <param>
7   <param-name>archive-proc-listener-type</param-name>
8   <param-value>java</param-value>
9 </param>
10 <!--
11   案件退避リスナーのパス
12   1. 案件退避リスナーの種類がjava : パッケージ名
13   2. 案件退避リスナーの種類がscript : WEB-INF/jsspからのパス
14   3. 案件削除リスナーの種類がlogic-flow : ロジックフロー定義情報
15 -->
16 <param>
17   <param-name>archive-proc-listener-path</param-name>
18   <param-value></param-value>
19 </param>

```

※[ワークフローパラメータ]画面からも設定することが可能です。

設定方法の詳細については、「[IM-Workflow 管理者操作ガイド](#)」または「[IM-Workflow 仕様書](#)」を参照してください。

テンプレート

ユーザプログラムおよび各リスナーのプログラムを作成する際のテンプレートが提供されています。

- スクリプト開発モデル

<./jssp/src/sample/im_workflow/template/>

No	処理	物理名
1	案件開始処理	MatterStartProcess.js
2	案件終了処理	MatterEndProcess.js
3	アクション処理	ActionProcess.js
4	到達処理	ArriveProcess.js
5	分岐開始処理／分岐終了処理	RuleCondition.js
6	未完了案件削除処理リスナー	WorkflowActvMatterDeleteListener.js
7	完了案件削除処理リスナー	WorkflowCplMatterDeleteListener.js
8	過去案件削除処理リスナー	WorkflowArcMatterDeleteListener.js
9	案件退避処理リスナー	WorkflowMatterArchiveListener.js
10	処理対象者プラグイン	WorkflowAuthorityExecEventListener.js
11	案件終了処理（トランザクションなし）	MatterEndProcessNoTran.js

- JavaEE開発モデル

JavaEE開発モデル[javaファイル]のサンプルプログラムについては、製品メディアに保存されています。

また、製品最新情報ダウンロードページ (<http://www.intra-mart.jp/download/product/index.html>) から入手することもできます。

<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/template/>

No	処理	物理名
1	案件開始処理	MatterStartProcess.java
2	案件終了処理	MatterEndProcess.java
3	アクション処理	ActionProcess.java
4	到達処理	ArriveProcess.java
5	分岐開始処理／分岐終了処理	RuleCondition.java
6	未完了案件削除処理リスナー	WorkflowActvMatterDeleteListener.java
7	完了案件削除処理リスナー	WorkflowCplMatterDeleteListener.java
8	過去案件削除処理リスナー	WorkflowArcMatterDeleteListener.java
9	案件退避処理リスナー	WorkflowMatterArchiveListener.java
10	処理対象者プラグイン	WorkflowAuthorityExecEventListener.java
11	クローラ登録文書追加リスナー	WorkflowCrawlingAddListener.java
12	案件終了処理（トランザクションなし）	MatterEndProcessNoTran.java

サンプルプログラム

IM-Workflow のインストール時“サンプルデータセットアップ”を行い、サンプルデータをインポートした場合に使用できるサンプルプログラムについて説明します。

サンプルプログラムは、スクリプト開発モデルとJavaEE開発モデルのサンプルプログラムがあります。開発モデルの違いはありますが、どちらのサンプルも「物品購買」の申請書であり、動作仕様は同一です。

画面

Contents

- [申請／一時保存／申請（起票案件）／再申請画面](#)
- [処理画面](#)
- [確認画面](#)
- [処理詳細／参照詳細／過去案件詳細／確認詳細画面](#)

PC用画面とスマートフォン用画面について説明します。

PC用画面

The screenshot shows a web browser window with a title bar '物品購買 - スクリプト開発モデル -'. The main content area contains a form with the following fields:

- 品名 * (Item Name): A text input field.
- 数量 * (Quantity): A text input field.
- 金額 * (Amount): A text input field.
- 備考 (Remarks): A text area.

At the bottom of the form are two buttons: '申請' (Apply) and '一時保存' (Save Draft).

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/apply.html>
<./jssp/src/sample/im_workflow/purchase/screen/apply.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow-8.0.x-sample.jar/service-config-imw_sample_purchase.xml>
```

```
1 <service>
2 <service-id>apply</service-id>
3 <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApplyServiceController</controller-class>
4 <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApplyServiceTransition</transition-class>
5 <next-page>
6 <page-path>/sample/im_workflow/purchase/apply.jsp</page-path>
7 </next-page>
8 </service>
```



コラム

2020 Winter(Azalea) から「案件一覧」画面にコピー新規機能が追加されました。

コピー新規機能による申請画面への遷移時には、リクエストパラメータとして対象案件のユーザデータID（プロパティ名：imwOriginalUserDataId）が連携されます。

当該パラメータを利用してユーザデータを抽出することで、案件データを複製した申請を実現できます。

サンプルプログラムに実装例がありますので参考にしてください。

「案件一覧」画面の操作方法については「[IM-Workflow ユーザ操作ガイド](#)」 - 「[案件一覧](#)」を参照してください。

スマートフォン用画面

The screenshot shows a mobile application interface with a dark theme. At the top, there is a back arrow and the title '物品購買 - スクリプト開発モデル -'. Below the title are four input fields:

- 品名 * (Item Name)
- 数量 * (Quantity)
- 金額 * (Amount)
- 備考 (Remarks)

At the bottom, there are two buttons: '申請' (Apply) and '一時保存' (Save Draft). A home indicator bar is visible at the very bottom of the screen.

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow_smartphone/purchase/screen/apply.html>
<./jssp/src/sample/im_workflow_smartphone/purchase/screen/apply.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow_smartphone-8.0.x-sample.jar/service-config-imw_sp_sample_purchase.xml>
```

```
1 <service>
2 <service-id>apply</service-id>
3 <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApplyServiceController</controller-class>
4 <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApplyServiceTransition</transition-class>
5 <next-page>
6 <page-path>/sample/im_workflow_smartphone/purchase/apply.jsp</page-path>
7 </next-page>
8 </service>
```

処理画面

PC用画面とスマートフォン用画面について説明します。

PC用画面

物品購買 - スクリプト開発モデル -	
品名	パソコン
数量	1
金額	100000
合計	100000
備考	<input type="text"/>

処理

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/approve.html>
<./jssp/src/sample/im_workflow/purchase/screen/approve.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow-8.0.x-sample.jar/service-config-imw_sample_purchase.xml>
```

```
1 <service>
2 <service-id>approve</service-id>
3 <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApproveServiceController</controller-class>
4 <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApproveServiceTransition</transition-class>
5 <next-page>
6 <page-path>/sample/im_workflow/purchase/approve.jsp</page-path>
7 </next-page>
8 </service>
```

スマートフォン用画面



- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow_smartphone/purchase/screen/approve.html>
<./jssp/src/sample/im_workflow_smartphone/purchase/screen/approve.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow_smartphone-8.0.x-sample.jar/service-config-imw_sp_sample_purchase.xml>
```

```
1 <service>
2 <service-id>approve</service-id>
3 <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApproveServiceController</controller-class>
4 <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ApproveServiceTransition</transition-class>
5 <next-page>
6 <page-path>/sample/im_workflow_smartphone/purchase/approve.jsp</page-path>
7 </next-page>
8 </service>
```

確認画面

PC用画面とスマートフォン用画面について説明します。

PC用画面



- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/approve.html>
<./jssp/src/sample/im_workflow/purchase/screen/approve.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow-8.0.x-sample.jar/service-config-imw_sample_purchase.xml>
```

```

1 <service>
2   <service-id>confirm</service-id>
3   <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ConfirmServiceController</controller-class>
4   <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ConfirmServiceTransition</transition-class>
5   <next-page>
6     <page-path>/sample/im_workflow/purchase/confirm.jsp</page-path>
7   </next-page>
8 </service>

```

スマートフォン用画面

- スクリプト開発モデル

```

<./jssp/src/sample/im_workflow_smartphone/purchase/screen/confirm.html>
<./jssp/src/sample/im_workflow_smartphone/purchase/screen/confirm.js>

```

- JavaEE開発モデル

```

<./lib/im_workflow_smartphone-8.0.x-sample.jar/service-config-imw_sp_sample_purchase.xml>

```

```

1 <service>
2   <service-id>confirm</service-id>
3   <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ConfirmServiceController</controller-class>
4   <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.ConfirmServiceTransition</transition-class>
5   <next-page>
6     <page-path>/sample/im_workflow_smartphone/purchase/confirm.jsp</page-path>
7   </next-page>
8 </service>

```

[処理詳細](#) / [参照詳細](#) / [過去案件詳細](#) / [確認詳細画面](#)



- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/detail.html>
<./jssp/src/sample/im_workflow/purchase/screen/detail.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow-8.0.x-sample.jar/service-config-imw_sample_purchase.xml>
```

```
1 <service>
2 <service-id>detail</service-id>
3 <controller-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.DetailServiceController</controller-class>
4 <transition-class>jp.co.intra_mart.sample.workflow.purchase.controller.service.DetailServiceTransition</transition-class>
5 <next-page>
6 <page-path>/sample/im_workflow/purchase/detail.jsp</page-path>
7 </next-page>
8 </service>
```

処理詳細／参照詳細／過去案件詳細／確認詳細画面（以下、詳細画面）では、コンテンツ定義で定義した画面が表示されます。そのため、詳細画面にIM-Workflowの情報（案件名や添付ファイルなど）を表示する場合は、IM-Workflow が提供するタグライブラリを使用します。

案件の情報を表示するためのタグライブラリです。

案件番号	0000000018
案件名	物品購買
申請者	円山益男
申請基準日	2012/09/19

- スクリプト開発モデル detail.html

```
43 | </header>
44 | <imart type="workflowMatterData" systemMatterId=$data.imwSystemMatterId
45 |         displayItem="matter_number,matter_name,apply_user,apply_base_date" />
46 | <table class="imui-form">
```

- JavaEE開発モデル detail.jsp

```
51 | </header>
52 | <workflow:workflowMatterData systemMatterId='<%= (String)request.getAttribute("imwSystemMatterId")%>'
53 |         displayItem="matter_number,matter_name,apply_user,apply_base_date" />
54 | <table class="imui-form">
```

案件の添付ファイルを表示するためのタグライブラリです。

添付ファイル			
ファイル名	サイズ	登録者	登録日時
見積書	1 KB	円山益男	2012/09/19 19:49

- スクリプト開発モデル detail.html

```
69 | </table>
70 | <imart type="workflowMatterFile" systemMatterId=$data.imwSystemMatterId />
71 | </div>
```

- JavaEE開発モデル detail.jsp

```
78 | </table>
79 | <workflow:workflowMatterFile systemMatterId='<%= (String)request.getAttribute("imwSystemMatterId")%>' />
80 | </div>
```

※ 案件に添付ファイルがない場合は、表示されません。

これらの詳細画面は、スマートフォン版 IM-Workflow から画面遷移した際、PC用の画面を新しいウィンドウで開きます。スマートフォンからの画面遷移でPC用の画面を表示させたい場合は、明示的にクライアントタイプをPCに切り替える必要があります。

- スクリプト開発モデル detail.js

```
19 | function init ( request ) {
20 |   ClientTypeSwitcher.oneTimeSwitchTo('pc');
21 | }
```

- JavaEE開発モデル detail.jsp

```
7 | <%
8 | ClientTypeSwitcher.oneTimeSwitchTo("pc");
9 | %>
```

ClientTypeSwitcher について、詳細は「[APIドキュメント](#)」を参照してください。

また、新しいウィンドウで表示する画面にグローバルナビやマイメニューを表示させないようにするには、以下のフィルタに画面のパスを追加する必要があります。

```
<./conf/theme-head-only-path-config/*.xml>
```

- スクリプト開発モデル

```
7 | <path>/sample/im_workflow/purchase/screen/detail</path>
```

- JavaEE開発モデル

```
8 | <path>/imw_sample_purchase-detail.service</path>
```

ユーザプログラム

Contents

- [アクション処理プログラム](#)
- [案件終了処理プログラム](#)
- [分岐開始処理プログラム](#)

アクション処理プログラム

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/action/ActionProcess1.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/ActionProcess1.java>
```

サンプルデータでは[ActionProcess1]を申請ノードのアクション処理として定義されています。

[ActionProcess1]では、下記の2つの処理を行っています。

- ユーザアプリケーションのデータをテーブルに保存する。
 - 申請または一時保存を行った場合に、画面に入力された情報をユーザアプリケーションで定義している独自のテーブルに登録／更新しています。
- 案件番号を採番する。
 - 案件番号は、申請のアクション処理で設定する必要があります。
 - ここでは、IM-Workflow が提供する「WorkflowNumberingManager#getNumber()」で案件番号の採番を行っています。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/action/ActionProcess2.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/ActionProcess2.java>
```

サンプルデータでは[ActionProcess2]を申請ノードのアクション処理として定義しています。

[ActionProcess2]では、画面から入力された「数量×金額」である“合計金額”を案件プロパティとして登録する処理を行っています。

案件終了処理プログラム

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/action/MatterEndProcess.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/MatterEndProcess.java>
```

サンプルデータでは[MatterEndProcess]を案件終了処理として定義しています。

[MatterEndProcess]では、ユーザアプリケーションで定義している独自のテーブルの更新処理を行っています。

分岐開始処理プログラム

- フロー定義“分岐ルート[スクリプト開発モデル]”で使用されている分岐開始処理プログラム

```
<./jssp/src/sample/im_workflow/purchase/action/RuleCondition1.js>
<./jssp/src/sample/im_workflow/purchase/action/RuleCondition2.js>
<./jssp/src/sample/im_workflow/purchase/action/RuleCondition3.js>
```

- フロー定義“分岐ルート[JavaEE開発モデル]”で使用されている分岐開始処理プログラム

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/RuleCondition1.java>
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/RuleCondition2.java>
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/action/RuleCondition3.java>
```

[RuleCondition1]では、“合計金額”が10000未満の場合に結果フラグとして成功（true）を返却します。

[RuleCondition2]では、“合計金額”が10000以上50000未満の場合に結果フラグとして成功（true）を返却します。

[RuleCondition3]では、“合計金額”が50000以上の場合に結果フラグとして成功（true）を返却します。

リスナー

Contents

- 未完了案件削除処理リスナー
- 完了案件削除処理リスナー
- 過去案件削除処理リスナー
- 案件退避処理リスナー

未完了案件削除処理リスナー

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/listener/WorkflowActvMatterDeleteListener.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/listener/WorkflowActvMatterDeleteListener.java>
```

[WorkflowActvMatterDeleteListener]では、下記の2つの処理を行っています。

- ユーザアプリケーションのデータをテーブルから削除する。
 - 申請時に登録したユーザアプリケーションのデータを案件削除と同タイミングで削除しています。
- 案件プロパティを削除する。
 - 申請時に案件プロパティに登録した“合計金額”を案件プロパティから削除しています。

完了案件削除処理リスナー

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/listener/WorkflowCplMatterDeleteListener.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%  
/jp/co/intra_mart/sample/workflow/purchase/listener/WorkflowCplMatterDeleteListener.java>
```

[WorkflowCplMatterDeleteListener]では、次の処理を行っています。

- 申請時に登録したユーザアプリケーションのデータを案件削除と同タイミングで削除しています。
- ※ 案件プロパティの情報は、案件削除のタイミングで IM-Workflow モジュールが自動的に削除しますので、個別の削除は不要です。

過去案件削除処理リスナー

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/listener/WorkflowArcMatterDeleteListener.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%  
/jp/co/intra_mart/sample/workflow/purchase/listener/WorkflowArcMatterDeleteListener.java>
```

[WorkflowArcMatterDeleteListener]では、次の処理を行っています。

- 申請時に登録したユーザアプリケーションのデータを案件削除と同タイミングで削除しています。
- ※ 案件プロパティの情報は、案件削除のタイミングで IM-Workflow モジュールが自動的に削除しますので、個別の削除は不要です。

案件退避処理リスナー

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/listener/WorkflowMatterArchiveListener.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%  
/jp/co/intra_mart/sample/workflow/purchase/listener/WorkflowMatterArchiveListener.java>
```

[WorkflowMatterArchiveListener]では、次の処理を行っています。

- ユーザアプリケーションで定義している独自のテーブルの更新処理を行っています。

呼び出し画面の初期表示値指定

ここで記載している内容は、次の観点で共通です。

- 開発モデル
- クライアントタイプ

IM-Workflow で提供する各処理（申請／再申請／申請（起票案件）／一時保存／処理／確認）画面の呼び出し時に、呼び出し画面における初期表示値を外部指定する方法を説明します。

指定可能なパラメータ

「workflowOpenPage」タグの内部に下記パラメータを記述することにより、呼び出し画面における初期表示値を外部指定することが可能です。

No	パラメータ（物理名）	パラメータ（論理名）	呼び出し画面側の対応項目	動作対象呼び出し画面
1	imwMatterName	案件名	案件名	申請／一時保存／申請（起票案件）／再申請
2	imwComment	コメント	コメント	すべて
3	imwForcedParamFlag	強制パラメータフラグ	※動作制御用フラグ	-

また、下記のような条件のとき、「imwForcedParamFlag」（強制パラメータフラグ）の値に「1」を指定した場合のみ、初期表示値指定が反映されます。「imwForcedParamFlag」（強制パラメータフラグ）の値に「1」を指定しない場合、または、「imwForcedParamFlag」（強制パラメータフラグ）を記述しない場合は、登録されている情報が優先されます。

No	呼び出し画面	条件
1	申請	一時保存からの申請時
2	一時保存	一時保存情報の再保存時
3	申請（起票案件）	-
4	再申請	-

実装例

サンプルとして提供されている「物品購買」の申請書において、申請画面で入力される「品名」を「案件名」に、「備考」を「コメント」に初期表示する例です。

なお、サンプルはPC用画面のみ用意しています。

スマートフォン用画面の場合も全体の流れは同じです。実装中で使用するタグライブラリや Client-side JavaScript API が異なることに注意してください。



下記のプログラムが、初期表示を行うための処理が記述されたプログラムです。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/apply_display.html>
```

- JavaEE開発モデル

```
< (展開したwar) /sample/im_workflow/purchase/apply_display.jsp>
```

これらのファイルを、以下のファイル名に変更し、上書き保存することで、申請画面において本機能の動作確認を行うことができます。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/apply.html>
```

- JavaEE開発モデル

```
< (展開したwar) /sample/im_workflow/purchase/apply.jsp>
```

以下のような処理を記述することで、初期表示を行うことができます。

```

1 <imart type="head">
2 <title>
3   <imart type="string" value=$msg.cap010 escapeXml="true" escapeJs="false" />
4 </title>
5
6 <imart type="workflowOpenPageCsjs" />
7 <script src="ui/libs/jquery-validation-1.9.0/jquery.validate.js"></script>
8 <script type="text/javascript">
9   .
10  .
11  .
12
13 // 入力された品名と備考を、案件名、コメントにセットします。
14 function setParam() {
15   $('#imwMatterName').val($('#item_name').val());
16   $('#imwComment').val($('#item_comment').val());
17   $('#imwForcedParamFlag').val('1');
18 }
19
20 $(function(){
21   .
22   .
23   .
24   $('#openPage1').click(function(){
25     // 値をセットする処理を呼び出します。
26     setParam();
27     workflowOpenPage('1');
28   });
29   .
30   .
31   .
32
33 }
34
35 </script>
36 </imart>
37
38 <imart type="workflowOpenPage"
39   name="workflowOpenPageForm"
40   id="workflowOpenPageForm"
41   method="POST"
42   target="_top"
43   imwUserDataId=$data.imwUserDataId
44   imwSystemMatterId=$data.imwSystemMatterId
45   imwAuthUserCode=$data.imwAuthUserCode
46   imwApplyBaseDate=$data.imwApplyBaseDate
47   imwNodeId=$data.imwNodeId
48   imwFlowId=$data.imwFlowId
49   imwCallOriginalParams=$data.imwCallOriginalParams
50   imwNextScriptPath=$data.imwCallOriginalPagePath>
51
52 //案件名、コメント値を保持するエリアを定義します。
53 <input type="hidden" name="imwMatterName" id="imwMatterName" />
54 <input type="hidden" name="imwComment" id="imwComment" />
55 <input type="hidden" name="imwForcedParamFlag" id="imwForcedParamFlag" />
56   .
57   .
58   .
59 </imart>

```

処理対象者プラグインの作成

IM-Workflow の各ノードに指定する「処理対象者」に、独自に作成した処理対象者を追加する方法を説明します。

IM-Workflow の処理対象者は、プラグインという形で機能を拡張できます。

プラグインを追加する場合には、拡張ポイントに応じた内容でプラグインの実装を作成し、対象の拡張ポイントへPluginするための設定ファイルを記述します。拡張ポイントと、プラグインの関係は intra-mart Accel Platform のAPIである“PluginManager”によって管理されます。

対象ノード（拡張ポイント）

処理対象者プラグインは、ノードの種類により「extension point」が決められています。

No	ノード	extension point
1	承認（※1）	jp.co.intra_mart.workflow.plugin.authority.node.approve
2	承認（※2）	jp.co.intra_mart.workflow.plugin.authority.node.approve.static
3	動的承認	jp.co.intra_mart.workflow.plugin.authority.node.dynamic
4	確認	jp.co.intra_mart.workflow.plugin.authority.node.confirm

※1 前ノードが、“申請ノード”または“承認ノード”の場合

※2 前ノードが、“申請ノード”または“承認ノード”以外の場合

サンプルの説明

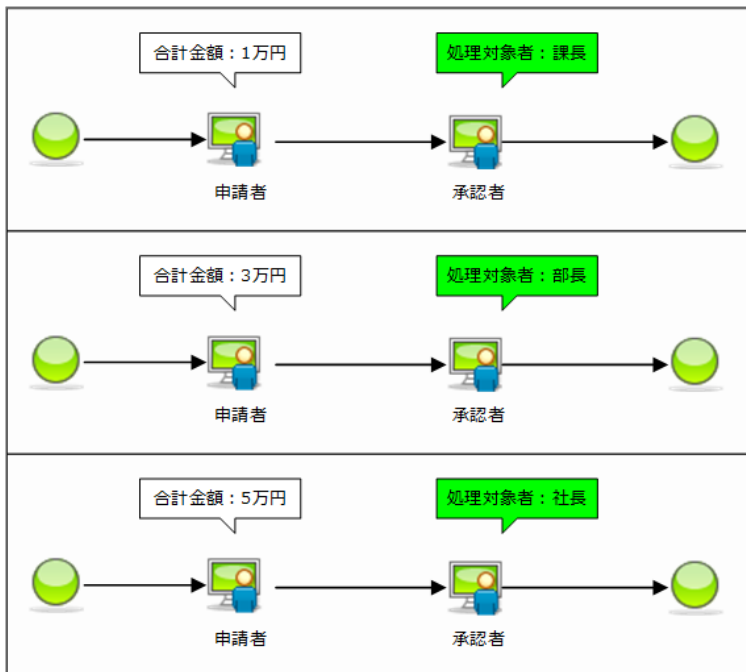
サンプルで提供する“処理対象者プラグイン”は、同じくサンプルで提供されている“物品購買”の画面と連携しています。

“物品購買”の画面で入力された「数量」と「金額」からの「合計金額」により、次の承認者を決定します。

具体的には、「合計金額」により、

- 1万円未満
 - 課長
- 1万円以上かつ5万円未満
 - 部長
- 5万円以上
 - 社長

と、処理対象者に役職が割り当てられます。



サンプルの実行準備

ここでは、承認ノードに対して、「合計金額」で処理対象者を決めるプラグインを使用してみます。

下記のファイルを編集します。

```
<./plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve/plugin.xml>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <plugin>
3    <extension point="jp.co.intra_mart.workflow.plugin.authority.node.approve" >
4
5      <authority
6        name="%jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.script"
7        id="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.script"
8        version="7.2.0"
9        rank="910"
10       <b>enable="true"</b>
11      <configPage>
12        <script pagePath="sample/im_workflow/purchase/plugin/authority/item_total/itemTotalConfig">
13          <parameter key="pluginName" value="SAMPLE.IMW.CAP.030" />
14        </script>
15      </configPage>
16      <extend>
17        <script
18          file="sample/im_workflow/purchase/plugin/authority/item_total/WorkflowAuthorityExecEventListener" />
19        </script>
20      </extend>
21    </authority>
22
23    <authority
24      name="%jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.javaee"
25      id="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.javaee"
26      version="7.2.0"
27      rank="920"
28      <b>enable="true"</b>
29      <configPage>
30        <javaee applicationId="imw_sample_purchase" serviceId="authority_item_total">
31          <parameter key="pluginName" value="SAMPLE.IMW.CAP.031" />
32        </javaee>
33      </configPage>
34      <extend>
35        <java
36          class="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.item_total.WorkflowAuthorityExecEventListener" />
37        </java>
38      </extend>
39    </authority>
40  </extension>
41 </plugin>

```

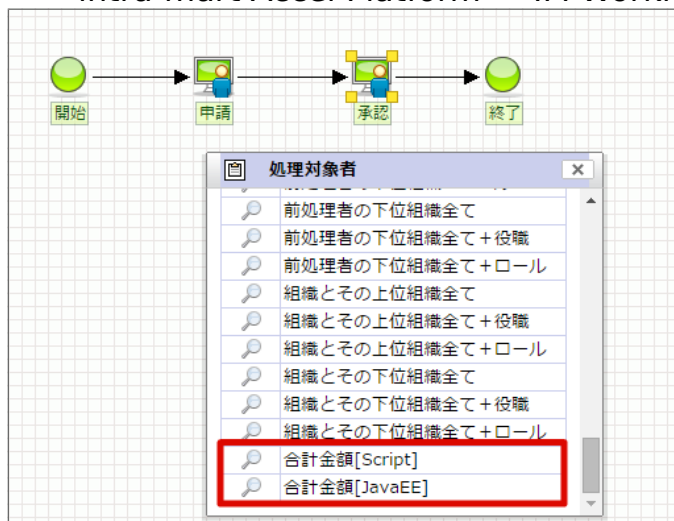
スクリプト開発モデル

JavaEE開発モデル

上記ファイルを編集後、サーバを再起動します。

[ルート定義]画面より、次のようなルートを作成します。

承認ノードの処理対象者の検索を行うと、下記のように「合計金額[Script]」および「合計金額[JavaEE]」が表示されます。



「合計金額[Script]」および「合計金額[JavaEE]」は、実装方法（開発言語）の違いによるもので、処理内容に関して違いはありません。

「合計金額[Script]」または「合計金額[JavaEE]」を選択し、ルートを作成します。

次に、[フロー定義]画面より、上記で作成したルート定義を使用したフロー定義を作成します。

この時、コンテンツは、サンプルで提供されている「スクリプト開発モデル」または、「JavaEE開発モデル」を選択してください。

サンプルの実行

「[サンプルの実行準備](#)」で作成したフロー定義で申請を行います。

The screenshot shows the '物品購買 - スクリプト開発モデル' (Purchase Item - Script Development Model) form. The form has the following fields: 品名 (Item Name), 数量 (Quantity), 金額 (Amount), and 備考 (Remarks). The 数量 and 金額 fields are highlighted with a red box. There are two buttons at the bottom: 申請 (Apply) and 一時保存 (Save Temporarily).

入力した「数量」と「金額」からの「合計金額」により、承認ノードの処理対象者が変わることを確認します。

[処理済]一覧画面より、申請を行った案件のフローを参照します。



- 合計金額が1万円未満の場合



- 合計金額が1万円以上かつ5万円未満の場合



- 合計金額が5万円以上の場合



「合計金額」により、処理対象者が違うことを確認します。

処理対象者プラグインの構成

処理対象者プラグインを作成するには、次の3ファイルを作成する必要があります。

Contents

- <plugin.xml>ファイル定義
 - pluginタグ設定
 - configPage設定
 - extend設定

<plugin.xml>ファイル定義

「plugin.xml」は、「PluginManager」によって管理されるファイルです。
 処理対象者プラグインを新規に作成する場合は、plugin.xml を新規作成します。
 IM-Workflow が提供するサンプルの plugin.xml を参考に作成されることを推奨します。

intra-mart Accel Platform が提供するプラグインの仕様については、以下のドキュメントも合わせて参照してください。

- 「[プラグイン仕様書](#)」 - 「[プラグインの構成](#)」

作成したプラグインを intra-mart Accel Platform 上に追加する方法は以下のドキュメントを参照してください。

- 「[IM-Workflow 管理者操作ガイド](#)」 - 「[処理対象者プラグインを設定する](#)」

本項では、次の plugin.xml の内容をもとに、処理対象者プラグインを構成するそれぞれの要素、属性について説明します。

/plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve/plugin.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <plugin>
3    <extension point="jp.co.intra_mart.workflow.plugin.authority.node.approve" >
4
5    <authority
6      id="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.script"
7      name="%jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.script"
8      version="8.0.99"
9      rank="910"
10     enable="true">
11     <configPage>
12       <script pagePath="sample/im_workflow/purchase/plugin/authority/item_total/itemTotalConfig">
13         <parameter key="pluginName" value="SAMPLE.IMW.CAP.030" />
14       </script>
15     </configPage>
16     <extend>
17       <script
18         file="sample/im_workflow/purchase/plugin/authority/item_total/WorkflowAuthorityExecEventListener" />
19       </extend>
20     </authority>
21
22     <authority
23       id="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.javaee"
24       name="%jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.item_total.javaee"
25       version="8.0.99"
26       rank="920"
27       enable="true">
28       <configPage>
29         <javaee applicationId="imw_sample_purchase" serviceId="authority_item_total">
30           <parameter key="pluginName" value="SAMPLE.IMW.CAP.031" />
31         </javaee>
32       </configPage>
33       <extend>
34         <java class="jp.co.intra_mart.sample.workflow.purchase.plugin.authority.item_total.WorkflowAuthorityExecEventListener" />
35       </extend>
36     </authority>
37
38   </extension>
39 </plugin>

```

pluginタグ設定

処理対象者プラグインで重要になるのは、下記の要素です。

<extension point>		処理対象者プラグインを差し込むノードの種類により、<extension point>が変わります。差し込みたいノードの<extension point>を指定します。
<configPage>	<script pagePath> <javaee applicationId serviceId>	<configPage>は、[ルート定義]画面において、ノードに設定する処理対象者の一覧画面から、処理対象者プラグインが選択されたときに呼ばれるプログラムです。 このプログラムは、スクリプト開発モデルおよび、JavaEE開発モデルで記述することが可能です。スクリプト開発モデルでプログラムを作成する場合は、<script pagePath>にパスを指定します。JavaEE開発モデルでプログラムを作成する場合は、applicationId および serviceId を指定します。
< extend >	<script file> <java class>	< extend > に指定するプログラムは、処理対象者を決定するプログラムが対象です。 このプログラムは、スクリプト開発モデルおよび、JavaEE開発モデルで記述することが可能です。スクリプト開発モデルでプログラムを作成する場合は、<script file>にパスを指定します。JavaEE開発モデルでプログラムを作成する場合は、<java class>にパッケージを指定します。

その他の要素、属性については、「[PluginManagerのAPIドキュメント](#)」を参照してください。



コラム

処理対象者プラグインは、「キャッシュ設定」（「[IM-Workflow 仕様書](#)」）によりキャッシュされ、処理対象者プラグインのインスタンスは複数のスレッドで再利用しています。
そのため、処理対象者プラグインの実装にあたっては、メンバ変数を利用しないようにしてください。

サンプルの plugin.xml としては、下記を用意しています。

- 承認ノード

```
./plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve/plugin.xml
```

- 承認ノード

```
<./plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.approve.static/plugin.xml>
```

- 動的承認ノード

```
<./plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.dynamic/plugin.xml>
```

- 確認ノード

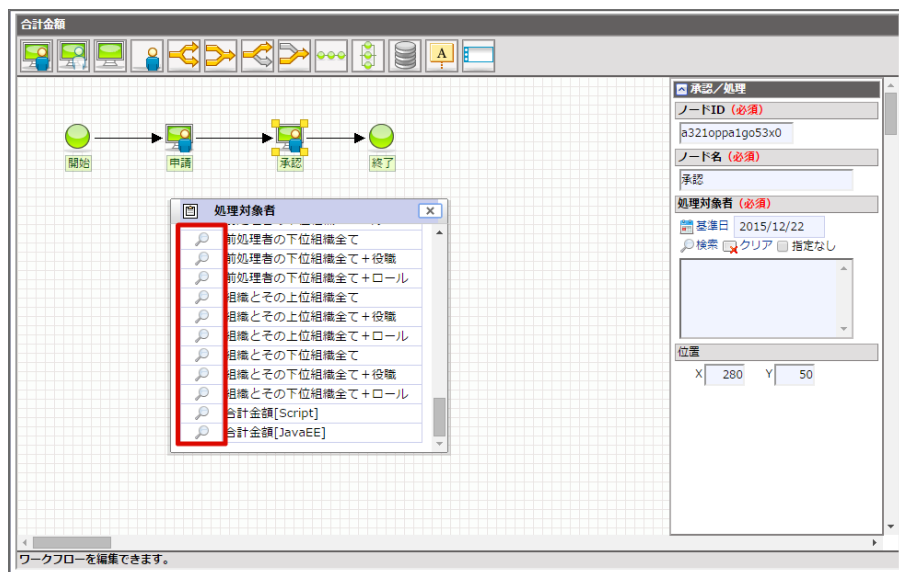
```
<./plugin/jp.co.intra_mart.sample.workflow.purchase.plugin.authority.node.confirm/plugin.xml>
```

configPage設定

前出の plugin.xml で<configPage>として指定したプログラムを作成します。

[ルート定義]画面において、ノードに設定する処理対象者の一覧画面から、処理対象者プラグインが選択されたときに呼ばれるプログラムです。

製品が提供している処理対象者プラグインのように検索を利用したい場合には、<configPage>で指定したプログラム内で検索画面を呼び出すように実装してください。



選択された対象者プラグインの情報を、[ルート定義]画面に引き渡します。

サンプルプログラムとしては、下記を用意しています。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/plugin/authority/item_total/itemTotalConfig.html>
<./jssp/src/sample/im_workflow/purchase/plugin/authority/item_total/itemTotalConfig.js>
```

- JavaEE開発モデル

```
<./lib/im_workflow-8.0.0-sample.jar/service-config-imw_sample_purchase.xml>
```

```

1 <service>
2   <service-id>authority_item_total</service-id>
3
4   <controller-
5     class>jp.co.intra_mart.sample.workflow.purchase.plugin.authority.item_total.controller.service.ItemTotalConfigServiceController</controller-
6     class>
7
8   <transition-
9     class>jp.co.intra_mart.sample.workflow.purchase.plugin.authority.item_total.controller.service.ItemTotalConfigTransition</transition-class>
10
11   <next-page>
    <page-path>/sample/workflow/purchase/plugin/authority/item_total/itemTotalConfig.jsp</page-path>
  </next-page>
</service>
```

extend設定

前出の plugin.xml で<extend>として指定したプログラムを作成します。

処理対象者を決定する際に実行されるプログラムです。

ここで指定するプログラムには、次の3つのメソッドを実装する必要があります。

メソッド	概要
execute(WorkflowAuthorityParameter workflowParam, WorkflowMatterParameter matterParam)	処理対象者を取得するメソッド 対象のノードに案件が到達したときに実行されます。
getTargetUserList(WorkflowAuthorityParameter workflowParam, WorkflowMatterParameter matterParam, WorkflowSortCondition[] sort)	処理対象ユーザの一覧を取得するメソッド ロケール単位にユーザモデルのリスト (List<UserDataModel>) を返却する必要があります。 [案件操作]-[ノード編集]画面の「状況確認」ボタン押下時に表示される[対象者状況確認]画面で使用されます。
getDisplayname(WorkflowAuthorityParameter workflowParam)	処理対象者プラグインの名称を取得するメソッド ロケール単位に取得してください。 プラグインの名称を表示するため使用されます。

サンプルプログラムとしては、下記を用意しています。

- スクリプト開発モデル

```
<./jssrc/sample/im_workflow/purchase/plugin/authority/item_total/WorkflowAuthorityExecEventListener.js>
```

- JavaEE開発モデル

```
<%サンプルプログラムディレクトリ%/jp/co/intra_mart/sample/workflow/purchase/plugin/authority/item_total/WorkflowAuthorityExecEventListener.java>
```



コラム

独自のプログラムを作成する際は、上述のサンプルプログラムのほか、「[テンプレート](#)」も併せて参照してください。
実装すべきロジックの詳細はサンプルおよびテンプレートに記載しています。

画面入力情報の保持

ここで記載している内容は、次の観点において共通です。

- 開発モデル
- クライアントタイプ

申請画面、一時保存画面、申請（起票案件）画面、再申請画面、処理画面、確認画面において、「閉じる」リンク（PC用画面）または「戻る」リンク（スマートフォン用画面）によって各画面を閉じた後に画面の再表示を行ったとき、入力内容を保持した状態で画面表示されます。

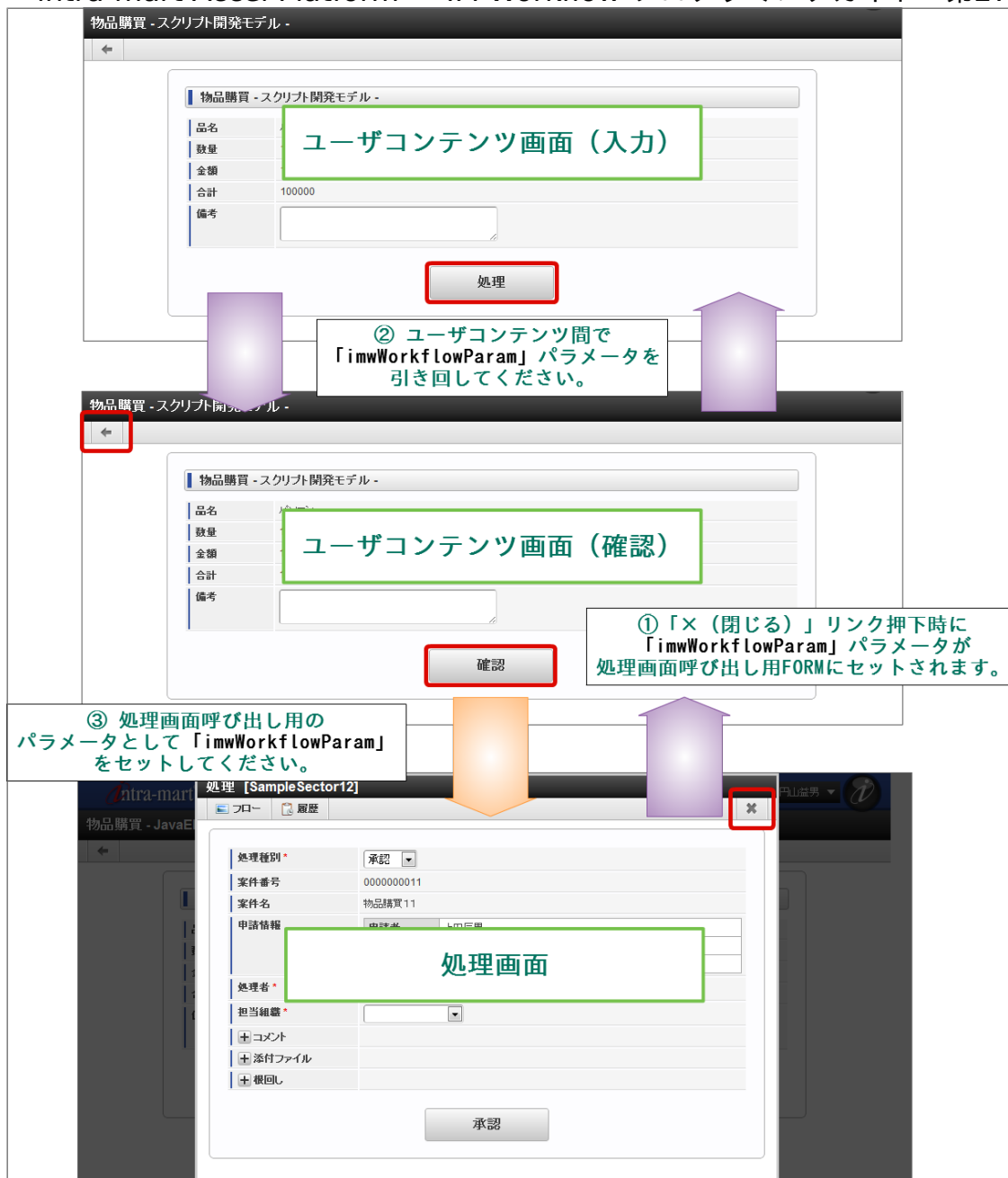
当機能の仕様概要は以下の通りです。

- 各処理画面の「閉じる」「戻る」リンク押下時に、呼出元ユーザコンテンツ内の画面呼出用タグライブラリによって生成されたFORMに対して「imWorkflowParams」というパラメータ名のhiddenタグを追加し、そのタグに入力情報を格納
- 再度画面表示した際にリクエストパラメータとして「imWorkflowParams」が含まれている場合、画面の初期表示処理で保持情報による復元表示を実行

リクエストパラメータの受け渡しによって入力情報再表示が行われるため、ユーザコンテンツが単一画面構成の場合は意識する必要がありませんが、複数画面で構成されている場合は以下対応が必要です。

各処理画面を閉じてからユーザコンテンツ間の画面遷移が行われ、その後入力内容を保持した状態で各処理画面の再表示を行う必要がある場合、「imWorkflowParams」パラメータをユーザコンテンツ間で引き回し、各処理画面表示用のタグライブラリのコンテンツ内に「imWorkflowParams」パラメータをhiddenタグで明示的に記述してください。

<ユーザコンテンツ 複数画面構成での実装イメージ>



呼び出し画面からのコールバック関数の指定

ここで記載している内容は、次の観点において共通です。

- 開発モデル
- クライアントタイプ

申請画面、一時保存画面、申請（起票案件）画面、再申請画面、処理画面、確認画面において、「閉じる」リンク（PC用画面）または「戻る」リンク（スマートフォン用画面）によって各画面を閉じる際のコールバック関数を指定可能です。

またコールバック関数は、「[処理完了後の画面遷移](#)」に記載のパラメータ (imwNext~) の指定を行っていない場合、IM-Workflow で提供する各処理（申請／再申請／申請（起票案件）／一時保存／処理／確認）画面の処理完了後にも実行されます。

呼出元のユーザコンテンツ画面の関数を実行する方法について説明します。

実装例

サンプルとして提供されている「物品購買」の申請書において、GreyBoxで表示される申請画面の閉じる処理が実行された際に、「物品購買」の申請書で定義された関数をコールバック関数として実行する例です。

なお、サンプルはPC用画面のみ用意しています。

スマートフォン用画面の場合も全体の流れは同じです。実装中で使用するタグライブラリや Client-side JavaScript API が異なることに注意してください。

下記のプログラムが、コールバック関数の実行を行うための処理が記述されたプログラムです。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/apply_callback.html>
```

- JavaEE開発モデル

```
< (展開したwar) /sample/im_workflow/purchase/apply_callback.jsp>
```

上記ファイルを、以下のファイル名に変更し、上書き保存することで、申請画面において本機能の動作確認を行うことができます。

- スクリプト開発モデル

```
<./jssp/src/sample/im_workflow/purchase/screen/apply.html>
```

- JavaEE開発モデル

```
< (展開したwar) /sample/im_workflow/purchase/apply.jsp>
```

以下のような処理を記述することで、コールバック関数の実行を行うことができます。

```

1 <imart type="head">
2
3 <imart type="workflowOpenPageCsjs"/>
4 <script type="text/javascript">
5
6 function onClickOpenPage(pageType) {
7   if (pageType != "1") {
8     if(!inputCheck()) {
9       return;
10    }
11  }
12
13  // 申請画面を表示する際に、コールバック関数を引数で指定します。
14  workflowOpenPage(pageType, callbackFnc);
15 }
16
17 // 申請画面を「閉じる」または「戻る」際に行う処理を記載します。
18 function callbackFnc() {
19   alert("Callback function is executed.");
20 }
21 .
22 .
23 .
24 <imart type="form" name="backForm" method="POST" page=$data.imwCallOriginalPagePath>
25   <imart type="hidden" imwCallOriginalParams=$data.imwCallOriginalParams />
26 </imart>
```

IM-Workflow で提供する各処理（申請／再申請／申請（起票案件）／一時保存／処理／確認）画面の処理完了後にコールバック関数が実行された場合、コールバック関数は処理された案件の情報を引数として受け取ることができます。

```

1 function callbackFnc(result) {
2   alert("Callback function is executed.");
3   alert( result.imwSystemMatterId ); // システム案件ID
4   alert( result.imwUserDataId ); // ユーザーデータID
5 }
```

処理種別と受け取ることのできる情報の関係は以下の通りです。

処理種別	システム案件ID imwSystemMatterId	ユーザーデータID imwUserDataId
申請	○	-
再申請	○	-
申請（起票案件）	○	-
一時保存	-	○
処理	○	-
確認	○	-

< 「○」 : 取得可能 / 「-」 : 取得不可能 >

標準画面を非同期で実行する場合の注意点

IM-Workflow バージョン 8.0.4より標準画面の処理を非同期に行う機能が追加されました。この機能が有効の場合、標準画面の呼び出し元画面で指定されたコールバック関数の振る舞いが異なります。

IM-Workflow で提供する各処理が非同期として受付された時点で処理完了を各処理画面に通知します。

ほぼ処理開始の時点で通知するイメージです。

従いまして、標準画面の呼び出し元画面で指定されたコールバック関数が実行された時点では各処理が完了していない可能性が高いです。

そのため処理種別が申請の場合は、システム案件IDを受け取ることはできません。

特記事項

Contents

- IM-Workflow バージョン8.0.2 における改善

IM-Workflow バージョン8.0.2 における改善

IM-Workflow バージョン8.0.2 から、連続処理／連続確認中のコールバック呼び出しの動作仕様を改善しています。

- IM-Workflow バージョン8.0.1 までの動作仕様
 - コールバック関数の指定有無に関わらず、コールバック関数は実行されません。
- IM-Workflow バージョン8.0.2 以降の動作仕様
 - 「[処理完了後の画面遷移](#)」に記載のパラメータ (imwNext~) を指定しない場合には各処理完了後にコールバック関数が実行されます。

※ IM-Workflow スマートフォン では、連続処理／連続確認機能が存在しないため、上記を意識する必要はありません。

処理完了後の画面遷移

ここで記載している内容は、次の観点において共通です。

- 開発モデル
- クライアントタイプ

IM-Workflow で提供する各処理（申請／再申請／申請（起票案件）／一時保存／処理／確認）画面の処理後に、任意の画面に遷移することが可能です。

遷移先を指定するためのパラメータ

IM-Workflow で提供する各処理（申請／再申請／申請（起票案件）／一時保存／処理／確認）画面の呼び出し時、「workflowOpenPage」タグの属性に下記パラメータを記述すると、処理完了後の遷移先を指定することができます。

No	パラメータ（物理名）	省略	説明
1	imwNextScriptPath	可	処理完了後に遷移する画面のスク립トパス 処理後の遷移先がスク립ト開発画面の場合に指定が必要です。
2	imwNextApplicationId	可	処理完了後に遷移する画面のアプリケーションID 処理後の遷移先がjavaEE開発画面の場合に指定が必要です。
3	imwNextServiceId	可	処理完了後に遷移する画面のサービスID 処理後の遷移先がjavaEE開発画面の場合に指定が必要です。
4	imwNextPagePath	可	処理完了後に遷移する画面のページパス 処理後の遷移先がURL(旧名称:JSP or Servlet)の場合に指定が必要です。

実現したい画面遷移によって指定する属性を決定してください。

- 処理後にユーザコンテンツの呼出元一覧画面に遷移したい場合
 - 「imwNextScriptPath」に、一覧から渡された「imwCallOriginalPagePath」を指定してください。
※ 連続処理、連続確認の場合は、次の案件ノードがあれば、該当のユーザコンテンツに遷移します。
次の案件ノードがなければ、呼出元一覧画面に遷移します。
- 処理後に任意の画面に遷移したい場合
 - 「imwNext~」に、遷移先の画面パスを指定してください。
- 処理後にユーザコンテンツ独自のコールバック関数を実行して処理画面を閉じる、または処理画面を閉じるのみ実行したい場合
 - 「imwNext~」に何も設定しないでください。

遷移先画面が受け取ることのできるリクエストパラメータ

遷移元の処理画面の種類によって、遷移先では下記の情報をリクエストパラメータとして受け取る事ができます。

No	遷移元処理画面	パラメータ（物理名）	パラメータ（論理名）	備考
1	申請／再申請／申請（起票案件）／処理／確認	imwSystemMatterId	システム案件ID	-
2	一時保存	imwUserDataId	ユーザデータID	-

No	遷移元処理画面	パラメータ（物理名）	パラメータ（論理名）	備考
3	すべて	imwCallOriginalParams	呼出元パラメータ	ユーザコンテンツが一覧画面からリクエストパラメータとして受け取ることできる値と同じ値が受け取れます。
4	すべて ※連続処理／連続確認中の場合のみ	imwCallOriginalPagePath	呼出元ページパス	ユーザコンテンツが一覧画面からリクエストパラメータとして受け取ることできる値と同じ値が受け取れます。

特記事項

Contents

- [IM-Workflow バージョン8.0.2 における改善](#)
- [意図しないURLに対するバリデーション](#)

IM-Workflow バージョン8.0.2 における改善

IM-Workflow バージョン8.0.2 から、連続処理／連続確認中の画面遷移仕様を改善しています。

- IM-Workflow バージョン8.0.1 までの動作仕様
 - 処理完了後の遷移先指定は無視されます。
 - 処理完了後の遷移先指定の有無に関わらず、処理完了後は次の案件のユーザコンテンツが表示されます。
- IM-Workflow バージョン8.0.2 以降の動作仕様
 - 処理完了後の遷移先指定が行われている場合、処理完了後は指定された画面に遷移します。

※ IM-Workflow スマートフォン では、連続処理／連続確認機能が存在しないため、上記を意識する必要はありません。

意図しないURLに対するバリデーション

「imwNextPagePath」には、ルーティング設定によってマッピングされたURLが指定されることを前提としていますが、絶対URLを指定された場合でも動作します。

これは処理完了後の画面遷移先として IM-Workflow と連携する外部システムのURLも設定可能とするため、IM-Workflow Ver.7.2.x との互換性を保つための仕様です。

「workflowOpenPage」タグライブラリ、および「spWorkflowOpenPage」タグライブラリでは、明示的に外部システム等のURLを指定される可能性のある「imwNextPagePath」と、「imwNextApplicationId」および「imwNextServiceId」に対して、テナント単位で設定が可能な「セーフURLリスト」に存在するかチェックを行います。

チェックには、intra-mart Accel Platform が提供するAPI「SafeUrlManager#isSafe(String url)」を使用しています。

ワークフローエンジンとしては上述の通りURLバリデーションチェックを行っていますが、同様のチェックがユーザコンテンツで必要となる場合があります。

たとえば、ユーザコンテンツでの画面遷移処理（例：「imwCallOriginalPagePath」を利用して一覧へ戻る）における遷移先パスが何らかのタイミング（ユーザコンテンツ間での画面遷移の最中など）で意図しないURLに改ざんされた場合、改ざん後のURLへの遷移を防止するためには、URLバリデーションチェック処理を独自に実装していただく必要があります。

チェックには「SafeUrlManager#isSafe(String url)」が利用可能です。

APIの詳細については「[APIドキュメント](#)」を参照してください。

また、セーフURLの設定方法については「[テナント管理者操作ガイド](#)」を参照してください。

「SafeUrlManager#isSafe(String url)」を利用したURLバリデーションチェックのサンプル実装（スクリプト開発モデル）を示します。

```

1  var safeUrlManager = new SafeUrlManager();
2  var result = safeUrlManager.isSafe(url);
3  if (result.error) {
4      // URLチェック実行に失敗した場合の処理
5      Transfer.toErrorPage(
6          {
7              "title": "URLバリデーションエラー",
8              "message": "URLチェック実行に失敗しました。管理者に連絡してください。"
9          }
10     );
11 }
12 if (!result.data) {
13     // 指定したURLがセーフURLリストに存在しない場合の処理
14     Transfer.toErrorPage(
15         {
16             "title": "URLバリデーションエラー",
17             "message": "安全ではないURLが指定されています。管理者に連絡してください。"
18         }
19     );
20 }

```

ユーザコンテンツと連続処理／連続確認の連携方法

ここで記載している内容は、次の観点において共通です。

- 開発モデル

「workflowOpenPage」タグの属性「imwNext～」を指定してIM-Workflow処理後に任意の画面（呼出元一覧画面以外の画面）に遷移した場合、または「workflowOpenPage」タグの属性「imwNext～」を指定せずにIM-Workflow処理後の画面遷移を行わない場合の、ユーザコンテンツと連続処理／連続確認の連携方法について説明します。

なお、IM-Workflow スマートフォンでは、連続処理／連続確認機能が存在しないため、ここで記載の事項を意識する必要はありません。

連続処理／連続確認を継続実行する

連続処理、連続確認を継続し、次の案件ノードに対応するユーザコンテンツ画面に遷移するためには、次の実装を行ってください。

- 一覧から渡された「imwCallOriginalPagePath」が指し示す画面に遷移してください。
- 一覧から渡された「imwCallOriginalParams」を遷移先画面へのリクエストパラメータとして設定してください。

連続処理／連続確認を中断する

連続処理、連続確認を中断し、一覧から渡された「imwCallOriginalPagePath」が指し示す画面に遷移するためには、「imwCallOriginalPagePath」への画面遷移の前に、セッションからクライアント固有情報を削除してください。

セッションキーは、「**IMW_LAST_PROCESSED_MATTER_INFO_IN_SERIAL**」です。

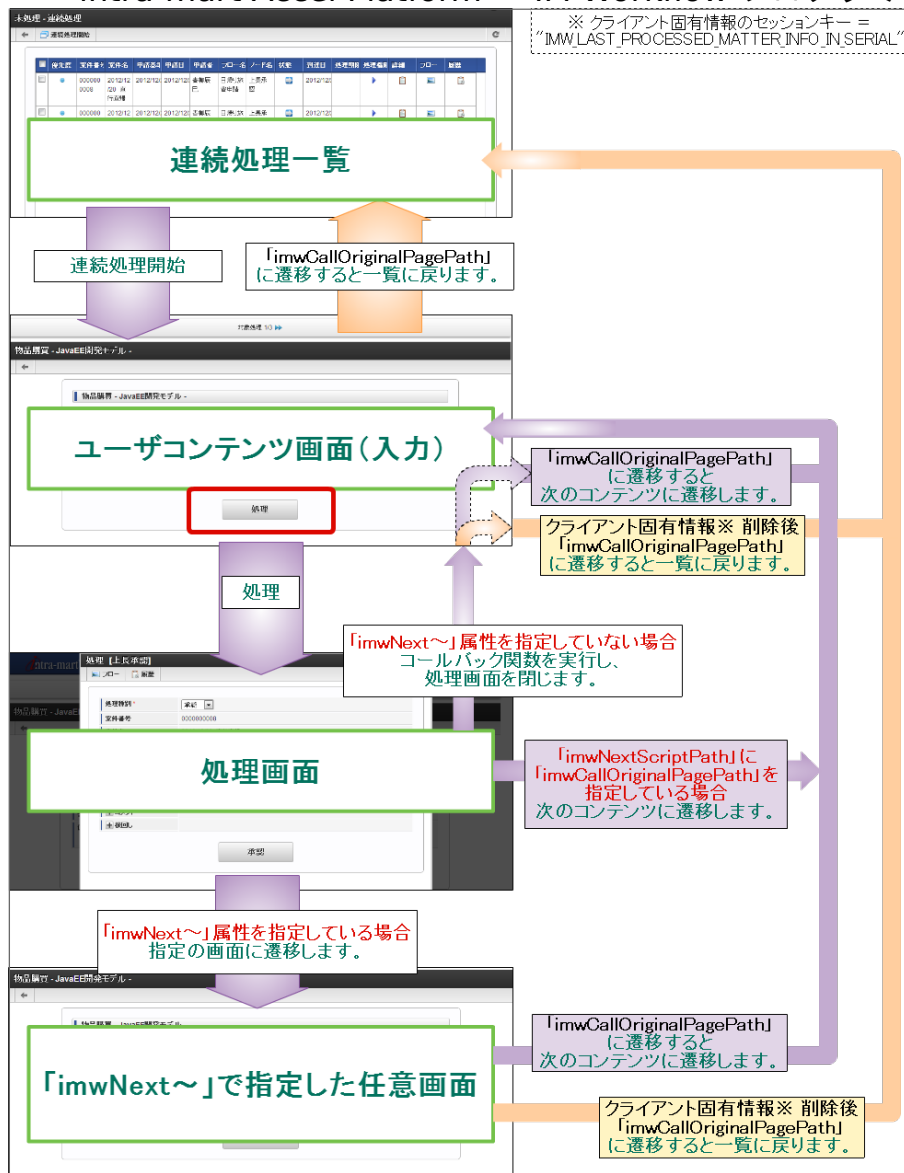
スクリプト開発モデルでセッションからクライアント固有情報を削除する場合は、次のメソッドを利用します。

```
Client.remove(Strign key)
```

JavaEE開発モデルでセッションからクライアント固有情報を削除する場合は、次のメソッドを利用します。

```
HttpSession.removeAttribute(java.lang.String name)
```

連続処理の場合の画面遷移を図示します。なお、連続確認の場合も画面遷移は同様です。



PC版ユーザコンテンツをスマートフォン用画面としても利用する

ここで記載している内容は、次の観点で共通です。

- 開発モデル

PC版ユーザコンテンツとして作成した画面を、スマートフォン用画面として動作させる方法を説明します。
この方法を採用すると、PC版ユーザコンテンツとスマートフォン版ユーザコンテンツをひとつの画面でまかなうことも可能です。

ただし、スマートフォン端末でPC版ユーザコンテンツを表示した場合、さまざまな制限事項があります。
そのため、PC版のユーザコンテンツとスマートフォン版のユーザコンテンツは、それぞれ独自に実装することを推奨します。

詳しくは「リリースノート」の制限事項を参照してください。

必要な作業

マスタ設定と、実装の修正を行う必要があります。

Contents

- マスタ定義のスマートフォン用画面設定を行う
- クライアントタイプをPCに切り替える
- 補足修正

マスタ定義のスマートフォン用画面設定を行う

「サイトマップ」-「マスタ定義-コンテンツ定義」より、設定対象のコンテンツ定義に対し「画面」を選択しスマートフォン用の画面定義を新規作成、または編集してください。

画面パスとして、スマートフォン用の画面として利用するPC版ユーザコンテンツを指定してください。

以降、必要に応じて、フロー定義の個別設定などを行ってください。

マスタ定義の新規作成、編集手順は「[IM-Workflow 管理者操作ガイド](#)」を参照してください。

以上を行ったうえで、スマートフォン端末で対象のフローの申請画面を表示すると、PC版ユーザコンテンツが表示されます。

ただし、この状態ではPC版ユーザコンテンツにスマートフォン用の画面テーマが適用されてしまい、レイアウトが崩れてしまう場合があります。

そこで、PC版ユーザコンテンツの実装に対して修正を行います。

クライアントタイプをPCに切り替える

ユーザコンテンツの実装において、クライアントタイプをPCに切り替える必要があります。

画面表示を行う際のサーバサイドロジックにおいて、ClientTypeSwitcher.oneTimeSwitchTo を利用し、ユーザコンテンツとして表示する画面のクライアントタイプを無条件でPCに切り替えてください。

```
ClientTypeSwitcher.oneTimeSwitchTo("pc");
```

ClientTypeSwitcher について、詳細は「[APIドキュメント](#)」を参照してください。

実装の修正を行う対象は、スマートフォン用画面として動作させるPC版ユーザコンテンツすべてです。

以上を行うことで、レイアウトが崩れることなくPC版ユーザコンテンツをスマートフォン端末で表示することができます。

この状態で、IM-Workflow が提供する案件の各処理画面（GreyBox上に表示される画面）が正常に表示されない（画面が表示されない、画面レイアウトが崩れる）場合のみ、以降の作業を行ってください。

補足修正

ワークフロー処理を実行する画面を表示するためのClient-side JavaScript API 「workflowOpenPage」の引数として、各種一覧画面からリクエストパラメータとして受け取った「画面種別 (imwPageType)」をそのまま受け渡している場合、修正が必要です。

クライアントタイプがスマートフォンの場合、各種一覧からは画面種別としてスマートフォン用画面の値が受け渡されます。

「workflowOpenPage」の引数には、PC用の画面種別の値を受け渡してください。

画面種別のクライアントタイプ別対応は下表のとおりです。

画面種別	申請	一時保存	申請 (起票案件)	再申請	処理	確認
クライアントタイプ						
PC	0	1	2	3	4	5
スマートフォン	10	11	12	13	14	15

ユーザコンテンツ画面への不正な直接アクセスを抑制する

ここで記載している内容は、次の観点で共通です。

- クライアントタイプ

IM-Workflow 標準機能では、IM-Workflow の各種一覧画面からユーザコンテンツ画面に遷移することができます。

この場合、IM-Workflow の標準機能は、ログインユーザが対象のコンテンツ画面の表示権限を保持しているか判定を行い、権限がない場合はエラー画面を表示します。

上記の通常遷移時以外の場合、IM-Workflow の標準機能によるユーザコンテンツ画面の表示権限の判定が行われません。

例えば、ユーザコンテンツ画面のURLに直接アクセスが行われた場合、IM-Workflow の標準機能による表示権限の判定が行われないため、ユーザコンテンツ画面のつくりによっては、表示権限を持たないユーザにユーザコンテンツ画面の内容を閲覧されてしまう可能性があります。

上記の状態でも、ユーザコンテンツ画面の表示後に各種処理（申請、承認など）を実行するタイミングでは、IM-Workflow の標準機能による処理権限の判定が行われるため、不正な処理が実行されてしまうことはありません。

ただし、表示権限のないユーザにユーザコンテンツ画面を閲覧されてしまうことが運用上の問題となる場合には、以降の対応を行うことにより、ユーザコンテンツ画面への不正な直接アクセスを抑制することが可能です。

対象者

以下の対応を検討している方を対象としています。

- ユーザコンテンツ画面へのアクセス権限について、セキュリティ強化を図りたい方
- intra-mart Accel Platform の認可機構を利用し、ユーザコンテンツ画面の表示権限を制御したい方
- すでに実施済みのセキュリティ対応について、IM-Workflow 標準の方法に切替えたい方

対象パス種別

ユーザコンテンツ定義の画面定義において、以下のパス種別として登録する画面を対象としています

- JavaEE開発モデル
- URL(旧名称:JSP or Servlet)

パス種別が「スクリプト開発モデル」であるユーザコンテンツ画面については、スクリプト開発のセキュアな機構で直接のアクセスが抑止されているため、対応の必要はありません。

対応方法

Contents

- [認可設定](#)
- [ユーザコンテンツ画面の追加開発（カスタマイズ）](#)

対応方法としては、以下のいずれかを選択可能です。

1. 認可設定
2. ユーザコンテンツ画面の追加開発（カスタマイズ）

どちらの方法を選択すべきかは、下表を参照してください。

要件	推奨する対応方法
ユーザコンテンツ画面の実装を改修できない	認可設定
アクセス権設定を認可機構で統一的に扱いたい	認可設定
IM-Workflowの標準機能と同等のユーザコンテンツ画面表示権限判定を実行したい	ユーザコンテンツ画面の追加開発（カスタマイズ）

以降では、それぞれの対応方法の詳細について説明します。
運用形態や影響範囲を考慮の上、適当な方法を選択してください。

認可設定

認可機構により、ユーザコンテンツ画面を「リソース」として登録し、アクセス権設定を行います。
認可の仕様については「[認可仕様書](#)」を参照してください。

認可設定による対応の特徴は以下の通りです。

- ユーザの権限を認可機構で集約して管理することが可能です。
- ユーザコンテンツ画面の実装の改修は不要です。
- IM-Workflow のルート定義で設定される処理対象者を包含する範囲で認可設定を行う必要があります。
 - 例として、同一の申請用ユーザコンテンツ画面を、フローAとフローBで流用している場面を想定します。
 - フローAはルートAを利用しており、申請ノードの処理対象者は「サンプル課11」です。
 - フローBはルートBを利用しており、申請ノードの処理対象者は「サンプル部門02」です。
 - この場合、申請用ユーザコンテンツ画面の認可設定としては、「サンプル課11」と「サンプル部門02」からの実行を許可する設定を行う必要があります。

以降では、IM-Workflow のコンテンツ定義における「パス種別」ごとに、認可設定を行う際の参考となるドキュメントを紹介します。

パス種別「JavaEE開発モデル」の場合

「[移行ガイド](#)」の「[im-JavaEE Framework](#)」の認可設定部分を参照してください。

パス種別「URL」（旧名称:JSP or Servlet）の場合

TERASOLUNA Server Framework for Java (5.x) を利用して実装している場合、「[TERASOLUNA Server Framework for Java \(5.x\) プログラミングガイド](#)」の「[認可](#)」を参照してください。

ユーザコンテンツ画面の追加開発（カスタマイズ）

IM-Workflow が提供するタグライブラリ、またはAPIを利用し、ユーザコンテンツ画面の表示権限を判定します。

ユーザコンテンツ画面の表示権限とは、特定の案件を処理、または参照する場合に利用されるユーザコンテンツ画面を、IM-Workflow 標準の各種一覧画面（フロー一覧、未処理一覧など）から表示することのできる権限のことを指します。

ユーザコンテンツ画面の追加開発による対応の特徴は以下の通りです。

- IM-Workflow の標準機能と同等のユーザコンテンツ表示権限判定を行うことが可能です。
- ユーザコンテンツ画面の実装の改修が必要です。

対応方法としては、以下のいずれかを選択可能です。

- タグライブラリによる対応
- APIによる対応

タグライブラリによる対応

クライアントタイプ別で、ユーザコンテンツ画面の表示権限判定用タグライブラリが用意されています。

- クライアントタイプ=PC
 - 「workflowUserContentsAuth」
- クライアントタイプ=スマートフォン
 - 「spWorkflowUserContentsAuth」

ユーザコンテンツ画面で上記のタグライブラリを利用するのみで、ユーザコンテンツ画面の表示権限の判定を行うことが可能です。表示権限がない場合、HTTP 403エラーが発生します。

タグライブラリによる対応を行う場合は、「[APIドキュメント](#)」を併せて参照してください。

推奨実装

以下のルールで実装を行うことを推奨します。

1. ユーザコンテンツ画面に「[リクエストパラメータ](#)」として受け渡されたパラメータを、すべてリクエストスコープの属性として格納します。
2. タグライブラリを引数省略の形式で利用します。

ユーザコンテンツ画面が複数画面構成の場合、追加で下記実装を行うことを推奨します。

3. 「[リクエストパラメータ](#)」としてユーザコンテンツ画面に受け渡されたパラメータを、ユーザコンテンツ画面間を遷移する際に引き回します。そのうえで、上記の1、2の実装を各ユーザコンテンツ画面で行います。

上記のルールを採用することにより、以下の実装上のメリットがあります。

- タグライブラリを統一的な手法で組み込むことが可能です。
- ひとつのユーザコンテンツ画面が複数の画面種別に対応した実装となっている場合でも、画面種別の差異によってタグライブラリに指定するパラメータを切り替える必要がなくなります。

実装例

IM-Workflow JavaEE開発モデルの下記サンプルをもとに、推奨実装例1、2を紹介します。

- クライアントタイプ=PCの場合
 - アプリケーションID : imw_sample_purchase
 - サービスID : apply
- クライアントタイプ=スマートフォンの場合
 - アプリケーションID : imw_sp_sample_purchase
 - サービスID : apply

このサンプルでは、以下の画面種別に対応しています。

- 申請画面
- 一時保存画面
- 申請（起票案件）画面
- 再申請画面

それでは、順を追って実装例を示します。

1. ユーザコンテンツ画面に「[リクエストパラメータ](#)」として受け渡されたパラメータを、すべてリクエストスコープの属性として格納します。

HttpServletRequest#setAttribute(String, String) を利用し、リクエストスコープの属性にパラメータを格納します。

※サンプルではあらかじめ実装されています。

```
<%サンプルプログラムディレクトリ%
/jp/co/intra_mart/sample/workflow/purchase/controller/service/ApplyServiceTransition.java>
```

```

1 package jp.co.intra_mart.sample.workflow.purchase.controller.service;
2
3 import javax.servlet.http.HttpServletRequest;
4
5 import jp.co.intra_mart.framework.base.service.DefaultTransition;
6 import jp.co.intra_mart.framework.base.service.ServicePropertyException;
7 import jp.co.intra_mart.framework.base.service.TransitionException;
8
9 public class ApplyServiceTransition extends DefaultTransition {
10
11     @Override
12     public String getNextPage() throws ServicePropertyException, TransitionException {
13         final ApplyServiceResult serviceResult = (ApplyServiceResult) getResult();
14         return getNextPagePath(serviceResult.getNextPageServiceId());
15     }
16
17     @Override
18     public void setInformation() throws TransitionException {
19         final HttpServletRequest request = getRequest();
20         final ApplyServiceResult serviceResult = (ApplyServiceResult) getResult();
21         // 受け渡されたパラメータをリクエストスコープに設定します。
22         request.setAttribute("imwGroupld", serviceResult.getImwGroupld());
23         request.setAttribute("imwUserCode", serviceResult.getImwUserCode());
24         request.setAttribute("imwPageType", serviceResult.getImwPageType());
25         request.setAttribute("imwUserDataId", serviceResult.getImwUserDataId());
26         request.setAttribute("imwSystemMatterId", serviceResult.getImwSystemMatterId());
27         request.setAttribute("imwNodeId", serviceResult.getImwNodeId());
28         request.setAttribute("imwArriveType", serviceResult.getImwArriveType());
29         request.setAttribute("imwAuthUserCode", serviceResult.getImwAuthUserCode());
30         request.setAttribute("imwApplyBaseDate", serviceResult.getImwApplyBaseDate());
31         request.setAttribute("imwContentsId", serviceResult.getImwContentsId());
32         request.setAttribute("imwContentsVersionId", serviceResult.getImwContentsVersionId());
33         request.setAttribute("imwRouteId", serviceResult.getImwRouteId());
34         request.setAttribute("imwRouteVersionId", serviceResult.getImwRouteVersionId());
35         request.setAttribute("imwFlowId", serviceResult.getImwFlowId());
36         request.setAttribute("imwFlowVersionId", serviceResult.getImwFlowVersionId());
37         request.setAttribute("imwCallOriginalParams", serviceResult.getImwCallOriginalParams());
38         request.setAttribute("imwCallOriginalPagePath", serviceResult.getImwCallOriginalPagePath());
39
40         request.setAttribute("item_name", serviceResult.getItemName());
41         request.setAttribute("item_amount", serviceResult.getItemAmount());
42         request.setAttribute("item_price", serviceResult.getItemPrice());
43         request.setAttribute("item_total", serviceResult.getItemTotal());
44         request.setAttribute("item_comment", serviceResult.getItemComment());
45     }
46 }

```

2. タグライブラリを引数省略の形式で利用します

タグライブラリを画面実装に追加します。

- クライアントタイプ=PCの場合

< (展開したwar) /sample/im_workflow/purchase/apply.jsp>

```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ taglib prefix="imart2ee" uri="http://www.intra-mart.co.jp/taglib/core/framework" %>
3 <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
4 <%@ taglib prefix="imart" uri="http://www.intra-mart.co.jp/taglib/core/standard" %>
5 <%@ taglib prefix="workflow" uri="http://www.intra-mart.co.jp/taglib/imw/workflow" %>
6 <imart2ee:HelperBean id="bean" class="jp.co.intra_mart.sample.workflow.purchase.controller.view.CommonHelperBean"/>
7
8 <!-- タグライブラリ (引数省略) -->
9 <workflow:workflowUserContentsAuth />
10
11 <imui:head>
12 .
13 .
14 .

```

- クライアントタイプ=スマートフォンの場合

< (展開したwar) /sample/im_workflow_smartphone/purchase/apply.jsp>

```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ taglib prefix="imartj2ee" uri="http://www.intra-mart.co.jp/taglib/core/framework" %>
3 <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
4 <%@ taglib prefix="imart" uri="http://www.intra-mart.co.jp/taglib/core/standard" %>
5 <%@ taglib prefix="workflow" uri="http://www.intra-mart.co.jp/taglib/imw/workflow-smartphone" %>
6 <imartj2ee:HelperBean id="bean" class="jp.co.intra_mart.sample.workflow.purchase.controller.view.CommonHelperBean"/>
7
8 <!-- タグライブラリ (引数省略) -->
9 <workflow:spWorkflowUserContentsAuth />
10
11 <imui:head>

```

APIによる対応

タグライブラリによる対応では実現できない要件がある場合は、ユーザコンテンツ画面の表示権限の判定APIを利用することで、任意の動作をさせることが可能です。

具体的には、次のような場合を想定します。

- 業務ロジックのとの兼ね合いで、タグライブラリを利用できない場合
- 表示権限がないと判定された際、HTTP 403エラーではなく任意の処理を行いたい場合

対応するAPIは「jp.co.intra_mart.foundation.workflow.util.auth.WorkflowAuthUtil」です。

権限判定の結果は boolean 値で返却されるため、結果をうけて任意の処理を行うことが可能です。

詳細は「[APIドキュメント](#)」を参照してください。

動的処理対象者設定機能

「動的処理対象者設定」機能とは、申請/処理画面のフロー設定項目をユーザコンテンツ画面からのリクエストパラメータで設定できる機能です。

ここで記載している内容は、次の観点で共通です。

- 開発モデル
- クライアントタイプ

なお、当機能は以下のバージョン以降で利用可能です。

- PC版 IM-Workflow 2014 Winter(Iceberg) 8.0.9 PATCH 001
- スマートフォン版 IM-Workflow 2015 Summer(Karen) 8.0.11

機能概要

「動的処理対象者設定」機能では、以下を実現可能です。

- 処理対象者の決定
 - ビジネスロジックによって決定した処理対象者を、標準処理画面で設定可能なノードに反映する
- 処理対象者検索時の暗黙条件の指定
 - 標準処理画面で設定可能なノードにおいて、利用者が処理対象者を検索・設定する際の暗黙条件を指定し、検索結果の絞り込みを行う

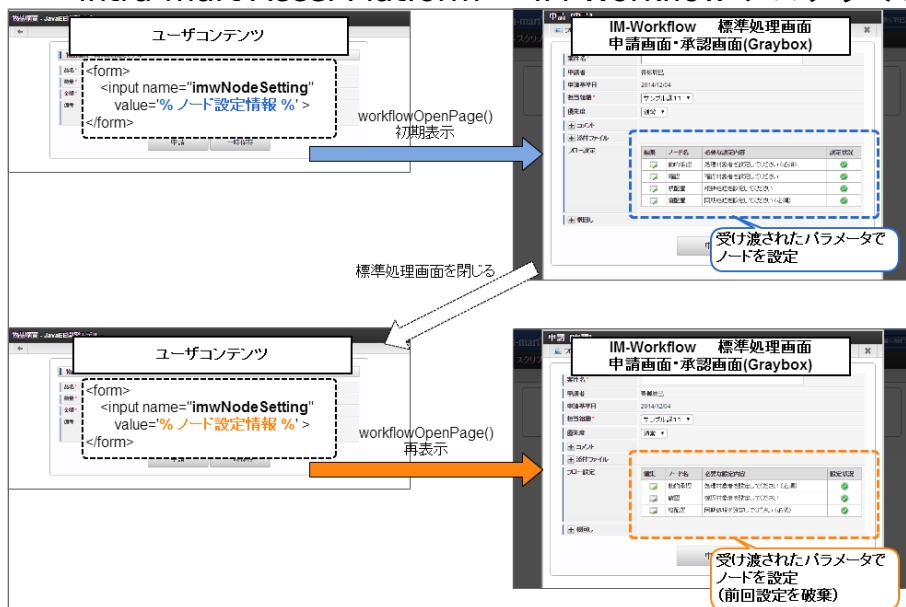
当機能によって設定が可能なノードは以下の通りです。

- 動的承認ノード
- 確認ノード
- 横配置ノード
- 縦配置ノード

利用方法

IM-Workflow の標準処理画面を表示する際に、ユーザコンテンツからパラメータを送信することで、フロー設定、およびノード設定を行います。

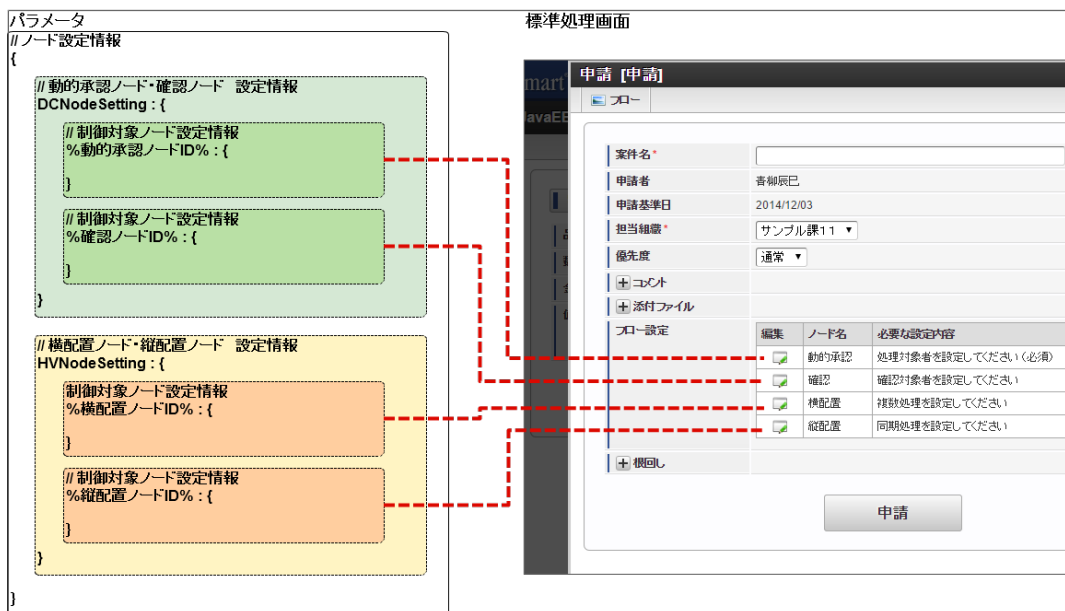
パラメータの送信方法は、設定用パラメータオブジェクトをJSON文字列に変換し、「imwNodeSetting」というキーで標準処理画面に受け渡します。



標準処理画面を閉じた後、ユーザコンテンツから標準処理画面を再度表示する際に改めてパラメータを受け渡した場合、前回の設定情報を破棄し、新しく受け渡されたパラメータによって設定されます。

標準処理画面を閉じた後、ユーザコンテンツから標準処理画面を再度表示する際にパラメータを受け渡さない場合、前回の設定情報を保持して再表示されます。

パラメータとフロー設定、ノード設定の概念図を以下に示します。



設定対象のノード単位で情報を作成し、ノード種別（動的承認ノード・確認ノード / 横配置ノード・縦配置ノード）でまとめ、最終的にひとつのパラメータとして生成し、標準処理画面に受け渡すことで各種設定を行います。

当機能を利用するノードと設定される対象のノードは、あらかじめフロー定義のノード設定において、処理対象者設定可能ノードの設定が行われている必要があります。

つまり、フロー設定を行う場合は、当機能を利用する・しないに関わらず、上図のように標準処理画面でフロー設定としてノードが表示される状態となるよう、フロー定義のノード設定が行われている必要があります。

処理対象者設定可能ノードの設定が行われていないノードに対してパラメータを送信しても、パラメータは無視され、当機能は実行されません。

以降の章で、具体的な利用方法やパラメータの詳細について説明します。

利用例

当機能の利用例を、ユーザコンテンツから送信するパラメータの例とともに紹介します。

なお、この章では申請時の例のみ掲載していますが、承認時にも当機能を利用することは可能で、申請時と流れは同様です。

また、この章ではスクリプト開発モデルでの実装例を紹介します。

処理対象者設定

動的承認ノードと横配置ノードに対し、処理対象者を設定する場合を例示します。

Contents

- 動的承認ノードの処理対象者をシステムで決定する
- 横配置ノードの配置数、処理対象者をシステムで決定する

動的承認ノードの処理対象者をシステムで決定する

ユーザコンテンツ側で決定した処理対象者を、動的承認ノードに反映します。
また、標準処理画面からは動的承認ノードの設定を行わせない制御を実施します。

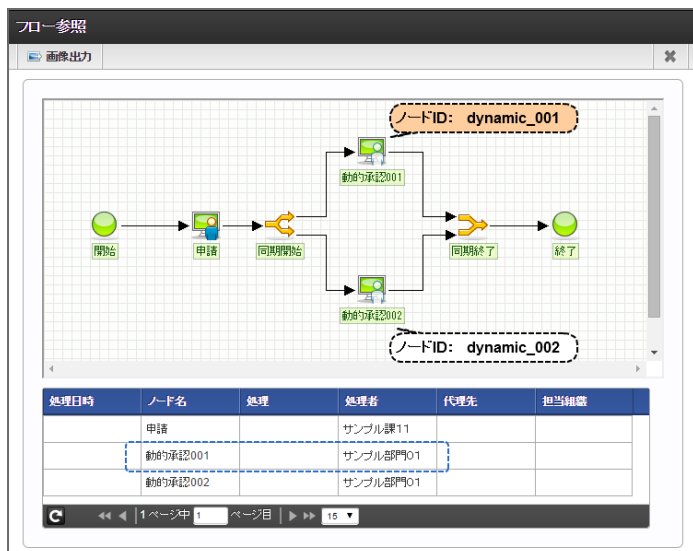
前提

申請時に動的承認ノードの設定を行います。

利用するフローは次の通りです。動的承認ノードがふたつ存在し、処理対象者は両方とも「サンプル部門01」に設定されています。

また、ふたつの動的承認ノードの設定を申請ノードで行えるようフロー定義を設定します。

今回は、「動的承認001」（ノードID：dynamic_001）を対象に設定を行います。



実装例

設定用パラメータを生成します。

```

1  var nodeSetting = {
2
3  "DCNodeSetting" : {
4
5  "dynamic_001" : { // 設定対象のノードIDをプロパティ名とする
6
7  "displayFlag" : false, // 画面表示をしない
8
9  "processTargetConfigs" : [ // 任意の処理対象者を指定
10 {
11 // ユーザ : maruyama
12 "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
13 "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.user",
14 "parameter" : "maruyama"
15 },
16 {
17 // ロール : IM-Workflow ユーザ
18 "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
19 "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.role",
20 "parameter" : "im_workflow_user"
21 }
22 ]
23 }
24 }
25 };

```

上記のパラメータをJSON文字列に変換します。スクリプト開発モデルでは、ImJson#toJSONString メソッドを利用します。

```
ImJson.toJSONString(nodeSetting);
```

JSON文字列に変換したパラメータを、「imwNodeSetting」というキーで標準処理画面に受け渡します。

ユーザコンテンツ画面で利用するタグライブラリ「workflowOpenPage」のボディ部にパラメータを定義するなどの対応を行い、標準処理画面が表示される際に生成したパラメータが受け渡されるよう実装してください。

- クライアントタイプ=PC の場合

```

1 <imart type="workflowOpenPage" >
2
3 <input type="hidden" name="imwNodeSetting" value="%JSON文字列に変換したパラメータ%" >
4
5 </imart>
    
```

- クライアントタイプ=スマートフォンの場合

```

1 <imart type="spWorkflowOpenPage" >
2
3 <input type="hidden" name="imwNodeSetting" value="%JSON文字列に変換したパラメータ%" >
4
5 </imart>
    
```

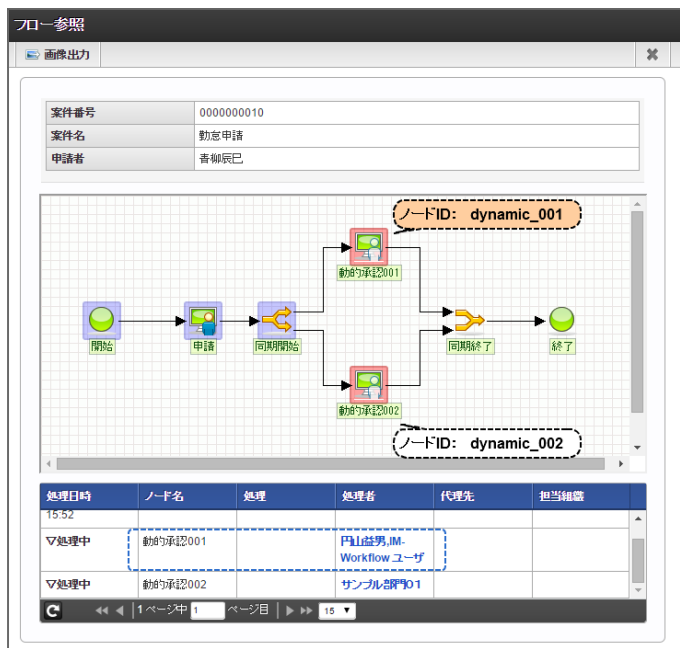
JSON文字列をクライアントに送信したりvalueとして設定する際は、必要に応じて適切なエスケープ処理を行ってください。

動作結果

標準処理画面に上述の「imwNodeSetting」パラメータを受け渡した場合、標準処理画面の「フロー設定」欄には、「動的承認001」ノードが表示されません。一方、パラメータで未指定の「動的承認002」ノードは画面上に表示されます。



申請を行います。
その結果、「動的承認001」ノードの処理対象者はパラメータで指定したとおりに設定されました。



横配置ノードの配置数、処理対象者をシステムで決定する

ユーザコンテンツ側で決定した処理対象者を、横配置ノードに反映します。
また、標準処理画面からは動的承認ノードの設定を行わせない制御を実施します。

前提

申請時に横配置ノードの設定を行います。

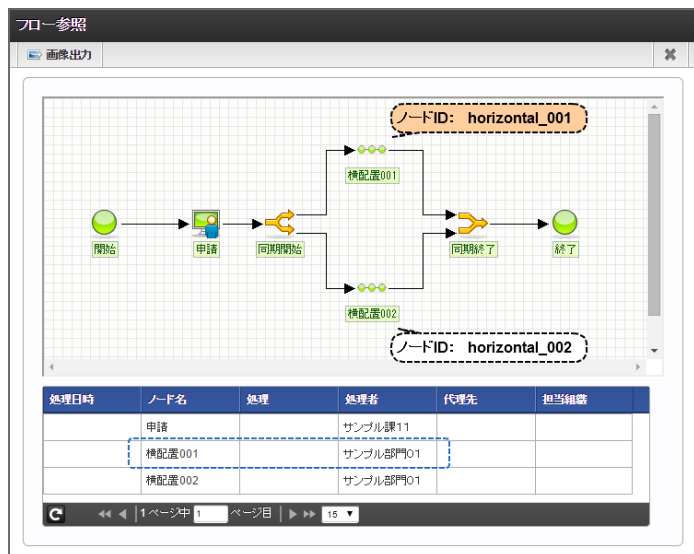
利用するフローは次の通りです。

横配置ノードがふたつ存在し、処理対象者は両方とも「サンプル部門01」に設定されています。

横配置ノードの割当可能ノード数は、ふたつとも 最小=1、最大=3 に設定されています。

また、ふたつの横配置ノードの設定を申請ノードで行えるようフロー定義を設定します。

今回は、「横配置001」（ノードID: horizontal_001）を対象に設定を行います。



実装例

設定用パラメータを生成します。

```

1  var nodeSetting = {
2
3  "HVNodeSetting" : {
4
5    "horizontal_001" : { // 設定対象のノードIDをプロパティ名とする
6
7    "displayFlag" : false, // 画面表示をしない
8
9    "matterNodeExpansions" : [ // ノード展開情報を指定
10
11    // ひとつめの展開ノード
12    {
13      "nodeName" : "node_name_001", // ノード名
14
15      "processTargetConfigModel" : [ // 任意の処理対象者を指定
16      {
17        // 組織：サンプル課11
18        "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
19        "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
20        "parameter" : "comp_sample_01^comp_sample_01^dept_sample_11"
21      },
22      {
23        // 組織：サンプル課12
24        "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
25        "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
26        "parameter" : "comp_sample_01^comp_sample_01^dept_sample_12"
27      }
28    ]
29  },
30  // ふたつめの展開ノード
31  {
32    "nodeName" : "node_name_002", // ノード名
33
34    "processTargetConfigModel" : [ // 任意の処理対象者を指定
35    {
36      // 組織：サンプル課21
37      "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
38      "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
39      "parameter" : "comp_sample_01^comp_sample_01^dept_sample_21"
40    },
41    {
42      // 組織：サンプル課22
43      "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
44      "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
45      "parameter" : "comp_sample_01^comp_sample_01^dept_sample_22"
46    }
47  ]
48  }
49  ]
50  }
51  }
52  };

```

パラメータをJSON文字列に変換し、「imwNodeSetting」として標準処理画面に受け渡します。

動作結果

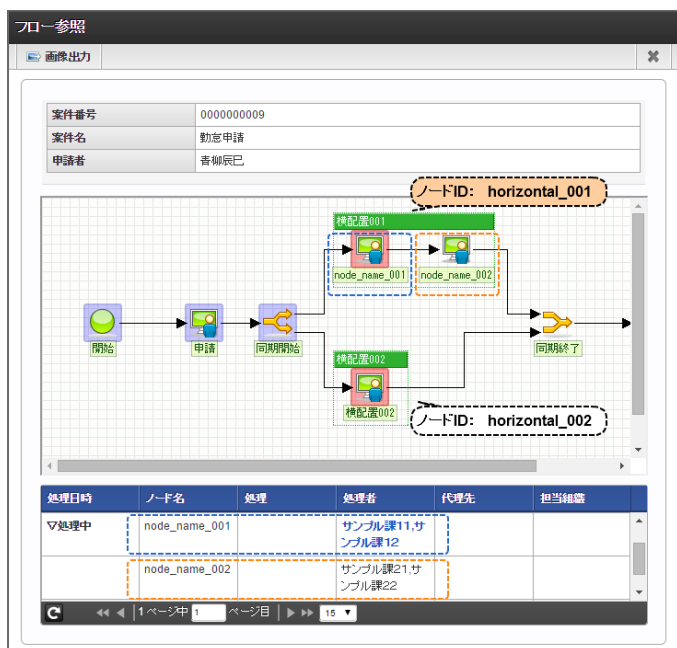
標準処理画面に上述の「imwNodeSetting」パラメータを受け渡した場合、標準処理画面の「フロー設定」欄には、「横配置001」ノードが表示されません。

一方、パラメータで未指定の「横配置002」ノードは画面上に表示されません。



申請を行います。

その結果、「横配置001」ノードの処理対象者はパラメータで指定したとおりに展開されました。



検索時の暗黙条件

動的承認ノードと横配置ノードに対し、検索時の暗黙条件を設定する場合を例示します。

Contents

- 動的承認ノードに設定可能な処理対象者を制限する
- 横配置ノードに設定可能な処理対象者を制限する

動的承認ノードに設定可能な処理対象者を制限する

標準処理画面から動的承認ノードの処理対象者を検索する際、ユーザコンテンツ側で決定した暗黙条件を適用します。

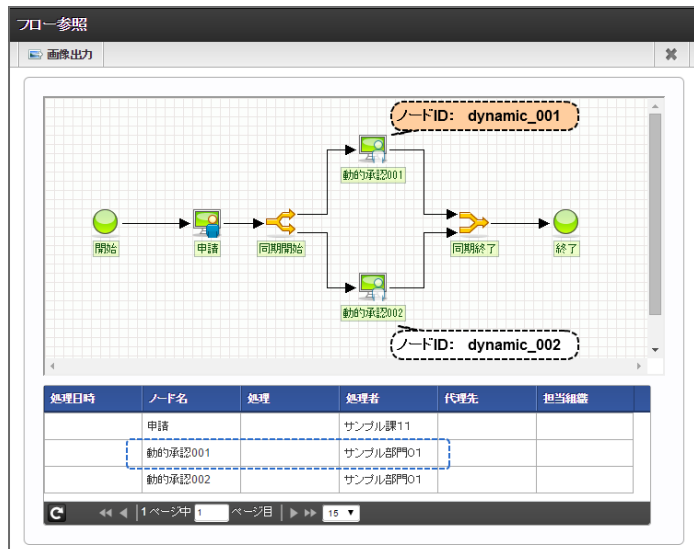
前提

申請時に動的承認ノードの設定を行います。

利用するフローは次の通りです。動的承認ノードがふたつ存在し、処理対象者は両方とも「サンプル部門01」に設定されています。

また、ふたつの動的承認ノードの設定を申請ノードで行えるようフロー定義を設定します。

今回は、「動的承認001」（ノードID: dynamic_001）を対象に設定を行います。



実装例

設定用パラメータを生成します。

```

1  var nodeSetting = {
2
3  "DCNodeSetting" : {
4
5    "dynamic_001" : { // 設定対象のノードIDをプロパティ名とする
6
7    "displayFlag" : true, // 画面表示をする
8
9    "searchCondition" : { // 処理対象者の検索時条件を指定
10
11    "criteria" : { // 暗黙条件を指定
12
13    "department_set_list" : [
14      {
15        // 組織 : サンプル部門01
16        "company_cd" : "comp_sample_01",
17        "department_set_cd" : "comp_sample_01",
18        "department" : {
19          "department_cd" : "dept_sample_10",
20          "compare" : "eq"
21        }
22      }
23    ]
24  }
25  },
26
27  "processTargetConfigs" : [] // 処理対象者を指定 (明示的に0件指定)
28  }
29  }
30  };

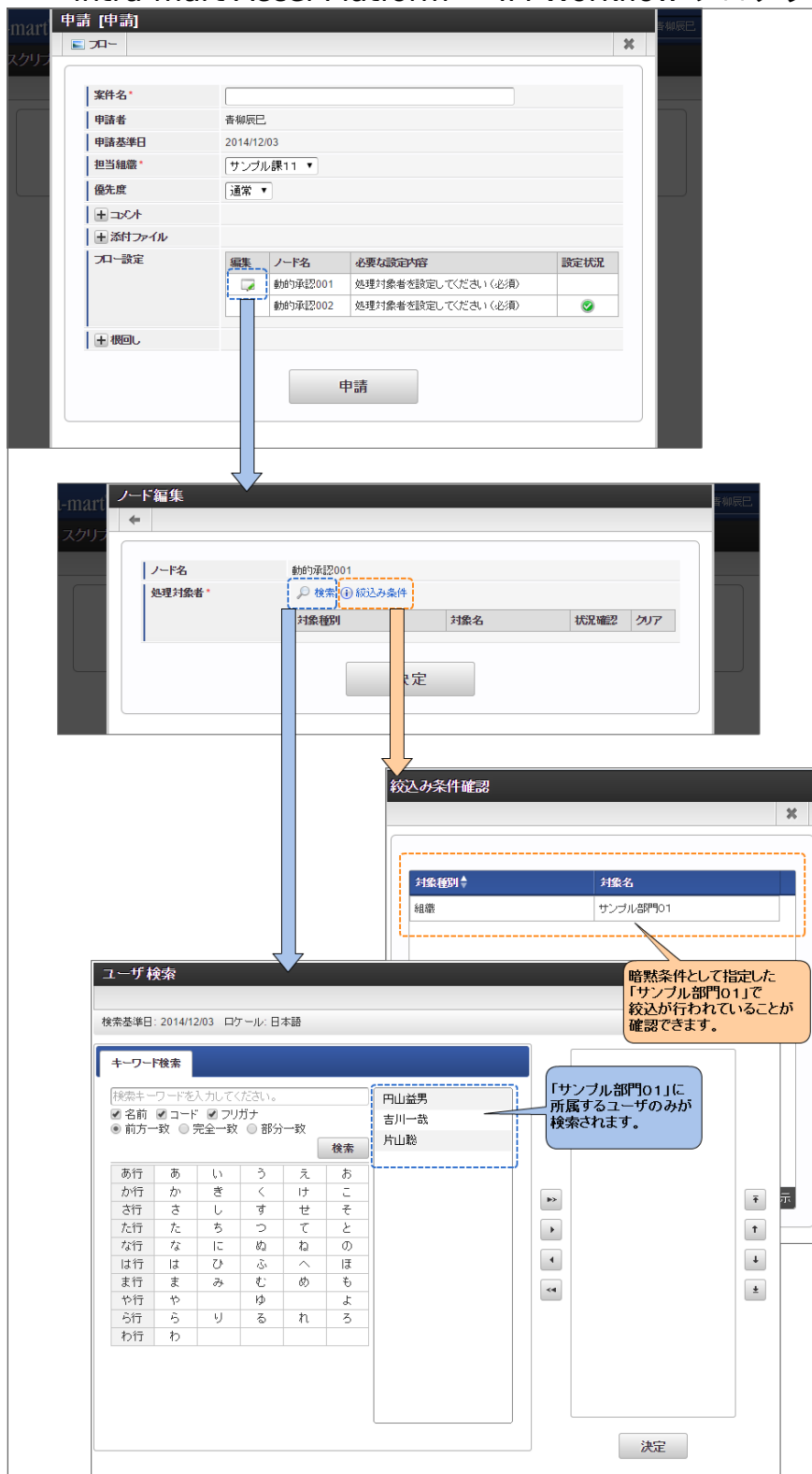
```

パラメータをJSON文字列に変換し、「imwNodeSetting」として標準処理画面に受け渡します。

動作結果

標準処理画面に上述の「imwNodeSetting」パラメータを受け渡した場合、標準処理画面の「フロー設定」欄には、「動的承認001」「動的承認002」ノードが表示されます。

「動的承認001」のノード編集画面を表示し、「検索」リンクを押下すると、「ユーザ検索 (キーワード タブ)」が表示されます。この画面で検索を実行すると、パラメータで指定した通り、「サンプル部門01」に所属するユーザのみが検索されます。



なお、パラメータによる設定を行っていない「動的承認002」の場合、検索結果に対する絞り込みは行われません。

横配置ノードに設定可能な処理対象者を制限する

標準処理画面から横配置ノードの処理対象者を検索する際、ユーザコンテンツ側で決定した暗黙条件を適用します。

前提

申請時に横配置ノードの設定を行います。

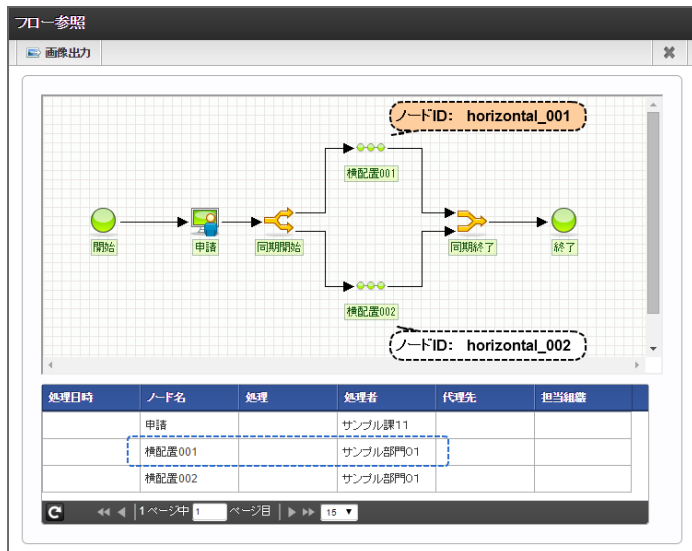
利用するフローは次の通りです。

横配置ノードがふたつ存在し、処理対象者は両方とも「サンプル部門01」に設定されています。

横配置ノードの割当可能ノード数は、ふたつとも 最小=1、最大=3 に設定されています。

また、ふたつの横配置ノードの設定を申請ノードで行えるようフロー定義を設定します。

今回は、「横配置001」（ノードID: horizontal_001）を対象に設定を行います。



実装例

設定用パラメータを生成します。

```

1  var nodeSetting = {
2
3  "HVNodeSetting" : {
4
5    "horizontal_001" : { // 設定対象のノードIDをプロパティ名とする
6
7    "displayFlag" : true, // 画面表示をする
8
9    "matterNodeExpansions" : [ // ノード展開情報を指定
10
11    // ひとつめの展開ノード
12    {
13      "nodeName" : "node_name_001", // ノード名
14
15      "searchCondition" : { // 処理対象者の検索時条件を指定
16
17      "criteria" : { // 暗黙条件を指定
18
19      "department_set_list" : [
20      {
21      // 組織：サンプル課11 / 役職：課長
22      "company_cd" : "comp_sample_01",
23      "department_set_cd" : "comp_sample_01",
24      "department" : {
25      "department_cd" : "dept_sample_11",
26      "compare" : "eq"
27      },
28      "post" : {
29      "post_cd" : "ps003",
30      "compare" : "eq"
31      }
32      }
33      ]
34      }
35      },
36
37      "processTargetConfigModel" : [] // 処理対象者を指定（明示的に0件指定）
38      },
39
40      // ふたつめの展開ノード
41      {
42      "nodeName" : "node_name_002", // ノード名
43
44      "searchCondition" : { // 処理対象者の検索時条件を指定
45
46      "criteria" : { // 暗黙条件を指定
47
48      "department_set_list" : [
49      {
50      // 組織：サンプル会社 / 役職：社長
51      "company_cd" : "comp_sample_01",
52      "department_set_cd" : "comp_sample_01",
53      "department" : {
54      "department_cd" : "comp_sample_01",
55      "compare" : "eq"
56      },
57      "post" : {
58      "post_cd" : "ps001",
59      "compare" : "eq"
60      }
61      }
62      ]
63      }
64      },
65
66      "processTargetConfigModel" : [] // 処理対象者を指定（明示的に0件指定）
67      }
68      ]
69      }
70      }
71      };

```

パラメータをJSON文字列に変換し、「imwNodeSetting」として標準処理画面に受け渡します。

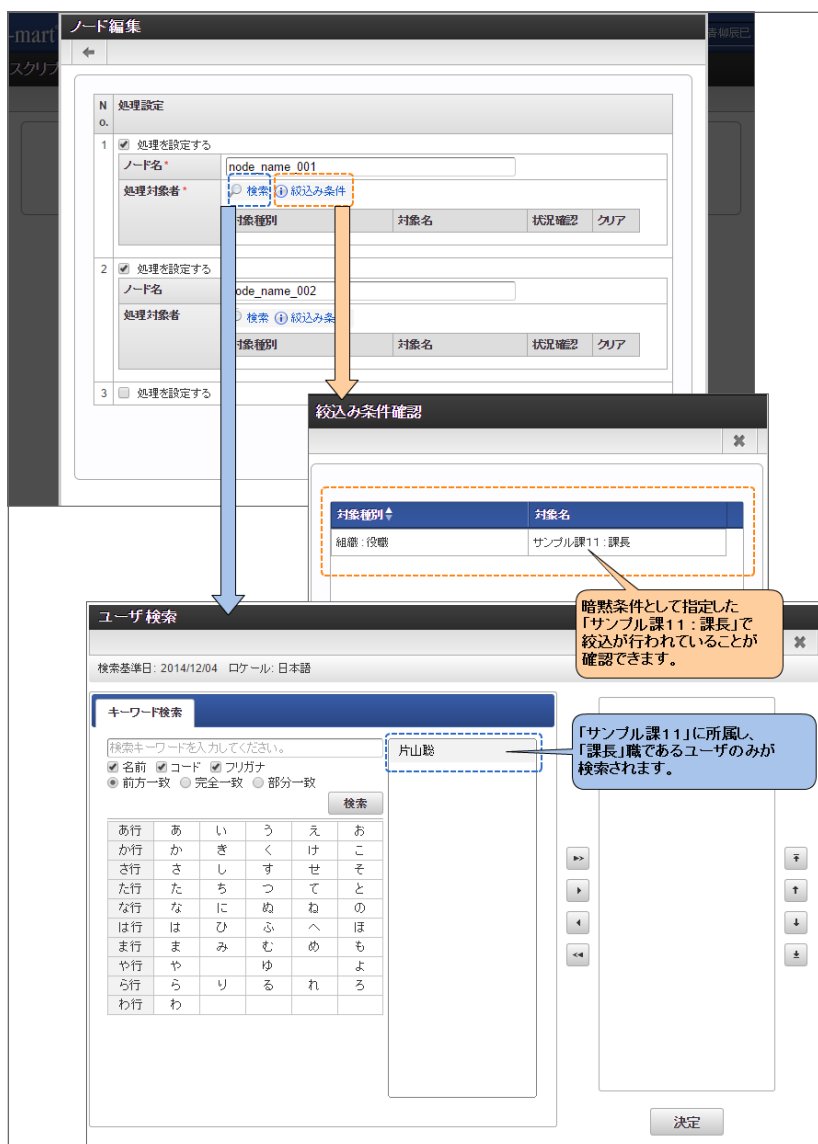
動作結果

標準処理画面に上述の「imwNodeSetting」パラメータを受け渡した場合、標準処理画面の「フロー設定」欄には、「横配置001」「横配置002」ノードが表示されます。

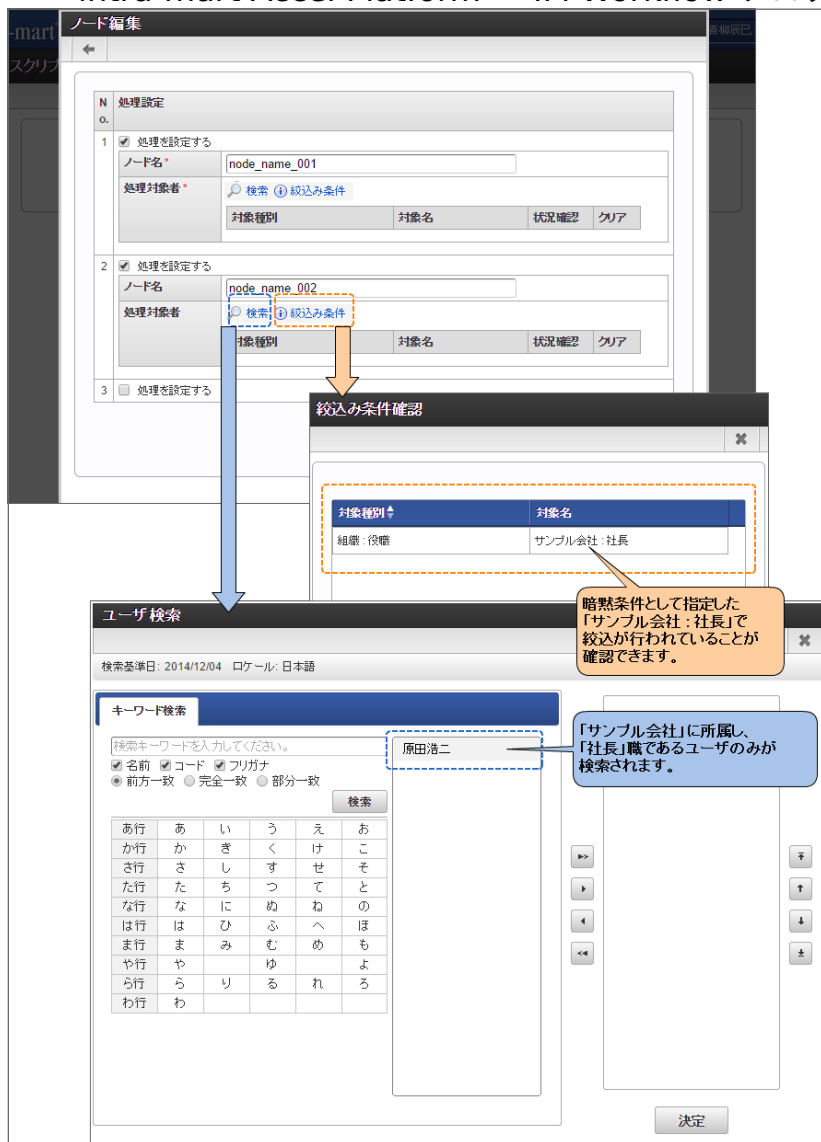


「横配置001」のノード編集画面を表示します。

まず、ひとつめの展開ノード「node_name_001」に対する「検索」リンクを押下すると、「ユーザ検索（キーワード タブ）」が表示されます。この画面で検索を実行すると、パラメータで指定した通り、「サンプル課11」に所属し、「課長」職であるユーザのみが検索されました。



次に、ふたつめの展開ノード「node_name_002」に対する「検索」リンクを押下すると、「ユーザ検索（キーワード タブ）」が表示されます。この画面で検索を実行すると、パラメータで指定した通り、「サンプル会社」に所属し、「社長」職であるユーザのみが検索されました。



なお、パラメータによる設定を行っていない「横配置002」の場合は、検索結果に対する絞り込みは行われません。

パラメータ詳細

機能を利用するうえで指定が必要なパラメータの詳細について記述します。

構造概要

パラメータは、個々のノードに対する設定情報オブジェクトを、ノード種別（動的承認ノード・確認ノード / 横配置ノード・縦配置ノード）で取りまとめた構造です。

個々のノード設定情報は、標準処理画面上で「フロー設定」として表示される各ノードに関連付けられます。

コードで表現すると、以下の通りです。ノードIDは例です。

```

1  {
2  // 動的承認ノード・確認ノード 設定情報
3  "DCNodeSetting" : {
4
5  // 設定対象ノードIDをプロパティキーとして指定（ノードIDが" dynamic_001"の場合の例）
6  "dynamic_001" : {
7  // 設定用の各種パラメータを指定
8
9  },
10 // 設定対象ノードIDをプロパティキーとして指定（ノードIDが" confirm_001"の場合の例）
11 "confirm_001" : {
12 // 設定用の各種パラメータを指定
13
14 }
15 },
16
17 // 横配置ノード・縦配置ノード 設定情報
18 "HVNodeSetting" : {
19
20 // 設定対象ノードIDをプロパティキーとして指定（ノードIDが" horizontal_001"の場合の例）
21 "horizontal_001" : {
22 // 設定用の各種パラメータを指定
23
24 },
25 // 設定対象ノードIDをプロパティキーとして指定（ノードIDが" vertical_001"の場合の例）
26 "vertical_001" : {
27 // 設定用の各種パラメータを指定
28
29 }
30 }
31 }

```

個々のノード設定情報の構造はノード種別によって異なります。
以降の章で詳細を解説します。

動的承認ノード・確認ノード

Contents

- [displayFlag](#)
- [enableFlag](#)
- [searchCondition](#)
- [criteria](#)
- [processTargetConfigs](#)

動的承認ノード、および確認ノードに対する設定を行う場合、「DCNodeSetting」オブジェクトに、設定対象のノード単位で設定情報を定義します。

動的承認ノード、また確認ノードをいずれかひとつ設定する際のパラメータ例を示します。

```

1  {
2  // 動的承認ノード・確認ノード 設定情報
3  "DCNodeSetting" : {
4
5  // 設定対象ノードIDをプロパティキーとして指定
6  "%ノードID%" : {
7
8  // 利用者に標準処理画面上からノード設定を行わせるか否かを制御します。
9  "displayFlag" : true,
10
11 // ノード編集画面において「有効」チェックボックスの初期値を制御します。
12 "enableFlag" : true,
13
14 // 処理対象者の検索時条件を指定します。
15 "searchCondition" : {
16
17     "criteria" : {
18         "department_set_list" : [
19             {
20                 "company_cd" : "comp_sample_01",
21                 "department_set_cd" : "comp_sample_01",
22                 "department" : {
23                     "department_cd" : "dept_sample_10",
24                     "compare" : "ge"
25                 }
26             }
27         ]
28     }
29 },
30
31 // 処理対象プラグイン情報を指定します。
32 "processTargetConfigs" : [
33     {
34         "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
35         "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.user",
36         "parameter" : "maruyama"
37     },
38     {
39         "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
40         "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.role",
41         "parameter" : "im_workflow_user"
42     }
43 ]
44 }
45 }
46 }

```

以降では、個々のパラメータについて説明します。

displayFlag

利用者に標準処理画面上からノード設定を行わせるか否かを制御します。

設定値・設定する内容	利用者に標準処理画面上からノード設定を行わせるか否かを制御します。 true の場合、標準処理画面の「フロー設定」項目に表示します。 false の場合、標準処理画面の「フロー設定」項目に表示しません。
単位・型	真偽値
省略時の動作	true（表示する）
親オブジェクト	設定対象ノードオブジェクト

enableFlag

ノード編集画面において「有効」チェックボックスの初期値を制御します。

設定値・設定する内容	この設定は、動的承認ノードに対してのみ有効です。 ノード編集画面において「有効」チェックボックスの初期値を制御します。 この設定を行った場合、「有効」チェックボックスを表示します。 true の場合、有効状態で初期表示します。 false の場合、無効状態で初期表示します。 displayFlagと組み合わせて、両方にfalseを設定した場合、該当の動的承認ノードを削除（ノードスキップ）できます。
単位・型	真偽値

省略時の動作	「有効」チェックボックスの表示有無は、フロー定義における設定（「動的承認ノードの削除」設定）に従います。 「有効」チェックボックスが表示される設定の場合、チェックの有無は現在のフロー状態に応じて決定されます。
親オブジェクト	設定対象ノードオブジェクト

searchCondition

処理対象者の検索時条件を指定します。

設定値・設定する内容	処理対象者の検索時条件を指定します。 「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。 当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索（キーワード タブ）」のみです。
単位・型	オブジェクト（次のプロパティを定義可能） <ul style="list-style-type: none"> criteria
省略時の動作	検索条件（暗黙条件）指定なしで動作します。
親オブジェクト	設定対象ノードオブジェクト

criteria

IM-共通マスタのユーザ検索（キーワード タブ）に対する暗黙条件を指定します。
ユーザ検索（キーワード タブ）に対する暗黙条件の仕様については「[IM-共通マスタ 検索画面仕様書](#)」を参照してください。

設定値・設定する内容	ユーザ検索（キーワード タブ）に対する暗黙条件を指定可能です。 具体的には、「 IM-共通マスタ 検索画面起動引数一覧 」において以下に該当する引数を指定可能です。 <ul style="list-style-type: none"> 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 対象となる引数 分類 = 「暗黙条件」
単位・型	オブジェクト 有効な暗黙条件、および暗黙条件の構造については、「 IM-共通マスタ 検索画面起動引数一覧 」を参照してください。
省略時の動作	searchCondition を指定した場合、省略することはできません。 searchCondition ・ criteriaの両方が省略された場合、検索条件（暗黙条件）指定なしで動作します。
親オブジェクト	searchCondition

processTargetConfigs

処理対象プラグイン情報を指定します。

設定値・設定する内容	処理対象プラグイン情報を指定します。 配列の要素として処理対象プラグインオブジェクトを複数設定可能です。
単位・型	配列（各要素はオブジェクト（次のプロパティを定義）） <ul style="list-style-type: none"> extensionPointId （拡張ポイントID） pluginId （プラグインID） parameter （パラメータ） <p>※指定可能な拡張ポイントIDは次のとおりです。</p> <ul style="list-style-type: none"> 動的承認ノード： jp.co.intra_mart.workflow.plugin.authority.node.dynamic 確認ノード： jp.co.intra_mart.workflow.plugin.authority.node.confirm <p>※利用可能なプラグインIDについては「IM-Workflow 仕様書」の「処理権限者プラグイン一覧」を参照してください。</p>
省略時の動作	現在設定済みの処理対象者が適用されます。
親オブジェクト	設定対象ノードオブジェクト

Contents

- [displayFlag](#)
- [searchCondition](#)
- [criteria](#)
- [matterNodeExpansions](#)
- [nodeName](#)
- [processTargetConfigModel](#)

横配置ノード、および縦配置ノードに対する設定を行う場合、「HVNodeSetting」オブジェクトに、設定対象のノード単位で設定情報を定義します。

横配置ノード、または縦配置ノードをいずれかひとつ設定する際のパラメータ例を示します。

```

1  {
2  // 横配置ノード・縦配置ノード 設定情報
3  "HVNodeSetting": {
4
5  // 設定対象ノードIDをプロパティキーとして指定
6  "%ノードID%": {
7
8  // 利用者に標準処理画面上からノード設定を行わせるか否かを制御します。
9  "displayFlag": true,
10
11 // 処理対象者の検索時条件を指定します。
12 "searchCondition": {
13   "criteria": {
14     "department_set_list": [
15       {
16         "company_cd": "comp_sample_01",
17         "department_set_cd": "comp_sample_01",
18         "department": {
19           "department_cd": "dept_sample_10",
20           "compare": "ge"
21         }
22       }
23     ]
24   }
25 },
26
27 // ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。
28 "matterNodeExpansions": [
29
30 // ひとつめの展開ノード
31 {
32   // ノード名を指定します。
33   "nodeName": "node_name_001",
34
35   // 処理対象者の検索時条件を指定します。
36   "searchCondition": {
37     "criteria": {
38       "department_set_list": [
39         {
40           "company_cd": "comp_sample_01",
41           "department_set_cd": "comp_sample_01",
42           "department": {
43             "department_cd": "dept_sample_10",
44             "compare": "ge"
45           }
46         }
47       ]
48     }
49   },
50
51 // 処理対象プラグイン情報を指定します。
52   "processTargetConfigModel": [
53     {
54       "extensionPointId": "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
55       "pluginId": "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
56       "parameter": "comp_sample_01^comp_sample_01^dept_sample_11"
57     },
58     {
59       "extensionPointId": "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
60       "pluginId": "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
61       "parameter": "comp_sample_01^comp_sample_01^dept_sample_12"
62     }
63   ]
64 },
65
66 // ふたつめの展開ノード
67 {
68   // ノード名を指定します。
69   "nodeName": "node_name_002",
70
71   // 処理対象者の検索時条件を指定します。
72   "searchCondition": {
73     "criteria": {
74       "department_set_list": [
75         {
76           "company_cd": "comp_sample_01",
77           "department_set_cd": "comp_sample_01",
78           "department": {
79             "department_cd": "dept_sample_20",
80             "compare": "eq"
81           }
82         }
83       ]
84     }
85   },
86
87 // ...

```

```

84     }
85     },
86
87     // 処理対象プラグイン情報を指定します。
88     "processTargetConfigModel" : [
89     {
90         "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
91         "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
92         "parameter" : "comp_sample_01^comp_sample_01^dept_sample_21"
93     },
94     {
95         "extensionPointId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic",
96         "pluginId" : "jp.co.intra_mart.workflow.plugin.authority.node.dynamic.department",
97         "parameter" : "comp_sample_01^comp_sample_01^dept_sample_22"
98     }
99     ]
100 }
101 ]
102 }
103 }
104 }
    
```

以降では、個々のパラメータについて説明します。

displayFlag

利用者に標準処理画面上からノード設定を行わせるか否かを制御します。

設定値・設定する内容	利用者に標準処理画面上からノード設定を行わせるか否かを制御します。 true の場合、標準処理画面の「フロー設定」項目に表示します。 false の場合、標準処理画面の「フロー設定」項目に表示しません。
単位・型	真偽値
省略時の動作	true (表示する)
親オブジェクト	設定対象ノードオブジェクト

searchCondition

処理対象者の検索時条件を指定します。

当パラメータは、以下の用途でそれぞれ設定が可能です。

- 設定対象の横配置ノード・縦配置ノード単体における全体設定
- ノード展開情報単位の個別設定

上記を同時に指定した場合、個別設定は全体設定より優先して動作します。

設定値・設定する内容	処理対象者の検索時条件を指定します。 「criteria」プロパティを設定することで、検索時の暗黙条件を指定可能です。 当プロパティを指定した場合、処理対象者を検索する際に利用可能なプラグインと検索タブは「ユーザ検索 (キーワード タブ)」のみです。 この挙動は、全体設定・個別設定を問わず、どちらか一方でも当プロパティを指定した場合に適用されます。
単位・型	オブジェクト (次のプロパティを定義可能) ■ criteria
省略時の動作	検索条件 (暗黙条件) 指定なしで動作します。
親オブジェクト	設定対象ノードオブジェクト、または matterNodeExpansions

criteria

IM-共通マスタのユーザ検索 (キーワード タブ) に対する暗黙条件を指定します。

ユーザ検索 (キーワード タブ) に対する暗黙条件の仕様については「[IM-共通マスタ 検索画面仕様書](#)」を参照してください。

設定値・設定する内容	<p>ユーザ検索（キーワード タブ）に対する暗黙条件を指定可能です。 具体的には、「IM-共通マスタ 検索画面起動引数一覧」において以下に該当する引数を指定可能です。</p> <ul style="list-style-type: none"> 対象となる検索画面・タブ 機能グループ = 「ユーザ検索画面」 検索画面タブ = 「キーワード」 対象となる引数 分類 = 「暗黙条件」
単位・型	<p>オブジェクト 有効な暗黙条件、および暗黙条件の構造については、「IM-共通マスタ 検索画面起動引数一覧」を参照してください。</p>
省略時の動作	<p>searchCondition を指定した場合、省略することはできません。 searchCondition ・ criteriaの両方が省略された場合、検索条件（暗黙条件）指定なしで動作します。</p>
親オブジェクト	searchCondition

matterNodeExpansions

ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。

設定値・設定する内容	<p>ノード展開情報を指定します。展開するノード数分オブジェクトを定義します。</p> <p>displayFlag が false（表示しない） の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最小」「最大」値による制限は行わず、当パラメータで定義したノード数で展開されます。 <p>displayFlag が true（表示する） の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> 展開可能なノード数の最小個数は、常にフロー定義で設定されている「割当可能ノード数」の「最小」値です。フロー定義の最小値より当パラメータで指定したノード数の方が小さい場合、処理を行うためには標準画面上での追加設定が必要です。 展開可能なノード数の最大個数は、以下の値のうち大きい方が適用されます。 <ul style="list-style-type: none"> フロー定義で設定されている「割当可能ノード数」の「最大」値 当パラメータで指定したノード数 <p>対象のノードを削除（ノードスキップ）する場合、以下の設定を行ってください。</p> <ul style="list-style-type: none"> フロー定義の「割当可能ノード数（最小）」が0の場合 matterNodeExpansionsの設定を省略、または空で指定してください。 フロー定義の「割当可能ノード数（最小）」が1以上の場合 displayFlagにFalse、matterNodeExpansionsに空を指定してください。
単位・型	<p>配列（各要素はオブジェクト（次のプロパティを定義可能））</p> <ul style="list-style-type: none"> nodeName processTargetConfigModel searchCondition
省略時の動作	<p>displayFlag が false（表示しない） の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> 設定対象のノードが未展開の場合、「割当可能ノード数」の「最小」値のノード数で展開されます。その際の処理対象者は、ルート定義で設定された処理対象者です。 設定対象のノードがすでに展開されている場合、現在設定済みのノード展開情報と処理対象者が適用されます。 <p>displayFlag が true（表示する） の場合の動作仕様は以下の通りです。</p> <ul style="list-style-type: none"> フロー定義で設定された「割当可能ノード数」の「最小」値・「最大」値、およびルート定義で設定された処理対象者で動作します。
親オブジェクト	設定対象ノードオブジェクト

nodeName

ノード名を指定します。

設定値・設定する内容	ノード名を指定します。
単位・型	String
省略時の動作	設定対象の横配置ノード・縦配置ノードのノード名が適用されます。

[processTargetConfigModel](#)

処理対象プラグイン情報を指定します。

設定値・設定する内容	処理対象プラグイン情報を指定します。 配列の要素として処理対象プラグインオブジェクトを複数設定可能です。
単位・型	配列（各要素はオブジェクト（次のプロパティを定義）） <ul style="list-style-type: none">extensionPointId（拡張ポイントID）pluginId（プラグインID）parameter（パラメータ） <p>※指定可能な拡張ポイントIDは次のとおりです。 jp.co.intra_mart.workflow.plugin.authority.node.dynamic</p> <p>※利用可能なプラグインIDについては「IM-Workflow 仕様書」の「処理権限者プラグイン一覧」を参照してください。</p>
省略時の動作	現在設定済みの処理対象者が適用されます。
親オブジェクト	matterNodeExpansions

