



# 目次

---

- 1. 改訂情報
- 2. はじめに
- 3. 権限リストを拡張する

## 改訂情報

---

---

変更年月日	変更内容
-------	------

---

2016-04-01	初版
------------	----

---

## はじめに

---

項目

- [このガイドについて](#)

## このガイドについて

---

このガイドでは、招待機能の拡張方法および注意点について解説します。

## 権限リストを拡張する

### 項目

- [この機能について](#)
- [実装済みの招待権限デコレータ](#)
- [実装方法](#)
- [設定方法](#)

## この機能について

招待情報には権限リストが表示されています。

この権限リストに表示されている各項目は招待権限デコレータと呼ばれるものです。

この招待権限デコレータによって招待されたユーザに対して、様々な処理(権限の付与など)を実行できます。例えば、以下のようなことが可能です。

- ロールを付与
- 組織に所属 など

ここでは招待権限デコレータの作成方法について解説します。

## 実装済みの招待権限デコレータ

標準で定義されている招待権限デコレータは以下の通りです。

**招待権  
限デコ  
レータ**

**ID** im\_role

**説明** 招待されたユーザに対してロールを付与します。  
パラメータで付与するロールを設定します。  
ロールの有効開始日は付与した日付となります。

付与するロールが存在しない場合は付与を行いません。  
警告ログが出力されます。

付与するロールがすでに付与されている場合は付与を行いません。  
警告ログが出力されます。

**デコ  
レータ  
クラス** [jp.co.intra\\_mart.foundation.external.user.invitation.decorator.RoleInvitationDecorator](http://jp.co.intra_mart.foundation.external.user.invitation.decorator.RoleInvitationDecorator)

パラ im\_external\_user/invitation/admin/decorators/role  
メータ  
編集用  
画面パ  
ス

---

招待 組織に所属する  
権限  
デコ  
レー  
タ

---

**ID** im\_department

---

**説明** 招待されたユーザを組織に所属させます。  
パラメータで所属させる組織を設定します。  
ユーザが組織に所属していない場合は、1つ目に指定された組織が主所属となります。  
所属開始日は所属した日付となります。

所属させる組織が存在しない場合は所属させません。  
警告ログが出力されます。

所属させる組織にすでに所属している場合は所属させません。  
警告ログが出力されます。

---

デコ [jp.co.intra\\_mart.foundation.external.user.invitation.decorator.DepartmentInvitationDecorator](http://jp.co.intra_mart.foundation.external.user.invitation.decorator.DepartmentInvitationDecorator)  
レー  
タク  
ラス

---

パラ im\_external\_user/invitation/admin/decorators/department  
メー  
タ編  
集用  
画面  
パス

---

招待 アプリケーションライセンスを付与する  
権限  
デコ  
レー  
タ

---

**ID** im\_app\_license

---

**説明** 招待されたユーザにアプリケーションライセンスを付与します。  
パラメータで付与するアプリケーションライセンスを設定します。

付与するアプリケーションライセンスが存在しない場合は付与を行いません。  
警告ログが出力されます。

付与するアプリケーションライセンスが上限により付与できない場合は付与を行いません。  
警告ログが出力されます。

付与するアプリケーションライセンスが既に付与されていた場合は付与を行いません。  
警告ログが出力されます。

**デコレータクラス** [jp.co.intra\\_mart.foundation.external.user.invitation.decorator.AppLicenseInvitationDecorator](http://jp.co.intra_mart.foundation.external.user.invitation.decorator.AppLicenseInvitationDecorator)

**パラメータ編集画面パス** im\_external\_user/invitation/admin/decorators/app\_license

**招待権限デコレータ** 外部ユーザ

**ID** im\_external\_user

**説明** 招待されたユーザを外部ユーザとします。  
パラメータはありません。

**デコレータクラス** [jp.co.intra\\_mart.foundation.external.user.invitation.decorator.ExternalUserInvitationDecorator](http://jp.co.intra_mart.foundation.external.user.invitation.decorator.ExternalUserInvitationDecorator)

**パラメータ編集画面パス** なし

招待権限デコレータの実装は、以下のインタフェースを実装して作成します。

### **jp.co.intra\_mart.foundation.external.user.invitation.decorator.InvitationDecorator**

以下の関数に処理を実装します。

#### **decorate(String userCd, String parameter)**

【関数のパラメータ】

<b>userCd</b>	招待したユーザコードです。
<b>parameter</b>	招待権限デコレータに設定されたパラメータです。 招待情報の権限リストの各招待権限デコレータで設定された値が渡されます。

---

作成したプログラムはコンパイル後にWEB-INF/classes配下に配置します。

### 招待権限デコレータのパラメータ編集画面の作成

---

招待権限デコレータにパラメータを設定する際に、パラメータ設定するための独自の画面(検索画面など)を提供する必要がある場合は作成が必要です。

パラメータを必要としない場合は作成する必要はありません。

このプログラムは招待情報の新規作成および編集画面に埋め込まれます。

なお、パラメータを単純な文字列として登録する場合は標準のプログラムが利用可能です。  
標準のプログラムのパスは以下の通りです。

#### **im\_external\_user/invitation/admin/decorators/standard**

ここでは im\_external\_user/invitation/admin/decorators/standard を参考にして解説します。

【HTML】



```

<!-- パラメータのテキストボックスがあるダイアログの定義です。-->
<imart type="imuiDialog" id=$bind.dialogId title=$bind.dialogName width="400" autoOpen="false"
modal="true" buttons=$bind.dialogButtons >
  <table class="imui-form">
    <tbody>
      <tr>
        <th class="nowrap">
          <label><imart type="message" id="CAP.Z.IWP.EXTERNAL.USER.INVITATION.PARAMETER" />
</label>
        </th>
        <td>
          <imart type="imuiTextbox" id=$bind.paramId name=$bind.paramId value=""
style="width:99%"/>
        </td>
      </tr>
    </tbody>
  </table>
</imart>

<script type="text/javascript">
  <!-- 初期登録処理です。-->
  jQuery(function() {
    <!-- 編集アイコンがクリックされた場合の処理関数を登録します。-->
    setAuthzEditor('<imart type="string" value=$bind.id />',function(id,params,paramInfo) {
      <!-- パラメータのテキストボックスにパラメータの値を設定します。-->
      jQuery('#<imart type="string" value=$bind.paramId />').val(params);
      <!-- ダイアログを開きます。-->
      jQuery('#<imart type="string" value=$bind.dialogId />').imuiDialog('open');
    });
  });

  <!-- ダイアログのOKボタンが押された場合の処理です。-->
  function <imart type="string" value=$bind.funcOkName/>() {
    <!-- パラメータのテキストボックスに入力された値を取得します。-->
    var value = jQuery('#<imart type="string" value=$bind.paramId />').val();
    <!-- パラメータ情報オブジェクトを作成します。-->
    var paramInfo = {
      "displayName" : value,
      "data" : []
    }
    <!-- 権限リストにパラメータ情報を保存します。-->
    setAuthzParam('<imart type="string" value=$bind.id />',value,paramInfo);
    <!-- ダイアログを閉じます。-->
    jQuery('#<imart type="string" value=$bind.dialogId />').imuiDialog('close');
  }

  <!-- ダイアログのCancelボタンが押された場合の処理です。-->
  function <imart type="string" value=$bind.funcNgName/>() {

    <!-- ダイアログを閉じます。-->
    jQuery('#<imart type="string" value=$bind.dialogId />').imuiDialog('close');
  }
</script>

```

上記のプログラムのクライアントJavaScriptの部分について説明します。

招待情報の権限リスト内の編集ボタンをクリックした時に呼び出される関数が事前に用意されています。  
登録関数は以下の通りです。この関数はページロード時に登録します。

### setAuthzEditor(id,func)

<b>id</b>	招待権限デコレータのIDです。 招待権限デコレータのIDは【サーバサイドJavaScript】のinit関数の引数(request)のidプロパティとして渡されます。
<b>func</b>	認証情報の権限リスト内の編集ボタンをクリックされた時に実行する関数を設定します。 この関数に渡される引数は以下の通りです。

### function(id,params,paramInfo)

<b>id</b>	招待権限デコレータのIDです。
<b>params</b>	実際のパラメータの値(文字列)です。
<b>paramInfo</b>	パラメータの情報オブジェクトです。 下記の <b>パラメータ情報オブジェクト</b> を参照してください。  この値は以下の処理で設定された値です。 <ul style="list-style-type: none"> <li>【サーバサイドJavaScript】の getParamInfo(params)の戻り値（初期値）</li> <li>【HTML】の setAuthzParam(id,params,paramInfo)で設定した <b>paramInfo</b> の値</li> </ul>

次に、パラメータを設定するダイアログなどで設定された内容を保存/表示する関数が事前に用意されています。

保存関数は以下の通りです

### setAuthzParam(id,params,paramInfo)

この関数を利用して設定したパラメータ情報オブジェクトは以下の関数を呼び出す時に引数に渡されます。

- 認証情報の権限リスト内の編集ボタンをクリックされた時に呼び出される関数

<b>id</b>	招待権限デコレータのIDです。
<b>params</b>	実際のパラメータの値(文字列)を指定します。
<b>paramInfo</b>	パラメータの情報オブジェクトです。 下記の <b>パラメータ情報オブジェクト</b> を参照してください。

**displayName** プロパティ 認証情報の権限リストのパラメータ部分に表示する内容(文字列)です。

**data** プロパティ 内部的に保管するパラメータの補足情報です。  
パラメータ設定ダイアログなどで既存のパラメータ情報を再表示する場合に必要な補足情報を保管しておくことができます。  
利用しない場合は空の配列を指定します。

### 【サーバサイドJavaScript】

```
var $bind = {};  
function init(request){  
  // request.idに招待権限デコレータのIDが設定されています。  
  $bind.id = request.id;  
  
  // 関数名やダイアログ内の識別子が重複しないように、固有の識別子を作成します。  
  // ダイアログのIDです。  
  $bind.dialogId = "dialog_" + request.id;  
  // ダイアログのタイトルです。  
  $bind.dialogName =  
  MessageManager.getMessage("CAP.Z.IWP.EXTERNAL.USER.INVITATION.DECORATOR.STANDARD.DIALOG.  
  
  // ダイアログ内のパラメータ用テキストボックスのIDです。  
  $bind.paramId = "param_" + request.id  
  // ダイアログでOKボタンが押されたときの関数名です。  
  $bind.funcOkName = $bind.dialogId + '_ok';  
  // ダイアログでCancelボタンが押されたときの関数名です。  
  $bind.funcNgName = $bind.dialogId + '_ng';  
  
  // ダイアログのボタン定義です。  
  $bind.dialogButtons = [  
    { text: 'OK', click: $bind.funcOkName},  
    { text: 'Cancel', click: $bind.funcNgName}  
  ];  
}  
  
// 初期表示のために呼び出される関数です。  
// この関数は必ず定義します。  
function getParamInfo(params) {  
  // パラメータ情報オブジェクトを作成して返却します。  
  return {  
    "displayName" : (params) ? params : "",  
    "data"       : []  
  };  
}
```

まずは、init関数で初期化処理を記述します。

init関数の引数(request)のid プロパティに招待権限デコレータのIDが設定されています。

次に以下の関数を定義します。

### getParamInfo(params)

引数には実際のパラメータの文字列が渡されます。

この引数の情報を元にパラメータ情報オブジェクトを作成して返却します。

この関数で返却したパラメータ情報オブジェクトは以下の関数を呼び出す時に引数に渡されます。

- 認証情報の権限リスト内の編集ボタンをクリックされた時に呼び出される関数



#### 注意

この関数が呼び出される前にinit関数は呼び出されません。

作成したプログラムはWEB-INF/jssp/src配下に配置します。

## 設定方法

---

作成したプログラムを利用するための設定方法は、「[設定ファイルリファレンス](#)」-「[招待権限リスト設定](#)」を参照してください。