

クライアントサイド入力チェック

海外csm(ati-csm-web)におけるクライアントサイド入力チェックについて記載する。
(海外csm以外で必要な場合はAP基盤へ連絡してください。)

方針

- 単項目の検証
 - 単純な入力検証のみ実施する。
 - 次の検証を対象とする。
 - 必須項目の検証
- 複数項目の検証、条件付きの単項目の検証
 - クライアントサイドでは実施しない。

入力検証に必要なJSファイル

エラーメッセージをモーダルダイアログで表示するためのJSを設定する

PC・SPそれぞれのベースとなるTiles定義にダイアログオープン用の共通JSとエラーメッセージダイアログ用のJSを追加する。

PC

Tiles実装例

```
<!-- omitted -->
<definition name="hoge-layout" template="/WEB-INF/views/common/hoge.jsp">
  <!-- omitted -->
  <put-list-attribute name="jsPathList" inherit="true">
    <!-- omitted -->
    <add-attribute value="/resources/js/dialog.js" /> <!-- <1> -->
    <add-attribute value="/resources/js/ati-warning-modal-dialog.js" /> <!-- <2> -->
    <!-- omitted -->
  <!-- omitted -->
<!-- omitted -->
```

- <1> ダイアログオープン用の共通JS(TCI提供)を追加する。
- <2> エラーメッセージダイアログ用のJSを追加する。

SP

Tiles実装例

```
<!-- omitted -->
<definition name="hoge-sp-layout" template="/WEB-INF/views/common/sp_hoge.jsp">
  <!-- omitted -->
  <put-list-attribute name="jsPathList" inherit="true">
    <!-- omitted -->
    <add-attribute value="/resources/js/aswtour_common.js" /> <!-- <1> -->
    <add-attribute value="/resources/js/ati-warning-modal-dialog-sp.js" /> <!-- <2> -->
    <!-- omitted -->
  <!-- omitted -->
<!-- omitted -->
```

- <1> ダイアログオープン用の共通JS(TCI提供)を追加する。
- <2> エラーメッセージダイアログ用のJSを追加する。

バリデーションルールの書き方

name属性値がフォーム内で一意になるフォームコントロール(テキストボックス、セレクトボックスなど)

JSP実装例

```
<form:form action="${zzzUrl}" method="post" modelAttribute="xxxForm"
data-ati-valid-validate="true" >
<!-- omitted -->
<!-- <1> -->
<form:input path="foo" type="text"
data-ati-valid-required="true"
data-ati-valid-required-message="Fooを入力してください。(ERR99991)" />
<form:select path="bar"
data-ati-valid-required="true"
data-ati-valid-required-message="Barを選択してください。(ERR99992)" />
<form:options items="${barList}" itemLabel="bar" itemValue="id" />
</form:select>
<!-- omitted -->
```

- <1> 対象のフォームコントロールに以下のdata属性を設定する。
 - data-ati-valid-required属性: "true"を設定する。
 - data-ati-valid-required-message属性: エラー時に表示するメッセージを設定する。

name属性値がフォーム内で複数あるフォームコントロール(チェックボックス、ラジオボタン)

JSP実装例

```
<!-- <1> -->
<form:form action="${zzzUrl}" method="post" modelAttribute="xxxForm"
data-ati-valid-validate="true"
data-ati-valid-foo-required="true"
data-ati-valid-foo-required-message="Fooにチェックを入れてください。(ERR99993)"
data-ati-valid-bar-required="true"
data-ati-valid-bar-required-message="Barを選択してください。(ERR99992)" >

<form:checkboxes path="foo" items="${fooList}"/>
<form:radiobuttons path="bar" items="${barList}"/>
<!-- omitted -->
```

- <1>
フォームに以下のdata属性を設定する。 [name]には入力検証の対象となるフォームコントロールのname属性値を設定すること。
 - data-ati-valid-[name]-required属性: "true"を設定する。
 - data-ati-valid-[name]-required-message属性: エラー時に表示するメッセージを設定する。

JSでバリデーションルールを設定する方法

name属性値が配列になっているフォームコントロール(name="fooList[0].hoge"等)については前述の書き方では対応できないため、JSでバリデーションルールを設定する。

JS実装例

```
$(function() {
    'use strict';

    // <1>
    var validationSettings = [
        {
            field: 'fooList[0].foo',
            ruleName: 'required',
            ruleParam: true,
            message: 'Fooを入力してください。(ERR99993)'
        },
        {
```

```

        field: 'barList[0].bar',
        ruleName: 'required',
        ruleParam: true,
        message: 'Barを選択してください。(ERR99992)'
    }
];

// <2>
ati.validator.configure($form, validationSettings);

});

```

- <1> field, ruleName, ruleParam, messageプロパティを持つオブジェクト配列にバリデーションルールを設定する。
 - field: 入力検証の対象となるフォームコントロールのname属性値を設定する。
 - ruleName: 'required' を設定する。
 - ruleParam: true を設定する。
 - message: エラー時に表示するメッセージを設定する
- <2> ati.validator.configure(\$form, validationSettings)メソッドでバリデーションルールをフォームに設定する。
 - 参考: [クライアントサイド入力検証](#)

入力検証の実施方法

フォームに data-ati-valid-validate属性を設定することで入力検証が有効となり、サブミット時に入力検証が行われる。但し、動的に追加されるform要素(モーダルダイアログのform要素など)をサブミットする場合は、個別画面のJSでの対応が必要となるため

[個別画面のJSで対応する場合](#) を参考にすること。

フォームの自動入力検証を有効にする

JSP実装例

```

<!-- <1> -->
<form:form action="{zzzUrl}" method="post" modelAttribute="xxxForm"
data-ati-valid-validate="true" >
<!-- omitted -->
</form:form>

```

- <1> フォームに以下のdata属性を設定する。
 - data-ati-valid-validate属性: "true"を設定する。

[JSでバリデーションルールを設定する方法](#) でバリデーションルールを設定している場合は設定不要。(ati.validator.configure メソッド内でフォームの自動入力検証を有効にしているため)

個別画面のJSで対応する場合

JS実装例

```

// 個別画面のJS
$form.on('submit', function(e){
    // omitted
    if (!ati.validator.valid($form)) // <1>
        return false;
    }
    // omitted
});

```

- <1> ati.validator.valid(\$form)メソッドで入力検証を実施する。
 - 結果がfalseの場合は、イベントをキャンセルする
 - 参考: [クライアントサイド入力検証](#)