

## 二重送信防止

### 対応方針

二重送信防止の対応方法として下記を適用する。

- JavaScriptによるボタンの2度押し防止
- PRGパターンの適用
- トランザクショントークンチェックについては実施しない。
  - [#1160](#) で検討され、API側の仕様上、トランザクショントークンによるチェックは不要となった。

### 参考

- [4.5. 二重送信防止 — TERASOLUNA Server Framework for Java \(5.x\) Development Guideline 5.3.0.RELEASE documentation](#)

### JavaScriptによるボタンの2度押し防止

- POSTメソッドでフォームをサブミットするものについては、AP基盤提供の共通JS(atd-app-initializer.js)でボタンの2度押し防止を適用しているため基本的には個別画面のJSで意識する必要は無い。
- 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以下の実装例を参考にすること。

### 個別画面のJSで対応する場合

#### 実装方針

- AP基盤提供の [フォームを制御するユーティリティ](#) を使用すること
  - atd.form.isSubmitBlocking : フォーム送信のブロック中判定
  - atd.form.blockSubmit : フォーム送信のブロック
  - atd.form.unblockSubmit : フォーム送信のブロック解除

動的に追加されたフォーム(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

#### 実装例

```
// 個別画面のJS
$dialog.on('submit', '#dialog-form', function() {
  // イベントの開始

  if(atd.form.isSubmitBlocking($form)) { // <1>
    return false;
  }

  // Ajaxリクエストを送信
  atc.ajax(contextPath'/search/DSBP0260/writeCartSession', {
    type: 'POST',
    data: $form.serialize(),
    context: {
      ...,
      form: $form // <2>
    }
  }).do(function(data) {

  // イベントの終了
}
```

- <1> イベント開始時に atd.form.isSubmitBlocking() メソッドでフォーム送信がブロック中かどうかを確認する。フォーム送信がブロック中の場合は、イベントをキャンセルする。

- <2> [Ajax](#) を利用する場合は、二重送信防止を適用するform要素を指定すること。

## PRGパターンの適用

- データの更新を伴う処理の場合は、PRGパターンを適用すること。

### 参考

- [4.5.2.2. PRGパターンについて — TERASOLUNA Server Framework for Java \(5.x\) Development Guideline 5.3.0.RELEASE documentation](#)