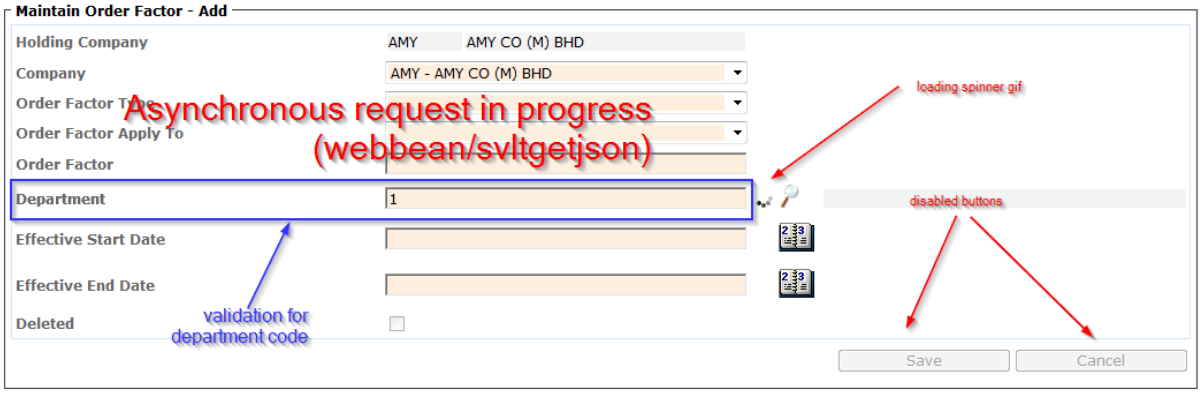
AsyncLoadingStatus.js



given

- one event handler which performs one or more asynchronous requests
- callbacks attached to those asynchronous requests
- buttons (to be disabled to prevent premature form submission)
- reference to a loading gif(s)

will

- disable buttons and show loading gif when event is triggered
- renable buttons and hide loading gif when *all* callbacks have returned

Usage

constructor `AsyncLoadingStatus` takes one (optional) configuration object as argument

CONFIGURATION OBJECT PROPERTIES

- **all options are optional**
  - even the configuration object is optional
  - but then you need to call .getHandler() and .getCallback() to do anything at all

options

- `buttons`
  - an array of buttons / a jquery object
  - if an array, each button can be any value accepted by the jQuery constructor e.g. element, selector, jQuery object
  - if unspecified: **defaults to all buttons on page**
- `loadingGif`
  - any value accepted by the jQuery constructor e.g. element, selector, jQuery object
  - if unspecified: **defaults to all instances of profit/images/ajax-loader.gif on the page**
- `handler`
  - the name of a handler (usually for 'change' event) to modify
  - if specified then you don't need to use asyncLoadingStatus.getHandler()

- callbacks
    - the name of a callback, or of a list of callbacks to modify
    - if specified then you don't need to use asyncLoadingStatus.getCallback()

INSTANCE METHODS

- getCallback
    - returns a modified callback
    - which will hide the loading gif and enable buttons when it's called
- getHandler
    - returns a modified handler
    - which will show the loading gif and disable buttons when it returns
- done
    - call this method if within your handler you abort and do not initiate the asynchronous request(s)
    - (so that the callbacks are never called)

Example usage

importing the external js file

Note that you need to also include jquery.

```
<script language="javascript" src="<%=BaseURL%>/profit/JScript/jquery-1.4.4.min.js"></script>
<script language="javascript"
src="<%=BaseURL%>/profit/JScript/AsyncLoadingStatus.min.js"></script>
```

placing the gif

by specifying element within which to append the gif

```
      AsyncLoadingStatus({
        placeGifIn: $('.place_gif_store'),
        baseURL: '<%=BaseURL%>' // you are required to specify baseURL when using placeGifIn
      });

      <tr>
        <td class="caption"><%=jbWResGUI.getRes("From Store")%></td>

        <td><input name="fromStore" maxlength="<%=SYSStoreLength%>"></td>
        <td class="place_gif_store">
          <img src="<%=BaseURL%>/profit/images/search.gif" alt="<%=jbWResGUI.getRes("Find")%>"
onclick="SearchCriteria('<%=BaseURL%>/profit/MT/StrmstSearch.jsp', 'FROM_STORE')"
style='cursor:hand;'>
        </td>
      </tr>
```

manual placement: put the ajax loading gif wherever you want it

```
   AsyncLoadingStatus({
        loadingGif: $('#ajax-loader')
      });

      <img id="ajax-loader" src="<%=BaseURL%>/profit/images/ajax-loader.gif"
alt="checking...">
```

async method

SvltGetJson

```
function onChangeStore(store) {
  var query = "...";

  // this function is redefined each time onChangeStore is called
  var callback = function(data) {
    // has access to 'store' via closure
  };

  // use .getCallback(callback, true) if the callback is defined each time the handler is
called
  $.getJSON("<%=BaseURL%>/servlet/SvltGetJson", {query: query},
asyncLoadingStatus.getCallback(callback, true));
}

var asyncLoadingStatus; // make this global so that we can see it in onChangeStore

$(document).ready(function() {
  asyncLoadingStatus = AsyncLoadingStatus({
    handler: 'onChangeStore'
  });
});
```

webbean

```
    function onChangeStore() {
      // ...
    }

    function onChangeStoreCalledBack() {
      // ...
    }

    $(document).ready(function() {
      AsyncLoadingStatus({
        handler: 'onChangeStore',
        callbacks: 'onChangeStoreCalledBack'
      });
    });
```

use of .done()

```
  function onChangeStore() {

      // nonasychronous checking
      // e.g. check if form field value is a number
      if (badvalue()) {
        asyncLoadingStatus.done(); // if within the onchange method the asynchronous requests
are not made
        return;
      }

      // asychronous checking
      checkAutostrmstByPrimaryKey();
    }
```

multiple asynchronous requests per handler

```
function onChangeStore() {
  // ...
}

function onChangeStoreCalledBack1() {
  // ...
}

function onChangeStoreCalledBack2() {
  // ...
}

$(document).ready(function() {
  AsyncLoadingStatus({
    handler: 'onChangeStore',
    callbacks: ['onChangeStoreCalledBack1', 'onChangeStoreCalledBack2']
  });
});
```

only disable some buttons

```
var asyncLoadingStatus = AsyncLoadingStatus({
  buttons: ['cmdSave', 'cmdAdd'] // only disable save buttons
});
```

features

- supports multiple callbacks per handler
  - will only hide gif/re-enable buttons when all callbacks have returned
- supports different instances of AsyncLoadingStatus sharing the same gif/buttons
  - will only hide gif/re-enable button when all callbacks in all instances involving the gif/button have returned
- supports callbacks which are defined each time the handler is called
  - often the case for the SvltGetJson method
  - use .getCallback(callback, true)
- supports cancellation of the request(s)
  - use .done()

issues

- due to webbean/SvltGetJson not responding upon some type of errors, callbacks may never be called
  - so the program will hide the gif and re-enable buttons after 60 seconds
  - you can always call .done() to make it happen sooner