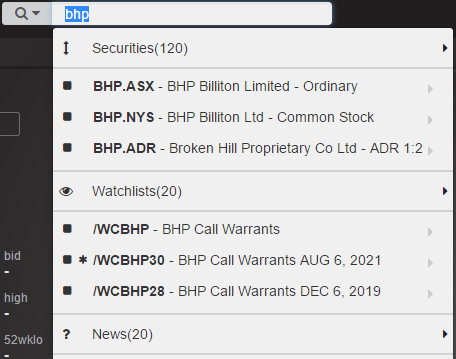
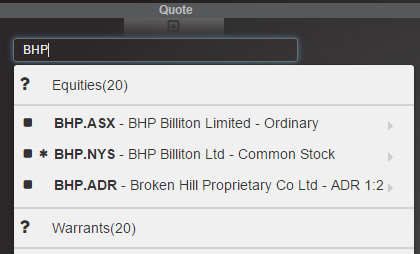
Search

Search box component is used on global header and in several widgets. They look like this:

On global header:



In Quote Widget:



# Relationship of classes in search



# Using search

Search box component is available if you want to use it as a whole:

<div class="container-hor" data-bind="component: {name: 'search-box', params: {name: 'test10', categories: [eSearchCategory.all]}}">

<search-box class="container-hor" params="name: 'test32', categories: [eSearchCategory.equities], showAdvance: false, showFlyouts: false" />

<search-box class="container-hor" params="name: 'test20', categories: [eSearchCategory.securities, eSearchCategory.watchlists, eSearchCategory.layout, eSearchCategory.google, eSearchCategory.people], icons: {'securities': 'fa fa-lock'}, searchers: [{ id: eSearchCategory.people, name: undefined, searcher: 'Dummy' }]" />

Or you can use the custom binding if you want to bind it with an existing input:

<input type="text" class="form-control code-input input-sm" placeholder="Search.." data-bind="***search: { name: 'quote', categories: [eSearchCategory.securities], autoBreakDown: true, showAdvance: false, showFlyouts: false, value: securityCode, event: {result: \_onResult} },*** value:securityCode, valueUpdate:'afterkeydown', event:{'keyup': enterSearch}, attr:{'data-element':'QuoteDetail\_input\_'+dataName()}" data-i18n="[placeholder]common.input.search">

# Custom binding

SearchBox.ts:

ko.bindingHandlers["allowBindings"] = {

init: function (elem, valueAccessor) {

var shouldAllowBindings = ko.unwrap(valueAccessor());

return { controlsDescendantBindings: !shouldAllowBindings };

}

};

ko.bindingHandlers["search"] = {

init: function (element: any,

valueAccessor: () => any,

allBindingsAccessor: () => any,

viewModel: any,

bindingContext: KnockoutBindingContext) {

var options = valueAccessor();

var box = new IressTraderPlus.Search.SearchBox(options);

var childBindingContext = bindingContext.createChildContext(box);

var div = document.createElement("div");

var parent = $(element).parent();

ko.renderTemplate("Tmpl\_SearchBox", childBindingContext, {}, div, "replaceChildren");

var ul = $(div).find("ul[aria-label='search-result']");

ul.attr("data-bind", "allowBindings: false");

ul.css("position", "relative");

ul.appendTo(parent);

if (!parent.hasClass("dropdown")) parent.addClass("dropdown");

var input = $(element);

if (!input.hasClass("dropdown-toggle")) input.addClass("dropdown-toggle");

input.removeAttr("data-toggle");

input.attr("data-toggle", "dropdown");

input.on("focus", function (event) {

box["\_onFocus"].apply(box, [box, event, input]);

});

input.on("blur", function (event) {

box["\_onBlur"].apply(box, [box, event, input]);

});

input.on("click", function (event) {

box["\_onClick"].apply(box, [box, event, input]);

});

input.on("keyup", function (event) {

box["\_onKeyup"].apply(box, [box, event, input]);

});

},

update: function (element, valueAccessor, allBindingsAccessor, viewModel, bindingContext) {

}

};