
COMP 4021
Internet Computing

jQuery and Autocomplete

Dik Lun Lee

Autocomplete in General

- ❑ Autocomplete displays a list of “suggested” values as user types and allows user to a value to trigger action
 - E.g., Type in a query in Bing’s search box and see how it works
- ❑ Imagine how much work is required to implement autocomplete with JavaScript only:
 - Create event trigger for each keystroke
 - Display a text box (div) under the search box
 - Fill the box with suggestions, which most likely is obtained dynamically from the server (e.g., server returns the most popular queries matching the partial query user has typed in)
 - Handle “small things”: Do not display suggestions when user is typing very fast; backspace also updates the list

jQuery Autocomplete

- ❑ JQuery autocomplete is a **widget** under JQuery UI
- ❑ A text field widget tied to a text field
- ❑ Needs a data source containing the values to be displayed
- ❑ Autocomplete is a JQuery “plugin”, download from:
 - <http://jqueryui.com/download/>

Data Source

❑ Source of data could be:

- Static list on client side, like an hardcoded array (see below)
- Dynamic list coming from server

```
var languages = ['java', 'javascript', 'perl', 'python', 'php'];  
$( "#searchBox" ).autocomplete( {  
    minLength: 2, // min # of chars input before suggestions are triggered  
    delay: 200, // wait 0.2 sec before trigger  
    source: languages  
});
```

... ..

```
<input type="text" id="searchBox" name="q" size="20" value="">
```

❑ Question: How to create the list dynamically?

Source Data from Server

- ❑ As user types in the search box, input string is passed to server program, which returns list of “suggestions”

```
$( "#searchBox" ).autocomplete( {  
  minLength: 2,  
  source: "test.php"  
} );
```

- ❑ If user typed in “ata”, term=ata is passed to the server program as if the URL is test.php?term=ata
- ❑ Server returns suggestions (e.g., strings matching term as prefix, infix or suffix) in JSON which is converted to an array

Function Source

- ❑ We can use function as data source, which makes an `$.ajax(..)` call with url and data for the server program, and a response function to return data

```
$( "#searchBox" ).autocomplete( {  
    source: function(request, response)  
    { ... .. }  
} );
```

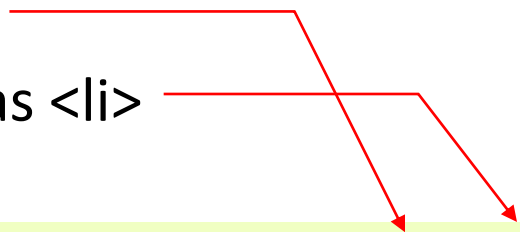
```
$.ajax({  
    url: "suggestion.php",  
    type: "POST",  
    dataType: "json",  
    data: {name: request.term },  
    success: function (data) {  
        ... .. }  
})
```

- ❑ **request**: an object with one attribute, term, whose value is the data in the text field
- ❑ **response**: callback function to receive an array of suggestions

Changing the Menu

- ❑ Control the display of the items in the “menu” (area for displaying suggestions)
- ❑ Menu is associated an element
- ❑ An **item** object is appended to as
 - **item**.label : string to display in menu
 - **item**.value : value to be inserted when item is selected

Process item.label to bold the suggestion properly, e.g., using regular expression



```
_renderItem: function( ul, item ) {  
    return $( "<li></li>" )  
        .attr( "data-value", item.value )  
        .append( item.label )  
        .appendTo( ul );  
}
```

Or use:

```
.data( "data-value", item.value )
```

Submitting Selected Item on Menu

- ❑ User moves mouse down the menu to select an item
- ❑ A “value” string is submitted to the server program (e.g., to do another retrieval)
- ❑ Needed only for the bonus points

```
select: function( event, ui ) {  
    $(event.target).val(ui.item.value);  
    $(event.target.form).submit();  
}
```


Take Home Message

- ❑ Autocomplete could have been very difficult to implement if you use JavaScript alone
- ❑ With jQuery UI autocomplete widget, it is a lot easier, although it is non-trivial to learn how to use it and you still have to take care of the server side programming
- ❑ There are many jQuery UI widget to make life easier