

jQuery UI: Autocompleters

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.



For customized training related to JavaScript or Java, email hall@coreservlets.com
Marty is also available for consulting and development support

Taught by lead author of *Core Servlets & JSP*,
co-author of *Core JSF* (4th Ed), and this tutorial.

Available at public venues, or
custom versions can be held on-site at your organization.

- **Courses developed and taught by Marty Hall**
 - JavaScript, jQuery, Ext JS, JSF 2.3, PrimeFaces, Java 8 programming, Spring Framework, Spring MVC, Android, GWT, custom mix of topics
 - Courses available in any state or country.
 - Maryland/DC companies can also choose afternoon/evening courses.
- **Courses developed and taught by coreservlets.com experts (edited by Marty)**
 - Hadoop, Hibernate/JPA, HTML5, RESTful Web Services

Contact hall@coreservlets.com for details



Topics in This Section

- Local data (source = array)
- Remote data (source = URL)
- Custom matcher (source = function)
- Custom menu styles
- Display values vs. insert values
- Performing side effects upon selection

5

coreservlets.com – custom onsite training



Setup for Examples

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **General setup**
 - jQuery and jQuery UI loaded as in previous lectures
 - smoothness theme is the default, as in previous lectures
- **Specific layout**
 - Each section is in its own accordion panel
 - All buttons are styled so they take on look of theme
 - HTML headings and sections use predefined CSS class names like “ui-widget-header” and “ui-widget-content” so that they take on the look of the theme
 - See section on themes for the exact HTML used

7

JavaScript for General Setup

- **Purpose**
 - Styles all the buttons
 - Sets up the accordion panel
 - Both concepts covered in previous lectures
- **Code**

```
$(function() {  
    ...  
    $(".button").button();  
    $("#accordion-panel").accordion({  
        collapsible: true, active: false, autoHeight: false  
    });  
});
```

8

Autocompleter Overview

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - Like the Google, Bing, and Yahoo home pages

- **HTML**

- Make a textfield with an id

```
<input id="field-id"/>
```

- **Basic JavaScript**

- Call `autocomplete()` on the textfield
- Supply a source option

```
$("#field-id").autocomplete({ source: ... });
```

 - There are three variations on the type of value for source: an array of values, a URL (string) that returns JSON array of choices, and a function
 - There are many options in addition to the required source

Options for source

- **Local data**

```
var choices = [ "Choice 1", "Choice 2", ...];
$("#field-id").autocomplete({ source: choices });
```

- **Remote data**

```
var url = "some relative URL that returns JSON array";
$("#field-id").autocomplete({ source: url });
```

- **Custom matching function**

```
function customMatcher(request, callback) {
    var arrayOfMatches = someLogic(request.term);
    callback(arrayOfMatches);
}
```

```
11 $("#field-id").autocomplete({ source: customMatcher });
```

Customizing Display of Values

- **Array of choices**

- The source option can directly take an array, can refer to a URL that returns an array, or can refer to a function that creates an array and passes it to a callback handler

- **Types of arrays**

- Strings
 - The value in the dropdown and the value that gets inserted upon selection are the same
- Objects with label and value properties
 - The label property is shown in dropdown, and the value property gets inserted upon selection

- **Difficult to use HTML in displayed values**

- In jQuery UI 1.8.1 and 1.8.2, values could contain HTML
- This was disabled in 1.8.3, and the fix is very obscure unless you use 3rd party extension
 - See <http://bugs.jqueryui.com/ticket/5918> for details

Most Important Other Options

- **select**
 - Function that operates when an item is selected
 - See upcoming example
- **minLength**
 - The number of characters that should be entered before suggestions are shown
 - Used when short text produces too many suggestions
 - Default is 1
- **delay**
 - The number of milliseconds after a keystroke to wait before showing suggestions
 - Default is 300

13

coreservlets.com – custom onsite training



Autocompleter with Local Data

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - Choices come from a JavaScript array
 - Case insensitive match anywhere in the choices (not just beginning of the word)

- **HTML**

- Make a textfield with an id

```
<input id="field-id"/>
```

- **Basic JavaScript**

- Make an array of choices (strings)

```
var choices = [ "Choice 1", "Choice 2", ...];
```
- Call autocomplete() on the textfield
- Supply the array as the source option

```
$("#field-id").autocomplete({ source: choices });
```

15

Example: HTML

Programming language:

```
<input id="langField1"/>
```

```
<button id="langSearchButton1">Search on Language</button>
```

16

Example: Main JavaScript

```
$(function() {  
    $("#langField1").autocomplete({ source: uiDemo.langArray });  
    ...  
});
```

Case insensitive match anywhere in the choices. So, entering "V" in the textfield will match "Java", "JavaScript", "Visual Basic", etc. See the upcoming custom matcher example for how to change the way that the match is performed.

```
var uiDemo = {}; // To avoid name conflicts
```

```
uiDemo.langString = "Java,C,C++,PHP,C#,Python,...";
```

```
uiDemo.langArray = uiDemo.langString.split(",");
```

These are 100 most popular programming languages, based on data from tiobe.com.
I put them into a single string and then split the string into an array to make it simpler to type them in.

17

Example: Auxiliary JavaScript

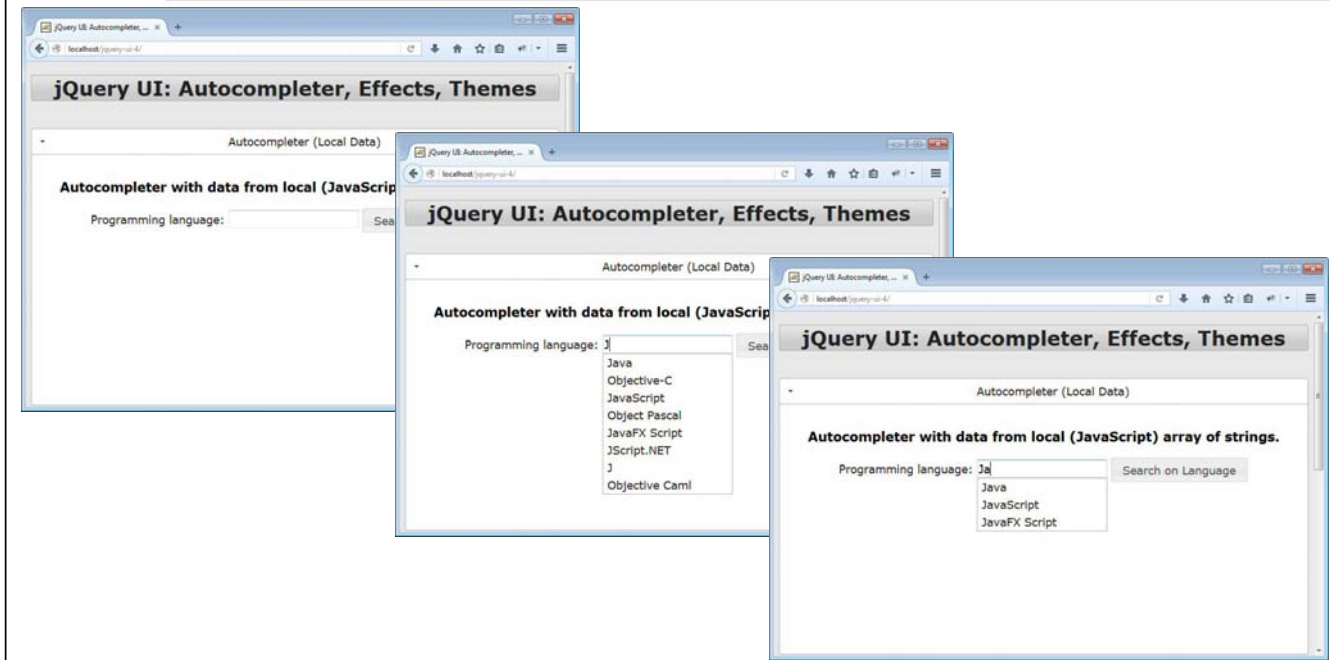
```
$(function() {  
    $("#langSearchButton1").click(function() {  
        uiDemo.googleSearch("#langField1");  
    });  
    ...  
});
```

```
uiDemo.googleSearch = function(selector) {  
    var language = escape($(selector).val());  
    window.location.href = "https://www.google.com/#q=" + language;  
};
```

Just to make the example do something, the button sends the entry from the textfield to Google. However, this functionality is really independent of the autocompleter. I won't repeat the code for the Google button in the later autocompleter examples.

18

Example: Results



coreservlets.com – custom onsite training



Autocompleter with Remote Data

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - jQuery UI calls the URL and automatically passes “term” parameter
 - Server returns a JSON array of strings (or objects, as we will see later)
 - Server can match any way it wants

- **HTML**

```
<input id="field-id"/>
```

- **Basic JavaScript**

- Call autocomplete() on the textfield.
- Supply a relative URL as the source option.

```
$("#field-id").autocomplete({ source: "some-address" });
```

- The remote address should take a “term” request parameter and return JSON (array of strings, or, as we will see later, object array)

21

Example: HTML

Programming language:

```
<input id="langField2"/>
```

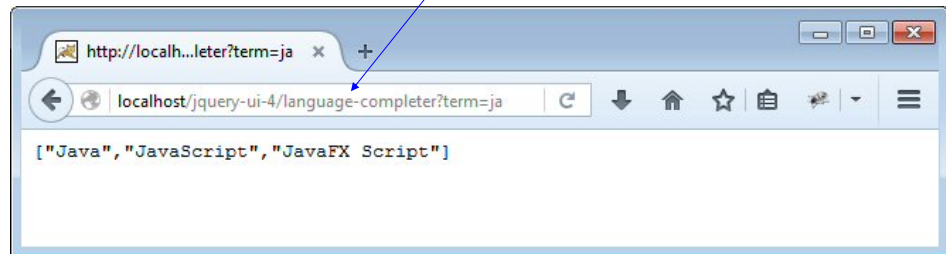
```
<button id="langSearchButton2">Search on Language</button>
```

22

Example: JavaScript

```
$(function() {  
    $("#langField2").autocomplete({  
        source: "language-completer"  
    });  
    ...  
});
```

URL of servlet that takes "term" request parameter and returns JSON array of strings.



23

Example: Server Code

```
@WebServlet("/language-completer")  
public class LanguageCompleter extends HttpServlet {  
    private static final String languageString =  
        "Java,C,C++,PHP,C#,Python,Visual Basic,...";  
    private static final String[] languageArray =  
        languageString.split(",");  
  
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        String languagePrefix = request.getParameter("term");  
        List<String> languages = findLanguages(languagePrefix);  
        response.setContentType("application/json");  
        PrintWriter out = response.getWriter();  
        out.print(new JSONArray(languages));  
    }  
}
```

Relative URL of servlet is language-completer. This can be set either in web.xml (any servlet version) or via the @WebServlet annotation (servlets 3.0+).

Uses the json.org utilities to turn Java List or array into JSON array. See separate lecture on automatic JSON generation. If you generate JSON by hand, remember that jQuery 1.4+ enforces strict JSON where property names must be in quotes and only double (not single) quotes are used. See sections on jQuery support for Ajax.

24

Example: Server Code (Continued)

```
private List<String> findLanguages(String languagePrefix) {  
    languagePrefix = languagePrefix.toUpperCase();  
    List<String> languages = new ArrayList<String>();  
    for(String language: languageArray) {  
        if(language.toUpperCase().startsWith(languagePrefix)) {  
            languages.add(language);  
        }  
    }  
    return(languages);  
}
```

Case insensitive match at the start of the choice. Server code can do any type of matching it wants; all that matters is that a JSON array is returned to the client.

25

Example: Results

Even though the server code uses the same set of languages, this is different result than for local data. With local data, jQuery automatically does a case-insensitive match anywhere within the string, so for the local data, entries like "Object Pascal" matched here. For remote data, your server-code can do anything it wants: it just needs to return an array of strings. In this case, the code matches only against the beginning of the string.

Autocompleter with Custom Matcher

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - Choices come from a JavaScript function. The function can use local array or remote array (e.g., via \$.ajax with dataType: "json"), and it can match however it wants.

- **HTML**

```
<input id="field-id"/>
```

- **Basic JavaScript**

- Make function that takes two args: request and callback. Use request.term to produce an array, then call callback(array)

```
function customMatcher(request, callback) {  
    var arrayOfMatches = someLogic(request.term);  
    callback(arrayOfMatches);  
}
```

- Call autocomplete() on the textfield; supply the function as the source option
`$("#field-id").autocomplete({ source: customMatcher });`

Example: HTML

Programming language:

```
<input id="langField3"/>
<button id="langSearchButton3">Search on Language</button>
```

29

Example: JavaScript

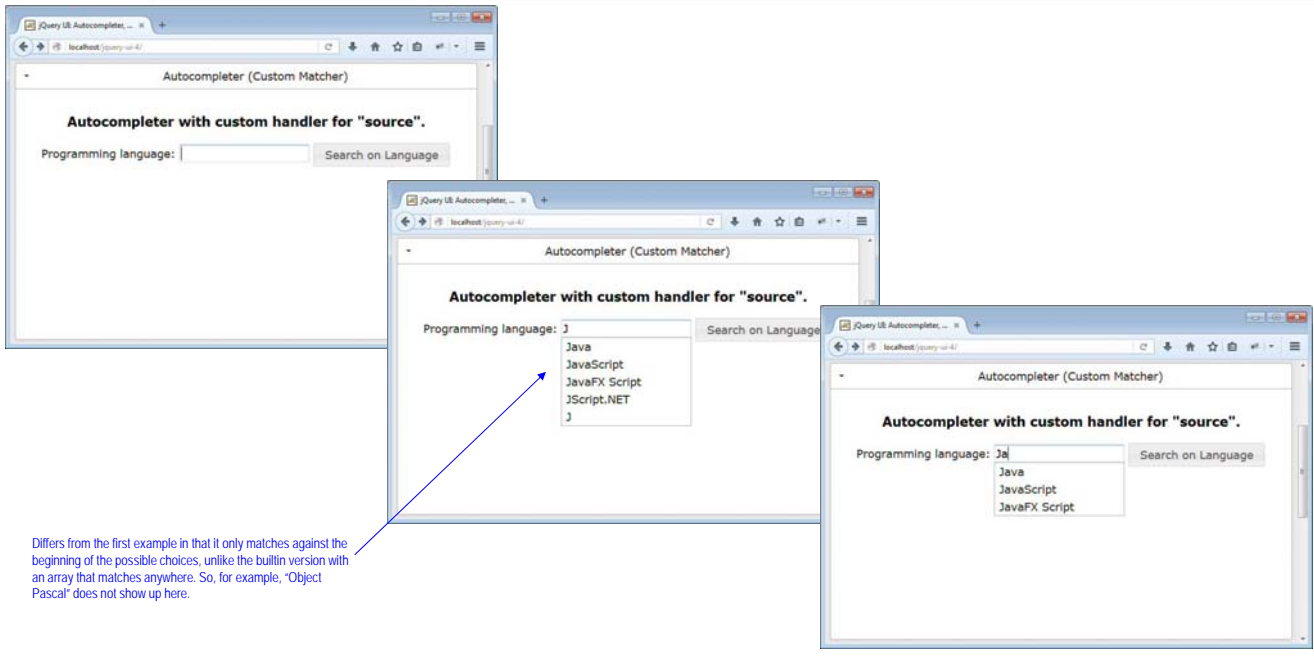
```
$(function() {
    $("#langField3").autocomplete({ source: uiDemo.langMatcher });
    ...
});
```

```
uiDemo.langMatcher = function(request, callback) {
    // Case-insensitive match at beginning of word.
    var regex = new RegExp("^" + request.term, "i");
    var matches = $.grep(uiDemo.langArray,
        function(item) {
            return(regex.test(item));
        });
    callback(matches);
};
```

30

Note that \$.grep is a jQuery utility function that returns an array of all elements that match test. Also note that someRegExp.test is a builtin method in normal JavaScript. Finally, if you want to worry about the user entering characters like . or * in the textfield, you can escape them using \$.ui.autocomplete.escapeRegex(request.term).

Example: Results



The image displays three browser windows showing an autocomplete interface titled "Autocompleter (Custom Matcher)". Each window has a "Programming language:" input field and a "Search on Language" button. The first window shows an empty input field. The second window shows the input field containing "J" and a dropdown menu listing "Java", "JavaScript", "JavaFX Script", "JScript.NET", and "J". A blue arrow points from a text block below to this dropdown. The third window shows the input field containing "Ja" and a dropdown menu listing "Java", "JavaScript", and "JavaFX Script".

Differs from the first example in that it only matches against the beginning of the possible choices, unlike the builtin version with an array that matches anywhere. So, for example, "Object Pascal" does not show up here.

coreservlets.com – custom onsite training



Autocompleter with CSS Styles Applied

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- You can apply your own CSS styles to the dropdown box
 - Use ".ui-autocomplete" or ".ui-autocomplete.ui-menu"
- Caution
 - Avoid fg and bg colors that might clash when you change the theme
 - Reasonable options: setting max-height (to support scrolling), setting opacity for semi-transparent menus, etc.

- **CSS**

```
.ui-autocomplete { ... }
```

- **JavaScript**

- If you apply styles directly as above, no extra JavaScript needed. You can also set a custom style, then apply it at runtime to the drop down

```
$( ".ui-autocomplete" ).toggleClass( "some-custom-name" );
```

33

Example: HTML

Programming language:

```
<input id="langField4"/>  
<button id="langSearchButton4">Search on Language</button>  
<button id="toggle-style-button">Toggle Menu Style</button>
```

34

Example: CSS

```
.fancy-menu {  
  opacity: 0.8;  
  max-height: 140px;  
  overflow-y: auto;  
}
```

If applied, results in a semi-transparent menu that has scrollbars if there are more than about 4 entries.

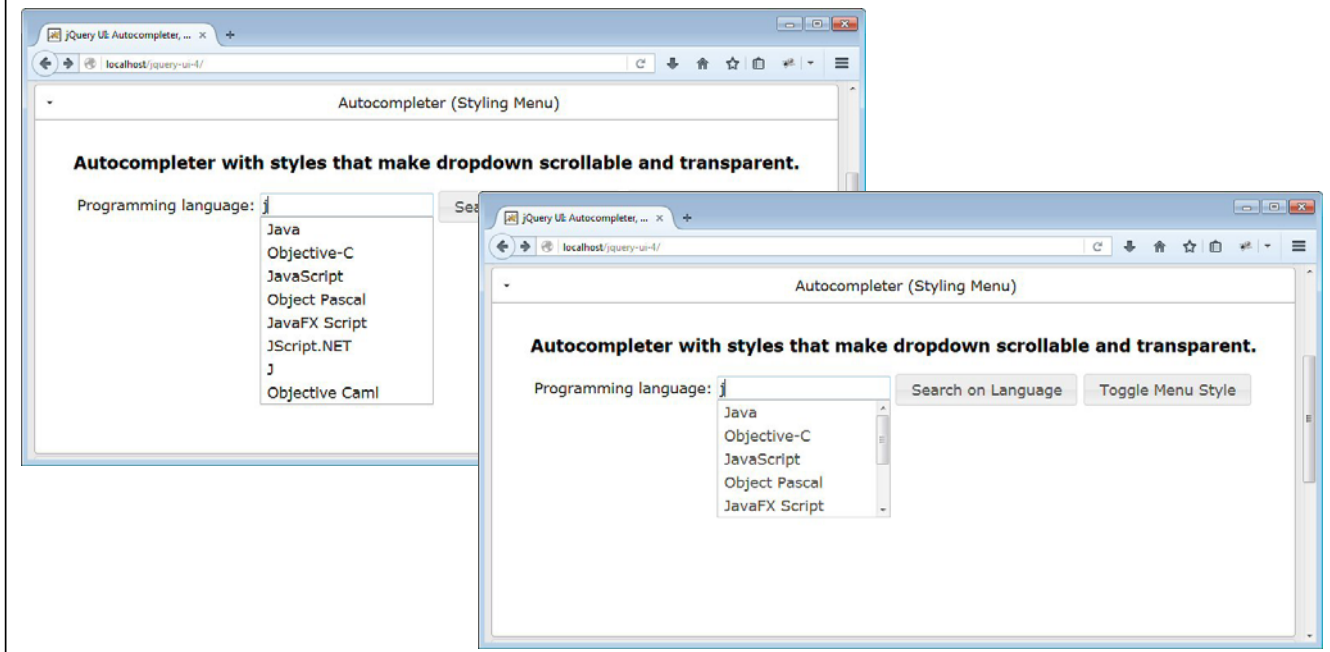
35

Example: JavaScript

```
$(function() {  
  $("#langField4").autocomplete({  
    source: uiDemo.langArray  
  });  
  $("#langSearchButton4").click(function() {  
    uiDemo.googleSearch("#langField4");  
  });  
  $("#toggle-style-button").click(function() {  
    $(".ui-autocomplete").toggleClass("fancy-menu");  
  });  
  ...  
});
```

36

Example: Results (Before and After Styling)



coreservlets.com – custom onsite training



Autocompleter with Array of Objects

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - Choices come from local array or array from server
 - Array contains JavaScript objects with at least 2 properties
 - label – what to show in dropdown for that entry
 - value – what to insert in textfield if that entry is selected

- **HTML**

```
<input id="field-id"/>
```

- **Basic JavaScript**

- Call autocomplete() on the textfield
- Supply array, URL, or function as the source option

```
$("#field-id").autocomplete({ source: ... });
```

 - But in all 3 cases, final result is array of objects with label and value properties, not array of strings

39

Simplified Example

- **Idea**

- Display dropdown values in mixed case
- Insert into textfield in upper case

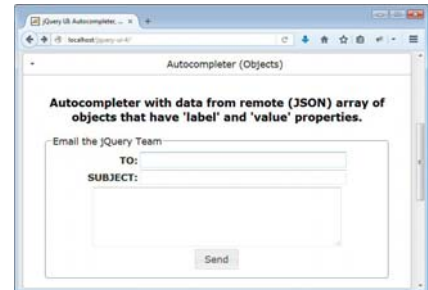
- **JavaScript code**

```
var choices = [ { label: "Java", value: "JAVA" },  
                { label: "Python", value: "PYTHON" },  
                ...  
              ];  
$("#field-id").autocomplete({ source: choices });
```

40

Example: HTML

```
<fieldset>
<legend>Email the jQuery Team</legend>
...
TO:
<input id="emailField1" size="40"/>
...
(some HTML that mimics a form for sending
email to member of the jQuery team)
</fieldset>
```



When the user types into the textfield, the full names of matching members of the jQuery and jQuery UI teams will be displayed. But, when the user selects a name from the list, both the name and the email address will be inserted into the textfield.

41

Example: JavaScript

```
$(function() {
    $("#emailField1").autocomplete({ source: "contact-completer" });
    ...
});
```

URL of servlet that takes "term" request parameter and returns JSON array of objects that have "label" (full name) and "value" (full name <email@address>) properties.

```
Result of http://localhost/jquery-ui-4/contact-completer?term=jo  
(Omitted some extra properties that are not used until a later example)  
[{"label": "John Resig",  
  "value": "John Resig <jresig3@example.com>"},  
 {"label": "Jorn Zaefferer",  
  "value": "Jorn Zaefferer <jz@example.com>"},  
 {"label": "Jonathan Sharp",  
  "value": "Jonathan Sharp <jsharp@example.com>"}]
```

42

Example: Server Code

```
@WebServlet("/contact-completer")
public class ContactCompleter extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String namePrefix = request.getParameter("term");
        List<Contact> contacts = findContacts(namePrefix);
        response.setContentType("application/json");
        PrintWriter out = response.getWriter();
        out.print(new JSONArray(contacts));
    }
}
```

Relative URL of servlet is
contact-completer.

Uses the json.org utilities to turn Java List or array into JSON array. Since the Contact objects have getLabel and getValue methods, the JSON objects will have label and value properties. It is not expected that non-Java programmers understand this code; just understand the output that you get for a given input, as shown on the earlier slide.

43

Example: Server Code (Continued)

```
private List<Contact> findContacts(String namePrefix) {
    namePrefix = namePrefix.toUpperCase();
    List<Contact> contacts = new ArrayList<Contact>();
    Contact[] jqueryContacts = ContactUtils.jqueryContacts();
    for(Contact contact: jqueryContacts) {
        String firstName = contact.getFirstName().toUpperCase();
        String lastName = contact.getLastName().toUpperCase();
        if(firstName.startsWith(namePrefix) || lastName.startsWith(namePrefix)) {
            contacts.add(contact);
        }
    }
    return(contacts);
}
```

Case insensitive match at the start of either first or last name

44

Example: Server Code (Contact Class)

```
public class Contact {
    private final String firstName, lastName, email, image;

    // Constructor and five getter methods for above instance vars

    public String getLabel() {
        return(firstName + " " + lastName);
    }

    public String getValue() {
        return(String.format("%s %s <%s>",
                               firstName, lastName, email));
    }
}
```

45

The json.org utilities automatically turn getter methods into properties. So, an instance of this class will be sent as
{ "firstName": "...", "lastName": "...", "label": "...", "value": "... }

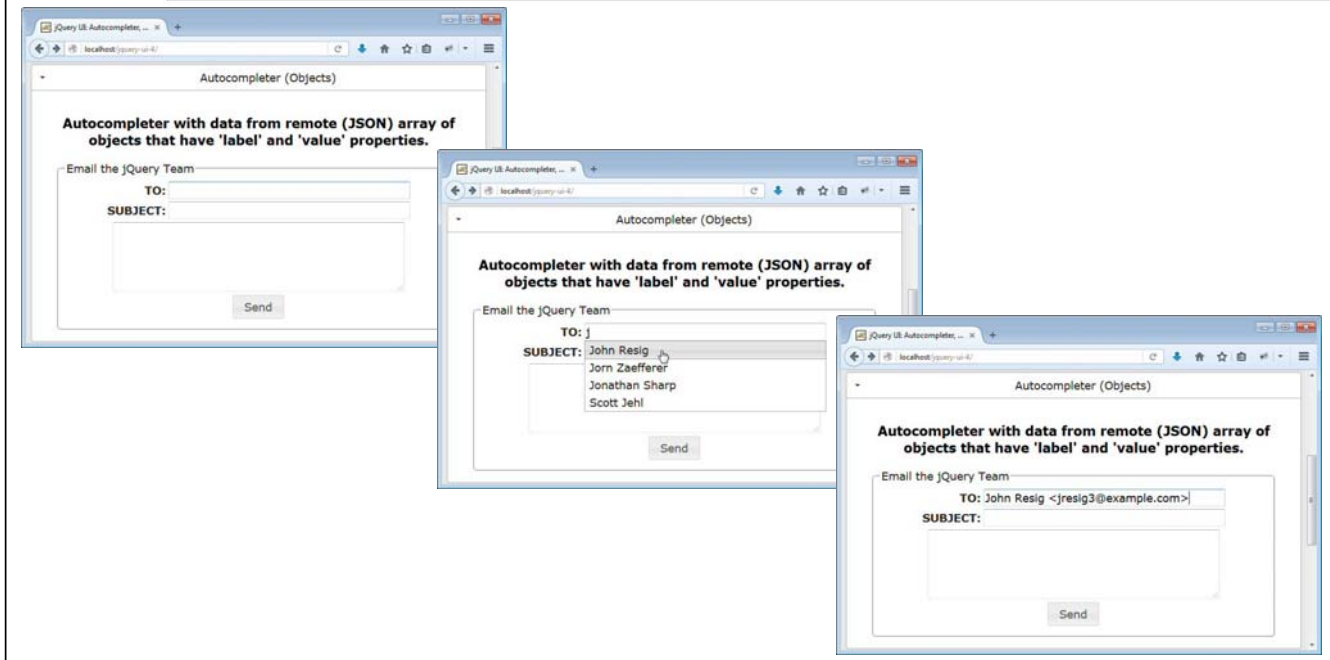
Example: Server Code (ContactUtils)

```
public class ContactUtils {
    private static Contact[] jqueryContacts = {
        new Contact("John", "Resig", "jresig3@example.com",
                    "http://static.jquery.com/.../john.jpg"),
        new Contact("Brandon", "Aaron", "brandon@example.com",
                    "http://static.jquery.com/.../brandon.jpg"),
        ...
    };
}
```

46

This gives the names and pictures of all members of the jQuery and jQuery UI teams, as given at <http://jquery.org/team>.
Email addresses are fictionalized to preserve privacy. The images are not used in this example, but will be in later example.

Example: Results



coreservlets.com – custom onsite training



Autocompleter: Using “select” to Perform Side Effects

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Overview

- **Idea**

- Use select to refer to function that will perform side effects
 - You can use the select option to tell jQuery UI to do additional side effects other than just inserting the selected entry (or value property of entry) into the textfield
 - Function takes event and ui object as argument. ui.item is the entry from the array. Use ui.item directly if you have an array of strings, or use ui.item.value if you have an array of objects. The objects can have extra properties with which you can perform additional tasks.

- **JavaScript**

- Call autocomplete()
- Supply function as the select option

```
function selectFunc(event, ui) {  
    $("#field-id").val(ui.item.value); // Insert into textfield  
    $("#other-id").doSomethingElse(ui.item.otherProp);  
}  
$("#field-id").autocomplete({ source: ..., select: selectFunc });
```

49

Simplified Example

- **Idea**

- In dropdown, show list of model numbers
- When one selected, insert into textfield in upper case
 - But also put text description in page

- **JavaScript code**

```
var choices =  
    [ { label: "a1234", value: "A1234", desc: "Cool Widget"},  
      { label: "b1234", value: "B1234", desc: "..." }, ... ];  
function selectHandler(event, ui) {  
    $("#field-id").val(ui.item.value); // Put val in textfield  
    $("#description-id").html(ui.item.desc); // Insert desc  
}  
$("#field-id").autocomplete({ source: choices,  
                             select: selectHandler });
```

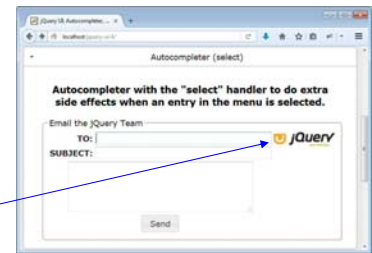
50

Example: HTML

```
<fieldset>
<legend>Email the jQuery Team</legend>
...
TO:
<input id="emailField2" size="40"/>
...

(some HTML that mimics a form for sending
email to member of the jQuery team)
</fieldset>
```

When the user types into the textfield, the full names of matching members of the jQuery and jQuery UI teams will be displayed. When the user selects a name from the list, both the name and the email address will be inserted into the textfield. In addition, the picture of the corresponding person will replace the jQuery UI logo.



51

Example: JavaScript

```
$(function() {
  $("#emailField2").autocomplete({
    source: "contact-completer",
    select: function(event, ui) {
      $("#emailField2").val(ui.item.value);
      $("#email-image").attr("src", ui.item.image);
    }
  });
  ...
});
```

Put value property of item (i.e., "Joe User <user@example.com>") into textfield

Set src attribute of image to the value of the image property of the selected item

52

Example: Server Code

- **Unchanged from previous example**

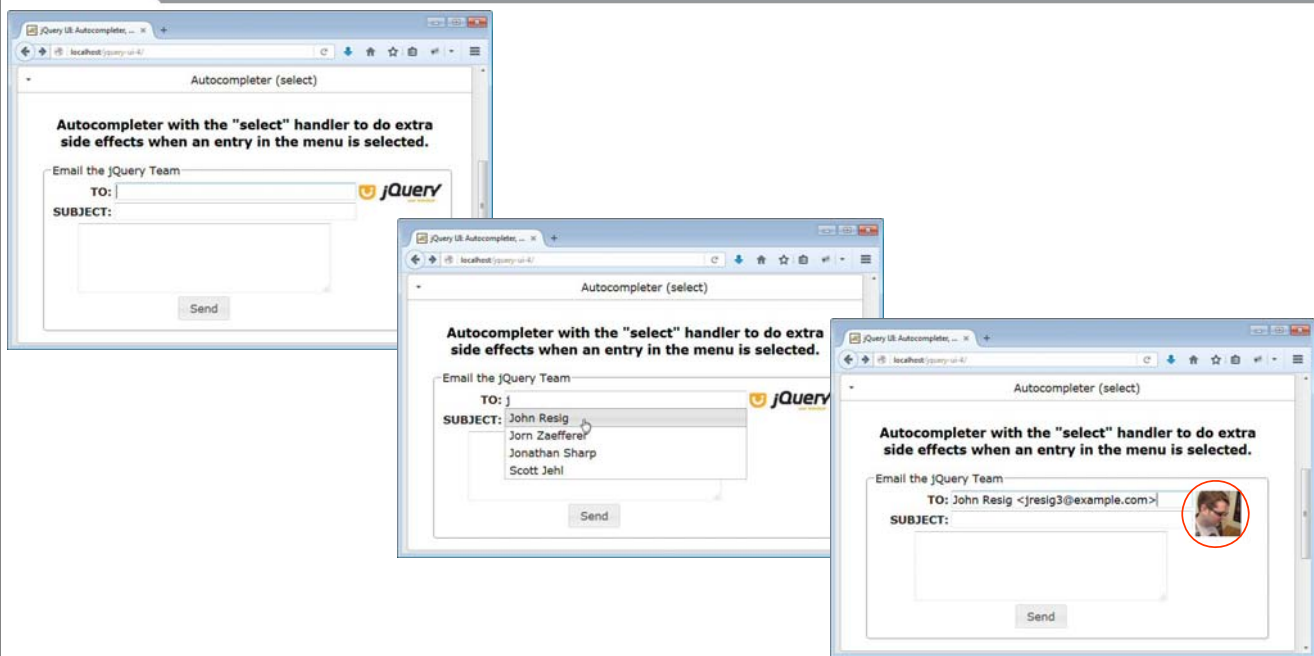
- In last example, we returned a JSON array of objects that had label, value, and image properties
 - Client code ignored the image property
- We will use the same server code in this example
 - But this time the client code will take the image property and use it for the src attribute of a previously blank image to the right of the input field

Result of <http://localhost/jquery-ui-4/contact-completer?term=jo>

```
[{"image": "http://static.jquery.com/org/images/team/john.jpg",  
  "label": "John Resig", "value": "John Resig <jresig3@example.com>"},  
 {"image": "http://static.jquery.com/org/images/team/joern.jpg",  
  "label": "Jorn Zaefferer", "value": "Jorn Zaefferer <jz@example.com>"},  
 {"image": "http://static.jquery.com/org/images/team/jonathan.jpg",  
  "label": "Jonathan Sharp", "value": "Jonathan Sharp <jsharp@example.com>"}]
```

53

Example: Results



Wrap-up

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

Summary

- **Local choices**

```
$("#field-id").autocomplete({ source: arrayOfChoices });
```

- **Remote choices**

```
$("#field-id").autocomplete({ source: "urlReturningArray" });
```

- **Custom choices**

```
$("#field-id").autocomplete({ source: matcherFunction });
```

- **Display values vs. insert values**

– The array should have **label** and **value** properties

- **Extra side effects**

```
$("#field-id").autocomplete({  
  source: anyOptionAbove,  
  select: functionToInsertIntoTextFieldThenDoMore  
});
```

Questions?

More info:

<http://www.coreservlets.com/javascript-jquery-tutorial/> – Tutorial on JavaScript, jQuery, and jQuery UI

<http://courses.coreservlets.com/course-materials/java.html> – General Java programming tutorial

<http://www.coreservlets.com/java-8-tutorial/> – Java 8 tutorial

<http://courses.coreservlets.com/java-training.html> – Customized Java training courses, at public venues or onsite at your organization

<http://coreservlets.com/> – JSF 2, PrimeFaces, Java 8, JavaScript, jQuery, Ext JS, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training
Many additional free tutorials at coreservlets.com (JSF, Android, Ajax, Hadoop, and lots more)

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.